

BEA 2025

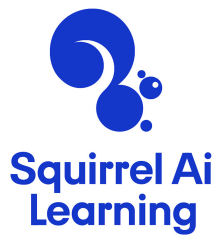
**The 20th Workshop on Innovative Use of NLP for Building
Educational Applications**

Proceedings of the Workshop

July 31 - August 1, 2025

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

Gold Level



©2025 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 979-8-89176-270-1

Introduction

This year marks the 20th edition of the *Workshop on Innovative Use of NLP for Building Educational Applications*. As in previous years, we are happy to welcome a plethora of work on various aspects and types of educational applications – from traditionally popular tasks around language learning to novel applications related to teaching math and programming languages. This year, we have also extended BEA to a 2-day event, which allowed us to accept more valuable work from our authors: in total, we received a record number of 169 submissions, and from these, we have accepted 12 papers as talks and 63 as poster and demo presentations, for an overall acceptance rate of 44 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. One aspect in which BEA continues to excel is the range of languages that are covered by the work submitted and presented at our workshop: this year, accepted papers feature work on educational applications developed for Arabic, English, Estonian, Finnish, Germanic languages, Indian languages, Italian, Romanian, Russian, and Spanish.

In addition to the diverse oral, poster and demo presentations, this year, Kostiantyn Omelianchuk from Grammarly will give a keynote on *How LLMs Are Reshaping GEC: Training, Evaluation, and Task Framing*. BEA 2025 will also include, for the first time, a half-day tutorial on *LLMs for Education: Understanding the Needs of Stakeholders, Current Capabilities and the Path Forward*. Finally, BEA 2025 has hosted a shared task on *Pedagogical Ability Assessment of AI-powered Tutors*, which attracted a large number of participants, and the program includes an oral presentation on the shared task from the organizers as well as extended poster sessions for shared tasks participants presenting their systems.

Last but not least, we would like to thank everyone who has been involved in organizing the BEA workshop this year. We are particularly grateful to our sponsors who keep providing their support to BEA: this year, our sponsors include Cambridge University Press & Assessment, Duolingo English Test, Grammarly, National Board of Medical Examiners, SigIQ.ai, and Squirrel Ai Learning.

BEA 2025 Organizing Committee

Organizing Committee

General Chair

Ekaterina Kochmar, MBZUAI

Program Chairs

Andrea Horbach, Hildesheim University
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 Chair

Zheng Yuan, University of Sheffield
Jill Burstein, Duolingo

Program Committee

Chairs

Bashar Alhafni, MBZUAI
Marie Bexte, FernUniversität in Hagen
Jill Burstein, Duolingo
Andrea Horbach, CAU Kiel
Ekaterina Kochmar, MBZUAI
Ronja Laarmann-Quante, Ruhr University Bochum
Anaïs Tack, KU Leuven; imec; UCLouvain
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
Syeda Sabrina Akter, George Mason University
Ali Al-Laith, University of Copenhagen
Giora Alexandron, Weizmann Institute of Science
David Alfter, Gothenburg University
Jatin Ambasana, Indian Institute of Technology Bombay
Jiyuan An, Beijing Language and Culture University
Antonios Anastasopoulos, George Mason University
Nico Andersen, DIPF | Leibniz Institute for Research and Information in Education
Aitor Arronte Alvarez, University of Hawaii at Manoa
Yuya Asano, University of Pittsburgh
Nischal Ashok Kumar, University of Massachusetts Amherst
Berk Atil, Pennsylvania State University
Shiva Baghel, Extramarks
Xiaoyu Bai, University of Potsdam
Jinhyun Bang, Samsung Research
Stefano Banno, University of Cambridge
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, University of Cambridge
Maryam Berijanlian, Michigan State University
Kay Berkling, Cooperative State University, Karlsruhe
Ummugul Bezirhan, Boston College, TIMSS and PIRLS International Study Center
Krishnakant Bhatt, IIT Bombay
Souvik Bhattacharyya, Lowe's
Abhidip Bhattacharyya, University of Massachusetts, Amherst
Serge Bibauw, Université catholique de Louvain
Louise Bloch, University of Applied Sciences and Arts Dortmund
Allison Bradford, University of California, Berkeley

Ted Briscoe, MBZUAI
Dominique Brunato, Institute of Computational Linguistics A. Zampolli"(ILC-CNR), Pisa
Ana-Maria Bucur, Interdisciplinary School of Doctoral Studies
Luciano Cabral, IFPE
Andrew Caines, University of Cambridge
Chris Callison-Burch, University of Pennsylvania
Jie Cao, University of Oklahoma
Dan Carpenter, North Carolina State University
Dumitru-Clementin Cercel, University Politehnica of Bucharest
Sophia Chan, Educational Testing Service Canada
Ignatios Charalampidis, University of Tuebingen
Andreas Chari, University of Glasgow
Danqing Chen, Technical University of Munich
Lei Chen, Jinan University
Mei-Hua Chen, Department of Foreign Languages and Literature, Tunghai University
Longfeng Chen, South China University of Technology
Artem Chernodub, ZenDesk
Mihail Chifligarov, Ruhr University Bochum
Luis Chiruzzo, Universidad de la Republica
Hyundong Cho, USC, Information Sciences Institute
Jinho D. Choi, Emory University
Evgeny Chukharev, Iowa State University
Yan Cong, Purdue University
Mark Core, University of Southern California
Sofía Correa Busquets, Pontificia Universidad Católica de Chile, National Center for Artificial Intelligence Chile, Foundational Research on Data Millenium Institute
Steven Coyne, Tohoku University / RIKEN
Scott Crossley, Georgia State University
Syaamantak Das, Indian Institute of Technology Bombay
Mihai Dascalu, University Politehnica of Bucharest
Tirthankar Dasgupta, Tata Consultancy Services Ltd.
Orphee De Clercq, LT3, Ghent University
Kordula De Kuthy, Universität Tübingen
Michiel De Vrindt, KU Leuven
Jasper Degraeuwe, Ghent University
FATIMA DEKMAK, American University of Beirut
Carrie Demmans Epp, University of Alberta
Dorottya Demszky, Stanford University
Aniket Deroy, IIT Kharagpur
Chris Develder, Ghent University
Srijita Dhar, Chittagong University of Engineering & Technology
Yuning Ding, FernUniversität in Hagen
Rahul Divekar, Educational Testing Service
George Duenas, Universidad Pedagogica Nacional
Marius Dumitran, University of Bucharest
Yo Ehara, Tokyo Gakugei University
Walid El Hefny, Leibniz-Institut für Wissensmedien (IWM)
Mohamed Elaraby, University of Pittsburgh
Ron Eliav, Bar-Ilan University
Jordan Esiason, North Carolina State University
Yao-Chung Fan, National Chung Hsing University

Effat Farhana, Auburn University
Mariano Felice, British Council
Nigel Fernandez, University of Massachusetts Amherst
Michael Flor, Educational Testing Service
Jennifer-Carmen Frey, EURAC Research
Benjamin Gagl, University of Cologne
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
Rujun Gao, Texas A&M University
Lingyu Gao, Toyota Technological Institute at Chicago
Ritik Garg, Extramarks Education Pvt. Ltd.
Voula Giouli, Aristotle University of Thessaloniki / ILSP, ATHENA RC
Sebastian Gombert, DIPF | Leibniz Institute for Research and Information in Education
Kiel Gonzales, University of the Philippines Diliman
Mark Edward Gonzales, De La Salle University
Cyril Goutte, National Research Council Canada
Pranav Gupta, Lowe's
Abigail Gurin Schleifer, Weizmann Institute of Science
Eleonora Guzzi, Universidade da Coruña
Ching Nam Hang, Assistant Professor, Yam Pak Charitable Foundation School of Computing and Information Sciences, Saint Francis University, Hong Kong
Ikhlusal Hanif, Universitas Indonesia
Jiangang Hao, Educational Testing Service
Ahatsham Hayat, University of Nebraska-Lincoln
Ping He, Northeastern University
Nicolas Hernandez, Nantes University
Michael Holcomb, University of Texas Southwestern Medical Center
Matias Hoyl, Stanford University
Chieh-Yang Huang, MetaMetrics Inc
Aiden Huang, Acton-Boxborough Regional High School
Chung-Chi Huang, Frostburg State University
Anna Huelsing, CAU
Leo Huovinen, Metropolia University of Applied Sciences
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
Raunak Jain, Intuit
Suriya Prakash Jambunathan, New York University
Qinjin Jia, North Carolina State University
Helen Jin, University of Pennsylvania
Abel John, Stanford University
Douglas Jones, MIT Lincoln Laboratory
Edmund Jones, Cambridge University Press & Assessment
Léane Jourdan, Nantes University
Samarth Kadaba, Stanford University
Indika Kahanda, University of North Florida
Tomoyuki Kajiwara, Ehime University
Honeiah Karimi, Cambium Assessment

Anisia Katinskaia, University of Helsinki
Fatemeh Kazemi Vanhari, McMaster University
Elma Kerz, Exaia Technologies
Fazel Keshtkar, St. John's University
Samin Khan, Stanford University
Darya Kharlamova, National Research University Higher School of Economics
Harksoo Kim, Konkuk University
Han Kyul Kim, University of Southern California
Levi King, Indiana University
Kasper Knudsen, ITU
David Kogan, Google
Mamoru Komachi, Hitotsubashi University
Charles Koutcheme, Aalto 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
Murathan Kurfali, RISE Research Institutes of Sweden
Alexander Kwako, University of California, Los Angeles
Kristopher Kyle, University of Oregon
Yunshi Lan, East China Normal University
Antonio Laverghetta Jr., Pennsylvania State University
Jaewook Lee, UMass Amherst
Celine Lee, Cornell University
Seolhwa Lee, Technical University of Darmstadt
Travis Lee, Tennessee Tech University
Bernardo Leite, Faculty of Engineering - University of Porto
Arun Balajee Lekshmi Narayanan, University of Pittsburgh
Xu Li, Zhejiang University
Hariz Liew, Singapore University of Social Sciences
Chuan-Jie Lin, National Taiwan Ocean University
Yudong Liu, Western Washington University
Naiming Liu, Rice University
Zhexiong Liu, University of Pittsburgh
Julian Lohmann, Christian Albrechts Universität Kiel
Benny Longwill, Educational Testing Service
Anastassia Loukina, Grammarly Inc
Crisron Rudolf Lucas, University College Dublin
Zhihao Lyu, CU Boulder
Sarah Löber, University of Tübingen
Denise Löfflad, Leibniz-Institut für Wissensmedien Tübingen
Wanjing (Anyu) Ma, Stanford University
Jakub Macina, ETH Zurich
Lieve Macken, Ghent University
Nitin Madnani, Duolingo
Hang Man, The University of Hong Kong
Zhenjiang Mao, University of Florida
Jacek Marciniak, Adam Mickiewicz University
Arianna Masciolini, University of Gothenburg
Sandeep Mathias, Presidency University

Kaushal Maurya, MBZUAI
Hunter McNichols, University of Massachusetts Amherst
Detmar Meurers, Leibniz-Institut für Wissensmedien (IWM)
Noah-Manuel Michael, Kiel University
Amit Mishra, AmityUniversityMadhyaPradesh
Daniel Mora Melanchthon, Leibniz Institute for Science and Mathematics Education
Sai Sathvik Motamarri, PES University
Phoebe Mulcaire, Duolingo
Laura Musto, Universidad de la Republica
Karthika N J, Indian Institute of Technology Bombay
Farah Nadeem, LUMS
Numaan Naeem, MBZUAI
Ryo Nagata, Konan University
Sungjin Nam, ACT, Inc
Diane Napolitano, The Washington Post
Aneet Narendranath, Michigan Technological University
Léo Nebel, LIP6 - Sorbonne Université
Kamel Nebhi, Education First
Seyed Parsa Neshaei, EPFL
Huy Nguyen, Amazon
Gebregziabihier Nigusie, Mizan-Tepi University
S Jaya Nirmala, National Institute of Technology Tiruchirappalli
Sergiu Nisioi, Human Language Technologies Research Center, University of Bucharest
Adam Nohejl, Nara Institute of Science and Technology
Eda Okur, Intel Labs
Kostiantyn Omelianchuk, Grammarly
Amin Omidvar, PhD student at the Department of Electrical Engineering and Computer Science, York University
Joshua Otten, GeorgeMasonUniversity
Daniel Oyeniran, University of Alabama
Ulrike Pado, HFT Stuttgart
Sankalan Pal Chowdhury, ETH Zurich
Nisarg Parikh, University of Massachusetts, Amherst
Jeiyoon Park, SOOP
Manooshree Patel, University of California, Berkeley
Kaushal Patil, University of Southern California
Kseniia Petukhova, MBZUAI
Henry Pit, University of Melbourne
Long Qin, Alibaba
Mengyang Qiu, Trent University
Marti Quixal, University of Tuebingen
Chatrine Qwaider, MBZUAI
Md. Abdur Rahman, Southeast University
Vatsal Raina, University of Cambridge
Sparsh Rastogi, Thapar Institute of Engineering and Technology
pranshu rastogi, Independent Researcher
Manav Rathod, University of California, Berkeley
Hanumant Redkar, Goa University, Goa
Robert Reynolds, Brigham Young University
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
Donya Rooein, Bocconi University
Aiala Rosá, Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Allen Roush, University of Oregon
Alla Rozovskaya, Queens College, City University of New York
Josef Ruppenhofer, Fernuniversität in Hagen
Stefan Ruseti, University Politehnica of Bucharest
Johannes Rückert, University of Applied Sciences and Arts Dortmund
Mariam Saeed, Applied Innovation Center
Trishita Saha, IIT Hyderabad
Jonathan Sakunkoo, Stanford University
Annabella Sakunkoo, Stanford University OHS
Omer Salem, Cairo University
Nicy Scaria, Indian Institute of Science
Nils-Jonathan Schaller, Leibniz Institute for Science and Mathematics Education
Veronica Schmalz, KULeuven
Stephanie Schoch, University of Virginia
Matthew Shardlow, Manchester Metropolitan University
Mayank Sharma, Graduate Student, Stanford University
Kevin Shi, University of California, Berkeley
Mariana Shimabukuro, Ontario Tech University
Hyo Jeong Shin, Sogang University
Gyu-Ho Shin, University of Illinois Chicago
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)
Mayank Soni, ADAPT Centre, Trinity College Dublin
Alexey Sorokin, Moscow State University
Anna Sotnikova, EPFL
KV Aditya Srivatsa, MBZUAI
Maja Stahl, Leibniz University Hannover
Felix Stahlberg, Google Research
Katherine Stasaski, Salesforce Research
Helmer Strik, Centre for Language and Speech Technology (CLST), Centre for Language Studies (CLS), Radboud University Nijmegen
David Strohmaier, University of Cambridge
Hakyung Sung, University of Oregon
Abhijit Suresh, Graduate Student
Andreas Säuberli, LMU Munich
Chuangchuang Tan, Beijing Jiaotong University
CheeWei Tan, Nanyang Technological University
Wenjia Tan, University of Macau
Nhat Tran, University of Pittsburgh
Felipe Urrutia, Center for Advanced Research in Education
Masaki Uto, The University of Electro-Communications
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
Giulia Venturi, Institute of Computational Linguistics Antonio Zampolli"(ILC-CNR)
Anthony Verardi, Duolingo
Amit Verma, Guvi Geek Network
Elena Volodina, University of Gothenburg
Anh-Duc Vu, University of Helsinki
Deliang Wang, The University of Hong Kong
Nikhil Wani, University of Southern California
Taro Watanabe, Nara Institute of Science and Technology
Yuchen Wei, Pennsylvania State University
Alistair Willis, The Open University
Steven Wilson, University of Michigan-Flint
Anna Winklerova, Faculty of Informatics Masaryk University
Hanna Woloszyn, University of Cologne
Simon Woodhead, Eedi
Anna Wroblewska, Faculty of Mathematics and Information Science, Warsaw University of Technology
Changrong Xiao, Tsinghua University
Hiroaki Yamada, Institute of Science Tokyo
Haiyin Yang, University of Florida
Roman Yangarber, University of Helsinki
Sahar Yarmohammadtoosky, NBME
Hanling Yi, Intellifusion, Inc.
Su-Youn Yoon, EduLab
Marcos Zampieri, George Mason University
Alessandra Zarcone, Technische Hochschule Augsburg
Fabian Zehner, DIPF | Leibniz Institute for Research and Information in Education
Kamyar Zeinalipour, University of Siena
Torsten Zesch, Computational Linguistics, FernUniversität in Hagen
Franklin Zhang, Bellevue College
Mike Zhang, Aalborg University
Jing Zhang, Emory University
Yiling Zhao, Stanford University
Yang Zhong, University of Pittsburgh
Yiyun Zhou, NBME
Ej Zhou, University of Cambridge
Jessica Zipf, University of Konstanz
Michael Zock, CNRS-LIS
Leonidas Zotos, University of Groningen
Bowei Zou, Institute for Infocomm Research
Robert Östling, Department of Linguistics, Stockholm University

Keynote Talk

How LLMs Are Reshaping GEC: Training, Evaluation, and Task Framing

Kostiantyn Omelianchuk
Grammarly

Abstract: This keynote will explore the evolving role of Large Language Models (LLMs) in training and evaluating Grammatical Error Correction (GEC) systems, using Grammarly as a case study. It will cover the shift from primarily using human-annotated corpora to semi-synthetic data generation approaches, examining its impact on model training, evaluation practices, and overall task definition. Key topics include task definition challenges, trade-offs between data types, observed biases in models, and recent advances in LLM-based evaluation techniques. The talk will also explore scalable approaches for multilingual GEC and outline implications for future research.

Bio: Kostiantyn Omelianchuk is an Applied Research Scientist and Area Tech Lead at Grammarly, where he works on practical applications of NLP, with a primary interest in Grammatical Error Correction (GEC). He has over nine years of experience in the field and has co-authored several papers, including GECToR: Grammatical Error Correction – Tag, Not Rewrite, a widely used approach in the GEC community. His research explores edit-based modeling, the use of large language models for text correction and simplification, and the transition from human-annotated to synthetic data for training and evaluation. His recent work focuses on multilingual GEC, LLM-based evaluation methods, and synthetic data generation.

Table of Contents

Large Language Models for Education: Understanding the Needs of Stakeholders, Current Capabilities and the Path Forward

Sankalan Pal Chowdhury, Nico Daheim, Ekaterina Kochmar, Jakub Macina, Donya Rooein, Mrinmaya Sachan and Shashank Sonkar 1

Comparing human and LLM proofreading in L2 writing: Impact on lexical and syntactic features

Hakyung Sung, Karla Csuros and Min-Chang Sung 11

MateInfoUB: A Real-World Benchmark for Testing LLMs in Competitive, Multilingual, and Multimodal Educational Tasks

Marius Dumitran, Mihnea Buca and Theodor Moroianu 24

Unsupervised Automatic Short Answer Grading and Essay Scoring: A Weakly Supervised Explainable Approach

Felipe Urrutia, Cristian Buc, Roberto Araya and Valentin Barriere 38

A Survey on Automated Distractor Evaluation in Multiple-Choice Tasks

Luca Benedetto, Shiva Taslimipoor and Paula Buttery 55

Alignment Drift in CEFR-prompted LLMs for Interactive Spanish Tutoring

Mina Almasi and Ross Kristensen-McLachlan 70

Leveraging Generative AI for Enhancing Automated Assessment in Programming Education Contests

Stefan Dascalescu, Marius Dumitran and Mihai Alexandru Vasiluta 89

Can LLMs Effectively Simulate Human Learners? Teachers' Insights from Tutoring LLM Students

Daria Martynova, Jakub Macina, Nico Daheim, Nilay Yalcin, Xiaoyu Zhang and Mrinmaya Sachan 100

Adapting LLMs for Minimal-edit Grammatical Error Correction

Ryszard Staruch, Filip Gralinski and Daniel Dzienisiewicz 118

COGENT: A Curriculum-oriented Framework for Generating Grade-appropriate Educational Content

Zhengyuan Liu, Stella Xin Yin, Dion Hoe-Lian Goh and Nancy Chen 129

Is Lunch Free Yet? Overcoming the Cold-Start Problem in Supervised Content Scoring using Zero-Shot LLM-Generated Training Data

Marie Bexte and Torsten Zesch 144

Transformer Architectures for Vocabulary Test Item Difficulty Prediction

Lucy Skidmore, Mariano Felice and Karen Dunn 160

Automatic concept extraction for learning domain modeling: A weakly supervised approach using contextualized word embeddings

Kordula De Kuthy, Leander Gurrbach and Detmar Meurers 175

Towards a Real-time Swedish Speech Analyzer for Language Learning Games: A Hybrid AI Approach to Language Assessment

Tianyi Geng and David Alfter 186

Multilingual Grammatical Error Annotation: Combining Language-Agnostic Framework with Language-Specific Flexibility

Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, SILIANG LIU and Jungyeul Park 202

<i>LLM-based post-editing as reference-free GEC evaluation</i> Robert Östling, Murathan Kurfali and Andrew Caines	213
<i>Increasing the Generalizability of Similarity-Based Essay Scoring Through Cross-Prompt Training</i> Marie Bexte, Yuning Ding and Andrea Horbach	225
<i>Automated Scoring of a German Written Elicited Imitation Test</i> Mihail Chifligarov, Jammila Laâguidi, Max Schellenberg, Alexander Dill, Anna Timukova, Anastasia Drackert and Ronja Laarmann-Quante	237
<i>LLMs Protégés: Tutoring LLMs with Knowledge Gaps Improves Student Learning Outcome</i> Andrei Kucharavy, Cyril Vallez and Dimitri Percia David	248
<i>LEVOS: Leveraging Vocabulary Overlap with Sanskrit to Generate Technical Lexicons in Indian Languages</i> Karthika N J, Krishnakant Bhatt, Ganesh Ramakrishnan and Preethi Jyothi	258
<i>Do LLMs Give Psychometrically Plausible Responses in Educational Assessments?</i> Andreas Säuberli, Diego Frassinelli and Barbara Plank	266
<i>Challenges for AI in Multimodal STEM Assessments: a Human-AI Comparison</i> Aymeric de Chillaz, Anna Sotnikova, Patrick Jermann and Antoine Bosselut	279
<i>LookAlike: Consistent Distractor Generation in Math MCQs</i> Nisarg Parikh, Alexander Scarlatos, Nigel Fernandez, Simon Woodhead and Andrew Lan ...	294
<i>You Shall Know a Word's Difficulty by the Family It Keeps: Word Family Features in Personalised Word Difficulty Classifiers for L2 Spanish</i> Jasper Degraeuwe	312
<i>The Need for Truly Graded Lexical Complexity Prediction</i> David Alfter	326
<i>Towards Automatic Formal Feedback on Scientific Documents</i> Louise Bloch, Johannes Rückert and Christoph Friedrich	334
<i>Don't Score too Early! Evaluating Argument Mining Models on Incomplete Essays</i> Nils-Jonathan Schaller, Yuning Ding, Thorben Jansen and Andrea Horbach	345
<i>Educators' Perceptions of Large Language Models as Tutors: Comparing Human and AI Tutors in a Blind Text-only Setting</i> Sankalan Pal Chowdhury, Terry Jingchen Zhang, Donya Rooein, Dirk Hovy, Tanja Käser and Mrinmaya Sachan	356
<i>Transformer-Based Real-Word Spelling Error Feedback with Configurable Confusion Sets</i> Torsten Zesch, Dominic Gardner and Marie Bexte	375
<i>Automated L2 Proficiency Scoring: Weak Supervision, Large Language Models, and Statistical Guarantees</i> Aitor Arronte Alvarez and Naiyi Xie Fincham	384
<i>Automatic Generation of Inference Making Questions for Reading Comprehension Assessments</i> Wanjing (Anya) Ma, Michael Flor and Zuwei Wang	398
<i>Investigating Methods for Mapping Learning Objectives to Bloom's Revised Taxonomy in Course Descriptions for Higher Education</i> Zahra Kolagar, Frank Zalkow and Alessandra Zarcone	415

<i>LangEye: Toward 'Anytime' Learner-Driven Vocabulary Learning From Real-World Objects</i> Mariana Shimabukuro, Deval Panchal and Christopher Collins	446
<i>Costs and Benefits of AI-Enabled Topic Modeling in P-20 Research: The Case of School Improvement Plans</i> Syeda Sabrina Akter, Seth Hunter, David Woo and Antonios Anastasopoulos	460
<i>Advances in Auto-Grading with Large Language Models: A Cross-Disciplinary Survey</i> Tania Amanda Nkoyo Frederick Eneye, Chukwuebuka Fortunate Ijezue, Ahmad Imam Amjad, Maaz Amjad, Sabur Butt and Gerardo Castañeda-Garza	477
<i>Unsupervised Sentence Readability Estimation Based on Parallel Corpora for Text Simplification</i> Rina Miyata, Toru Urakawa, Hideaki Tamori and Tomoyuki Kajiwara	499
<i>From End-Users to Co-Designers: Lessons from Teachers</i> Martina Galletti and Valeria Cesaroni	505
<i>LLMs in alliance with Edit-based models: advancing In-Context Learning for Grammatical Error Correction by Specific Example Selection</i> Alexey Sorokin and Regina Nasyrova	517
<i>Explaining Holistic Essay Scores in Comparative Judgment Assessments by Predicting Scores on Rubrics</i> Michiel De Vrindt, Renske Bouwer, Wim Van Den Noortgate, Marije Lesterhuis and Anaïs Tack	535
<i>Enhancing Arabic Automated Essay Scoring with Synthetic Data and Error Injection</i> Chatrine Qwaider, Bashar Alhafni, Kirill Chirkunov, Nizar Habash and Ted Briscoe	549
<i>Direct Repair Optimization: Training Small Language Models For Educational Program Repair Improves Feedback</i> Charles Koutchme, Nicola Dainese and Arto Hellas	564
<i>Analyzing Interview Questions via Bloom's Taxonomy to Enhance the Design Thinking Process</i> Fateme Kazemi Vanhari, Christopher Anand and Charles Welch	582
<i>Estimation of Text Difficulty in the Context of Language Learning</i> Anisia Katinskaia, Anh-Duc Vu, Jue Hou, Ulla Vanhatalo, Yiheng Wu and Roman Yangarber	594
<i>Are Large Language Models for Education Reliable Across Languages?</i> Vansh Gupta, Sankalan Pal Chowdhury, Vilém Zouhar, Donya Rooein and Mrinmaya Sachan	612
<i>Exploiting the English Vocabulary Profile for L2 word-level vocabulary assessment with LLMs</i> Stefano Banno, Kate Knill and Mark Gales	632
<i>Advancing Question Generation with Joint Narrative and Difficulty Control</i> Bernardo Leite and Henrique Lopes Cardoso	647
<i>Down the Cascades of Omethi: Hierarchical Automatic Scoring in Large-Scale Assessments</i> Fabian Zehner, Hyo Jeong Shin, Emily Kerzabi, Andrea Horbach, Sebastian Gombert, Frank Goldhammer, Torsten Zesch and Nico Andersen	660
<i>Lessons Learned in Assessing Student Reflections with LLMs</i> Mohamed Elaraby and Diane Litman	672
<i>Using NLI to Identify Potential Collocation Transfer in L2 English</i> Haiyin Yang, Zoey Liu and Stefanie Wulff	687

<i>Name of Thrones: How Do LLMs Rank Student Names in Status Hierarchies Based on Race and Gender?</i>	
Annabella Sakunkoo and Jonathan Sakunkoo	697
<i>Exploring LLM-Based Assessment of Italian Middle School Writing: A Pilot Study</i>	
Adriana Mirabella and Dominique Brunato	708
<i>Exploring task formulation strategies to evaluate the coherence of classroom discussions with GPT-4o</i>	
Yuya Asano, Beata Beigman Klebanov and Jamie Mikeska	716
<i>A Bayesian Approach to Inferring Prerequisite Structures and Topic Difficulty in Language Learning</i>	
Anh-Duc Vu, Jue Hou, Anisia Katinskaia, Ching-Fan Sheu and Roman Yangarber	737
<i>Improving In-context Learning Example Retrieval for Classroom Discussion Assessment with Re-ranking and Label Ratio Regulation</i>	
Nhat Tran, Diane Litman, Benjamin Pierce, Richard Correnti and Lindsay Clare Matsumura .	752
<i>Exploring LLMs for Predicting Tutor Strategy and Student Outcomes in Dialogues</i>	
Fareya Ikram, Alexander Scarlatos and Andrew Lan	765
<i>Assessing Critical Thinking Components in Romanian Secondary School Textbooks: A Data Mining Approach to the ROTEX Corpus</i>	
Madalina Chitez, Liviu Dinu, Marius Micluta-Campeanu, Ana-Maria Bucur and Roxana Rogobete	780
<i>Improving AI assistants embedded in short e-learning courses with limited textual content</i>	
Jacek Marciniak, Marek Kubis, Michał Gulczyński, Adam Szpilkowski, Adam Wiczarek and Marcin Szczepański	794
<i>Beyond Linear Digital Reading: An LLM-Powered Concept Mapping Approach for Reducing Cognitive Load</i>	
Junzhi Han and Jinho D. Choi	805
<i>GermDetect: Verb Placement Error Detection Datasets for Learners of Germanic Languages</i>	
Noah-Manuel Michael and Andrea Horbach	818
<i>Enhancing Security and Strengthening Defenses in Automated Short-Answer Grading Systems</i>	
Sahar Yarmohammadtoosky, Yiyun Zhou, Victoria Yaneva, Peter Baldwin, Saed Rezayi, Brian Clauser and Polina Harik	830
<i>EyeLLM: Using Lookback Fixations to Enhance Human-LLM Alignment for Text Completion</i>	
Astha Singh, Mark Torrance and Evgeny Chukharev	841
<i>Span Labeling with Large Language Models: Shell vs. Meat</i>	
Phoebe Mulcaire and Nitin Madnani	850
<i>Intent Matters: Enhancing AI Tutoring with Fine-Grained Pedagogical Intent Annotation</i>	
Kseniia Petukhova and Ekaterina Kochmar	860
<i>Comparing Behavioral Patterns of LLM and Human Tutors: A Population-level Analysis with the CIMA Dataset</i>	
Aayush Kucheria, Nitin Sawhney and Arto Hellas	873
<i>Temporalizing Confidence: Evaluation of Chain-of-Thought Reasoning with Signal Temporal Logic</i>	
Zhenjiang Mao, Artem Bisliouk, Rohith Nama and Ivan Ruchkin	882

<i>Automated Scoring of Communication Skills in Physician-Patient Interaction: Balancing Performance and Scalability</i>	
Saed Rezayi, Le An Ha, Yiyun Zhou, Andrew Houriet, Angelo D’Addario, Peter Baldwin, Polina Harik, Ann King and Victoria Yaneva	891
<i>Decoding Actionability: A Computational Analysis of Teacher Observation Feedback</i>	
Mayank Sharma and Jason Zhang	898
<i>EduCSW: Building a Mandarin-English Code-Switched Generation Pipeline for Computer Science Learning</i>	
Ruishi Chen and Yiling Zhao	908
<i>STAIR-AIG: Optimizing the Automated Item Generation Process through Human-AI Collaboration for Critical Thinking Assessment</i>	
Euigyum Kim, Seewoo Li, Salah Khalil and Hyo Jeong Shin	920
<i>UPSC2M: Benchmarking Adaptive Learning from Two Million MCQ Attempts</i>	
Kevin Shi and Karttikeya Mangalam	931
<i>Can GPTZero’s AI Vocabulary Distinguish Between LLM-Generated and Student-Written Essays?</i>	
Veronica Schmalz and Anaïs Tack	937
<i>Paragraph-level Error Correction and Explanation Generation: Case Study for Estonian</i>	
Martin Vainikko, Taavi Kamarik, Karina Kert, Krista Liin, Silvia Maine, Kais Allkivi, Annekatrin Kaivapalu and Mark Fishel	953
<i>End-to-End Automated Item Generation and Scoring for Adaptive English Writing Assessment with Large Language Models</i>	
Kamel Nebhi, Amrita Panesar and Hans Bantilan	968
<i>A Framework for Proficiency-Aligned Grammar Practice in LLM-Based Dialogue Systems</i>	
Luisa Ribeiro-Flucht, Xiaobin Chen and Detmar Meurers	978
<i>Can LLMs Reliably Simulate Real Students’ Abilities in Mathematics and Reading Comprehension?</i>	
KV Aditya Srivatsa, Kaushal Maurya and Ekaterina Kochmar	988
<i>LLM-Assisted, Iterative Curriculum Writing: A Human-Centered AI Approach in Finnish Higher Education</i>	
Leo Huovinen and Mika Hämäläinen	1002
<i>Findings of the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered Tutors</i>	
Ekaterina Kochmar, Kaushal Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack and Justin Vasselli	1011
<i>Jinan Smart Education at BEA 2025 Shared Task: Dual Encoder Architecture for Tutor Identification via Semantic Understanding of Pedagogical Conversations</i>	
Lei Chen	1034
<i>Wonderland_EDU@HKU at BEA 2025 Shared Task: Fine-tuning Large Language Models to Evaluate the Pedagogical Ability of AI-powered Tutors</i>	
Deliang Wang, Chao Yang and Gaowei Chen	1040
<i>bea-jh at BEA 2025 Shared Task: Evaluating AI-powered Tutors through Pedagogically-Informed Reasoning</i>	
Jihyeon Roh and Jinhyun Bang	1049

<i>CU at BEA 2025 Shared Task: A BERT-Based Cross-Attention Approach for Evaluating Pedagogical Responses in Dialogue</i>	
Zhihao Lyu	1060
<i>BJTU at BEA 2025 Shared Task: Task-Aware Prompt Tuning and Data Augmentation for Evaluating AI Math Tutors</i>	
Yuming Fan, Chuangchuang Tan and Wenyu Song	1073
<i>SYSUpporter Team at BEA 2025 Shared Task: Class Compensation and Assignment Optimization for LLM-generated Tutor Identification</i>	
Longfeng Chen, Zeyu Huang, Zheng Xiao, Yawen Zeng and Jin Xu	1078
<i>BLCU-ICALL at BEA 2025 Shared Task: Multi-Strategy Evaluation of AI Tutors</i>	
Jiyuan An, Xiang Fu, Bo Liu, Xuquan Zong, Cunliang Kong, Shuliang Liu, Shuo Wang, Zhenghao Liu, Liner Yang, Hanghang Fan and Erhong Yang	1084
<i>Phaedrus at BEA 2025 Shared Task: Assessment of Mathematical Tutoring Dialogues through Tutor Identity Classification and Actionability Evaluation</i>	
Rajneesh Tiwari and pranshu rastogi	1098
<i>Emergent Wisdom at BEA 2025 Shared Task: From Lexical Understanding to Reflective Reasoning for Pedagogical Ability Assessment</i>	
Raunak Jain and Srinivasan Rengarajan	1108
<i>Averroes at BEA 2025 Shared Task: Verifying Mistake Identification in Tutor, Student Dialogue</i>	
Mazen Yasser, Mariam Saeed, Hossam Elkordi and Ayman Khalafallah	1121
<i>SmolLab_SEU at BEA 2025 Shared Task: A Transformer-Based Framework for Multi-Track Pedagogical Evaluation of AI-Powered Tutors</i>	
Md. Abdur Rahman, MD AL AMIN, Sabik Aftahee, Muhammad Junayed and Md Ashiqur Rahman	1127
<i>RETUYT-INCO at BEA 2025 Shared Task: How Far Can Lightweight Models Go in AI-powered Tutor Evaluation?</i>	
Santiago Góngora, Ignacio Sastre, Santiago Robaina, Ignacio Remersaro, Luis Chiruzzo and Aiala Rosá	1135
<i>K-NLPers at BEA 2025 Shared Task: Evaluating the Quality of AI Tutor Responses with GPT-4.1</i>	
Geon Park, Jiwoo Song, Gihyeon Choi, Juoh Sun and Harksoo Kim	1145
<i>Henry at BEA 2025 Shared Task: Improving AI Tutor’s Guidance Evaluation Through Context-Aware Distillation</i>	
Henry Pit	1164
<i>TBA at BEA 2025 Shared Task: Transfer-Learning from DARE-TIES Merged Models for the Pedagogical Ability Assessment of LLM-Powered Math Tutors</i>	
Sebastian Gombert, Fabian Zehner and Hendrik Drachsler	1173
<i>LexiLogic at BEA 2025 Shared Task: Fine-tuning Transformer Language Models for the Pedagogical Skill Evaluation of LLM-based tutors</i>	
Souvik Bhattacharyya, Billodal Roy, Niranjan M and Pranav Gupta	1180
<i>IALab UC at BEA 2025 Shared Task: LLM-Powered Expert Pedagogical Feature Extraction</i>	
Sofía Correa Busquets, Valentina Córdova Véliz and Jorge Baier	1187

<i>MSA at BEA 2025 Shared Task: Disagreement-Aware Instruction Tuning for Multi-Dimensional Evaluation of LLMs as Math Tutors</i>	
Baraa Hikal, Mohmaed Basem, Islam Oshallah and Ali Hamdi	1194
<i>TutorMind at BEA 2025 Shared Task: Leveraging Fine-Tuned LLMs and Data Augmentation for Mistake Identification</i>	
FATIMA DEKMAK, Christian Khairallah and Wissam Antoun	1203
<i>Two Outliers at BEA 2025 Shared Task: Tutor Identity Classification using DiReC, a Two-Stage Disentangled Contrastive Representation</i>	
Eduardus Tjitrahardja and Ikhlasul Hanif	1212
<i>Archaeology at BEA 2025 Shared Task: Are Simple Baselines Good Enough?</i>	
Ana Roşu, Jany-Gabriel Ispas and Sergiu Nisioi	1224
<i>NLIP at BEA 2025 Shared Task: Evaluation of Pedagogical Ability of AI Tutors</i>	
Trishita Saha, Shrenik Ganguli and Maunendra Sankar Desarkar	1242
<i>NeuralNexus at BEA 2025 Shared Task: Retrieval-Augmented Prompting for Mistake Identification in AI Tutors</i>	
Numaan Naeem, Sarfraz Ahmad, Momina Ahsan and Hasan Iqbal	1254
<i>DLSU at BEA 2025 Shared Task: Towards Establishing Baseline Models for Pedagogical Response Evaluation Tasks</i>	
Maria Monica Manlises, Mark Edward Gonzales and Lanz Lim	1260
<i>BD at BEA 2025 Shared Task: MPNet Ensembles for Pedagogical Mistake Identification and Localization in AI Tutor Responses</i>	
Shadman Rohan, Ishita Sur Apan, Muhtasim Shochcho, Md Fahim, Mohammad Rahman, AKM Mahbubur Rahman and Amin Ali	1266
<i>Thapar Titan/s : Fine-Tuning Pretrained Language Models with Contextual Augmentation for Mistake Identification in Tutor–Student Dialogues</i>	
Harsh Dadwal, Sparsh Rastogi and Jatin Bedi	1278

Program

Thursday, July 31, 2025

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

10:30 - 11:00 *Coffee Break*

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

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

14:00 - 15:30 *Oral Session A*

A Bayesian Approach to Inferring Prerequisite Structures and Topic Difficulty in Language Learning

Anh-Duc Vu, Jue Hou, Anisia Katinskaia, Ching-Fan Sheu and Roman Yangarber

Enhancing Arabic Automated Essay Scoring with Synthetic Data and Error Injection

Chatrine Qwaider, Bashar Alhafni, Kirill Chirkunov, Nizar Habash and Ted Briscoe

Alignment Drift in CEFR-prompted LLMs for Interactive Spanish Tutoring

Mina Almasi and Ross Kristensen-McLachlan

You Shall Know a Word's Difficulty by the Family It Keeps: Word Family Features in Personalised Word Difficulty Classifiers for L2 Spanish

Jasper Degraeuwe

Assessing Critical Thinking Components in Romanian Secondary School Textbooks: A Data Mining Approach to the ROTEX Corpus

Madalina Chitez, Liviu Dinu, Marius Micluta-Campeanu, Ana-Maria Bucur and Roxana Rogobete

Unsupervised Automatic Short Answer Grading and Essay Scoring: A Weakly Supervised Explainable Approach

Felipe Urrutia, Cristian Buc, Roberto Araya and Valentin Barriere

15:30 - 16:00 *Coffee Break*

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

Thursday, July 31, 2025 (continued)

A Survey on Automated Distractor Evaluation in Multiple-Choice Tasks

Luca Benedetto, Shiva Taslimipour and Paula Buttery

Increasing the Generalizability of Similarity-Based Essay Scoring Through Cross-Prompt Training

Marie Bexte, Yuning Ding and Andrea Horbach

Automatic concept extraction for learning domain modeling: A weakly supervised approach using contextualized word embeddings

Kordula De Kuthy, Leander Gurrbach and Detmar Meurers

Automated Scoring of a German Written Elicited Imitation Test

Mihail Chifligarov, Jammila Laâguidi, Max Schellenberg, Alexander Dill, Anna Timukova, Anastasia Drackert and Ronja Laarmann-Quante

Challenges for AI in Multimodal STEM Assessments: a Human-AI Comparison

Aymeric de Chillaz, Anna Sotnikova, Patrick Jermann and Antoine Bosselut

Don't Score too Early! Evaluating Argument Mining Models on Incomplete Essays

Nils-Jonathan Schaller, Yuning Ding, Thorben Jansen and Andrea Horbach

LangEye: Toward 'Anytime' Learner-Driven Vocabulary Learning From Real-World Objects

Mariana Shimabukuro, Deval Panchal and Christopher Collins

Explaining Holistic Essay Scores in Comparative Judgment Assessments by Predicting Scores on Rubrics

Michiel De Vrindt, Renske Bouwer, Wim Van Den Noortgate, Marije Lesterhuis and Anaïs Tack

Name of Thrones: How Do LLMs Rank Student Names in Status Hierarchies Based on Race and Gender?

Annabella Sakunkoo and Jonathan Sakunkoo

Enhancing Security and Strengthening Defenses in Automated Short-Answer Grading Systems

Sahar Yarmohammadtoosky, Yiyun Zhou, Victoria Yaneva, Peter Baldwin, Saed Rezayi, Brian Clauser and Polina Harik

EduCSW: Building a Mandarin-English Code-Switched Generation Pipeline for Computer Science Learning

Ruishi Chen and Yiling Zhao

Thursday, July 31, 2025 (continued)

Paragraph-level Error Correction and Explanation Generation: Case Study for Estonian

Martin Vainikko, Taavi Kamarik, Karina Kert, Krista Liin, Silvia Maine, Kais Allkivi, Annekatrin Kaivapalu and Mark Fishel

Can LLMs Reliably Simulate Real Students' Abilities in Mathematics and Reading Comprehension?

KV Aditya Srivatsa, Kaushal Maurya and Ekaterina Kochmar

Transformer Architectures for Vocabulary Test Item Difficulty Prediction

Lucy Skidmore, Mariano Felice and Karen Dunn

Comparing human and LLM proofreading in L2 writing: Impact on lexical and syntactic features

Hakyung Sung, Karla Csuros and Min-Chang Sung

MateInfoUB: A Real-World Benchmark for Testing LLMs in Competitive, Multilingual, and Multimodal Educational Tasks

Marius Dumitran, Mihnea Buca and Theodor Moroianu

Investigating Methods for Mapping Learning Objectives to Bloom's Revised Taxonomy in Course Descriptions for Higher Education

Zahra Kolagar, Frank Zalkow and Alessandra Zarcone

Using NLI to Identify Potential Collocation Transfer in L2 English

Haiyin Yang, Zoey Liu and Stefanie Wulff

Improving In-context Learning Example Retrieval for Classroom Discussion Assessment with Re-ranking and Label Ratio Regulation

Nhat Tran, Diane Litman, Benjamin Pierce, Richard Correnti and Lindsay Clare Matsumura

Comparing Behavioral Patterns of LLM and Human Tutors: A Population-level Analysis with the CIMA Dataset

Aayush Kucheria, Nitin Sawhney and Arto Hellas

UPSC2M: Benchmarking Adaptive Learning from Two Million MCQ Attempts

Kevin Shi and Karttikeya Mangalam

Multilingual Grammatical Error Annotation: Combining Language-Agnostic Framework with Language-Specific Flexibility

Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, SILIANG LIU and Jungyeul Park

Thursday, July 31, 2025 (continued)

Automatic Generation of Inference Making Questions for Reading Comprehension Assessments

Wanjing (Anyu) Ma, Michael Flor and Zuowei Wang

Lessons Learned in Assessing Student Reflections with LLMs

Mohamed Elaraby and Diane Litman

Automated L2 Proficiency Scoring: Weak Supervision, Large Language Models, and Statistical Guarantees

Aitor Arronte Alvarez and Naiyi Xie Fincham

Advances in Auto-Grading with Large Language Models: A Cross-Disciplinary Survey

Tania Amanda Nkoyo Frederick Eneye, Chukwuebuka Fortunate Ijezue, Ahmad Imam Amjad, Maaz Amjad, Sabur Butt and Gerardo Castañeda-Garza

Exploring LLMs for Predicting Tutor Strategy and Student Outcomes in Dialogues

Fareya Ikram, Alexander Scarlatos and Andrew Lan

Temporalizing Confidence: Evaluation of Chain-of-Thought Reasoning with Signal Temporal Logic

Zhenjiang Mao, Artem Bisliouk, Rohith Nama and Ivan Ruchkin

18:00 - 21:00 *Workshop Dinner*

Friday, August 1, 2025

09:00 - 09:45 *Keynote Talk by Kostia Omelianchuk*

09:45 - 10:30 *Oral Session B*

LLMs in alliance with Edit-based models: advancing In-Context Learning for Grammatical Error Correction by Specific Example Selection
Alexey Sorokin and Regina Nasyrova

Findings of the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered Tutors
Ekaterina Kochmar, Kaushal Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack and Justin Vasselli

MSA at BEA 2025 Shared Task: Disagreement-Aware Instruction Tuning for Multi-Dimensional Evaluation of LLMs as Math Tutors
Baraa Hikal, Mohmaed Basem, Islam Oshallah and Ali Hamdi

10:30 - 11:00 *Coffee Break*

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

Leveraging Generative AI for Enhancing Automated Assessment in Programming Education Contests
Stefan Dascalescu, Marius Dumitran and Mihai Alexandru Vasiluta

Is Lunch Free Yet? Overcoming the Cold-Start Problem in Supervised Content Scoring using Zero-Shot LLM-Generated Training Data
Marie Bexte and Torsten Zesch

Towards a Real-time Swedish Speech Analyzer for Language Learning Games: A Hybrid AI Approach to Language Assessment
Tianyi Geng and David Alfter

LEVOS: Leveraging Vocabulary Overlap with Sanskrit to Generate Technical Lexicons in Indian Languages
Karthika N J, Krishnakant Bhatt, Ganesh Ramakrishnan and Preethi Jyothi

The Need for Truly Graded Lexical Complexity Prediction
David Alfter

Friday, August 1, 2025 (continued)

Educators' Perceptions of Large Language Models as Tutors: Comparing Human and AI Tutors in a Blind Text-only Setting

Sankalan Pal Chowdhury, Terry Jingchen Zhang, Donya Rooein, Dirk Hovy, Tanja Käser and Mrinmaya Sachan

Costs and Benefits of AI-Enabled Topic Modeling in P-20 Research: The Case of School Improvement Plans

Syeda Sabrina Akter, Seth Hunter, David Woo and Antonios Anastasopoulos

Are Large Language Models for Education Reliable Across Languages?

Vansh Gupta, Sankalan Pal Chowdhury, Vilém Zouhar, Donya Rooein and Mrinmaya Sachan

Span Labeling with Large Language Models: Shell vs. Meat

Phoebe Mulcaire and Nitin Madnani

STAIR-AIG: Optimizing the Automated Item Generation Process through Human-AI Collaboration for Critical Thinking Assessment

Euigyum Kim, Seewoo Li, Salah Khalil and Hyo Jeong Shin

End-to-End Automated Item Generation and Scoring for Adaptive English Writing Assessment with Large Language Models

Kamel Nebhi, Amrita Panesar and Hans Bantilan

bea-jh at BEA 2025 Shared Task: Evaluating AI-powered Tutors through Pedagogically-Informed Reasoning

Jihyeon Roh and Jinhyun Bang

K-NLPers at BEA 2025 Shared Task: Evaluating the Quality of AI Tutor Responses with GPT-4.1

Geon Park, Jiwoo Song, Gihyeon Choi, Juoh Sun and Harksoo Kim

IALab UC at BEA 2025 Shared Task: LLM-Powered Expert Pedagogical Feature Extraction

Sofía Correa Busquets, Valentina Córdova Véliz and Jorge Baier

TBA at BEA 2025 Shared Task: Transfer-Learning from DARE-TIES Merged Models for the Pedagogical Ability Assessment of LLM-Powered Math Tutors

Sebastian Gombert, Fabian Zehner and Hendrik Drachler

COGENT: A Curriculum-oriented Framework for Generating Grade-appropriate Educational Content

Zhengyuan Liu, Stella Xin Yin, Dion Hoe-Lian Goh and Nancy Chen

Friday, August 1, 2025 (continued)

Analyzing Interview Questions via Bloom's Taxonomy to Enhance the Design Thinking Process

Fatemeh Kazemi Vanhari, Christopher Anand and Charles Welch

Exploring LLM-Based Assessment of Italian Middle School Writing: A Pilot Study

Adriana Mirabella and Dominique Brunato

Beyond Linear Digital Reading: An LLM-Powered Concept Mapping Approach for Reducing Cognitive Load

Junzhi Han and Jinho D. Choi

BLCU-ICALL at BEA 2025 Shared Task: Multi-Strategy Evaluation of AI Tutors

Jiyuan An, Xiang Fu, Bo Liu, Xuquan Zong, Cunliang Kong, Shuliang Liu, Shuo Wang, Zhenghao Liu, Liner Yang, Hanghang Fan and Erhong Yang

Jinan Smart Education at BEA 2025 Shared Task: Dual Encoder Architecture for Tutor Identification via Semantic Understanding of Pedagogical Conversations

Lei Chen

CU at BEA 2025 Shared Task: A BERT-Based Cross-Attention Approach for Evaluating Pedagogical Responses in Dialogue

Zhihao Lyu

SYSUpporter Team at BEA 2025 Shared Task: Class Compensation and Assignment Optimization for LLM-generated Tutor Identification

Longfeng Chen, Zeyu Huang, Zheng Xiao, Yawen Zeng and Jin Xu

Emergent Wisdom at BEA 2025 Shared Task: From Lexical Understanding to Reflective Reasoning for Pedagogical Ability Assessment

Raunak Jain and Srinivasan Rengarajan

Henry at BEA 2025 Shared Task: Improving AI Tutor's Guidance Evaluation Through Context-Aware Distillation

Henry Pit

TutorMind at BEA 2025 Shared Task: Leveraging Fine-Tuned LLMs and Data Augmentation for Mistake Identification

FATIMA DEKMAK, Christian Khairallah and Wissam Antoun

BD at BEA 2025 Shared Task: MPNet Ensembles for Pedagogical Mistake Identification and Localization in AI Tutor Responses

Shadman Rohan, Ishita Sur Apan, Muhtasim Shochcho, Md Fahim, Mohammad Rahman, AKM Mahbubur Rahman and Amin Ali

Friday, August 1, 2025 (continued)

LLM-Assisted, Iterative Curriculum Writing: A Human-Centered AI Approach in Finnish Higher Education

Leo Huovinen and Mika Hämäläinen

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

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

Can LLMs Effectively Simulate Human Learners? Teachers' Insights from Tutoring LLM Students

Daria Martynova, Jakub Macina, Nico Daheim, Nilay Yalcin, Xiaoyu Zhang and Mrinmaya Sachan

Adapting LLMs for Minimal-edit Grammatical Error Correction

Ryszard Staruch, Filip Gralinski and Daniel Dzienisiewicz

Do LLMs Give Psychometrically Plausible Responses in Educational Assessments?

Andreas Säuberli, Diego Frassinelli and Barbara Plank

Towards Automatic Formal Feedback on Scientific Documents

Louise Bloch, Johannes Rückert and Christoph Friedrich

Transformer-Based Real-Word Spelling Error Feedback with Configurable Confusion Sets

Torsten Zesch, Dominic Gardner and Marie Bexte

Unsupervised Sentence Readability Estimation Based on Parallel Corpora for Text Simplification

Rina Miyata, Toru Urakawa, Hideaki Tamori and Tomoyuki Kajiwara

Exploiting the English Vocabulary Profile for L2 word-level vocabulary assessment with LLMs

Stefano Banno, Kate Knill and Mark Gales

Improving AI assistants embedded in short e-learning courses with limited textual content

Jacek Marciniak, Marek Kubis, Michał Gulczyński, Adam Szpilkowski, Adam Wiczarek and Marcin Szczepański

GermDetect: Verb Placement Error Detection Datasets for Learners of Germanic Languages

Noah-Manuel Michael and Andrea Horbach

Friday, August 1, 2025 (continued)

Automated Scoring of Communication Skills in Physician-Patient Interaction: Balancing Performance and Scalability

Saed Rezayi, Le An Ha, Yiyun Zhou, Andrew Houriet, Angelo D'Addario, Peter Baldwin, Polina Harik, Ann King and Victoria Yaneva

Can GPTZero's AI Vocabulary Distinguish Between LLM-Generated and Student-Written Essays?

Veronica Schmalz and Anaïs Tack

A Framework for Proficiency-Aligned Grammar Practice in LLM-Based Dialogue Systems

Luisa Ribeiro-Flucht, Xiaobin Chen and Detmar Meurers

RETUYT-INCO at BEA 2025 Shared Task: How Far Can Lightweight Models Go in AI-powered Tutor Evaluation?

Santiago Góngora, Ignacio Sastre, Santiago Robaina, Ignacio Remersaro, Luis Chiruzzo and Aiala Rosá

Archaeology at BEA 2025 Shared Task: Are Simple Baselines Good Enough?

Ana Roşu, Jany-Gabriel Ispas and Sergiu Nisioi

NLIP at BEA 2025 Shared Task: Evaluation of Pedagogical Ability of AI Tutors

Trishita Saha, Shrenik Ganguli and Maunendra Sankar Desarkar

LLM-based post-editing as reference-free GEC evaluation

Robert Östling, Murathan Kurfali and Andrew Caines

Estimation of Text Difficulty in the Context of Language Learning

Anisia Katinskaia, Anh-Duc Vu, Jue Hou, Ulla Vanhatalo, Yiheng Wu and Roman Yangarber

Exploring task formulation strategies to evaluate the coherence of classroom discussions with GPT-4o

Yuya Asano, Beata Beigman Klebanov and Jamie Mikeska

EyeLLM: Using Lookback Fixations to Enhance Human-LLM Alignment for Text Completion

Astha Singh, Mark Torrance and Evgeny Chukharev

Decoding Actionability: A Computational Analysis of Teacher Observation Feedback

Mayank Sharma and Jason Zhang

Friday, August 1, 2025 (continued)

Thapar Titan/s : Fine-Tuning Pretrained Language Models with Contextual Augmentation for Mistake Identification in Tutor–Student Dialogues

Harsh Dadwal, Sparsh Rastogi and Jatin Bedi

Wonderland_EDU@HKU at BEA 2025 Shared Task: Fine-tuning Large Language Models to Evaluate the Pedagogical Ability of AI-powered Tutors

Deliang Wang, Chao Yang and Gaowei Chen

BJTU at BEA 2025 Shared Task: Task-Aware Prompt Tuning and Data Augmentation for Evaluating AI Math Tutors

Yuming Fan, Chuangchuang Tan and Wenyu Song

SmolLab_SEU at BEA 2025 Shared Task: A Transformer-Based Framework for Multi-Track Pedagogical Evaluation of AI-Powered Tutors

Md. Abdur Rahman, MD AL AMIN, Sabik Aftahee, Muhammad Junayed and Md Ashiqur Rahman

LexiLogic at BEA 2025 Shared Task: Fine-tuning Transformer Language Models for the Pedagogical Skill Evaluation of LLM-based tutors

Souvik Bhattacharyya, Billodal Roy, Niranjana M and Pranav Gupta

DLSU at BEA 2025 Shared Task: Towards Establishing Baseline Models for Pedagogical Response Evaluation Tasks

Maria Monica Manlises, Mark Edward Gonzales and Lanz Lim

LookAlike: Consistent Distractor Generation in Math MCQs

Nisarg Parikh, Alexander Scarlatos, Nigel Fernandez, Simon Woodhead and Andrew Lan

From End-Users to Co-Designers: Lessons from Teachers

Martina Galletti and Valeria Cesaroni

15:30 - 16:00 *Coffee Break*

16:00 - 17:15 *Oral Session C*

Down the Cascades of Omethi: Hierarchical Automatic Scoring in Large-Scale Assessments

Fabian Zehner, Hyo Jeong Shin, Emily Kerzabi, Andrea Horbach, Sebastian Gombert, Frank Goldhammer, Torsten Zesch and Nico Andersen

Direct Repair Optimization: Training Small Language Models For Educational Program Repair Improves Feedback

Charles Koutchme, Nicola Dainese and Arto Hellas

Friday, August 1, 2025 (continued)

Advancing Question Generation with Joint Narrative and Difficulty Control

Bernardo Leite and Henrique Lopes Cardoso

Intent Matters: Enhancing AI Tutoring with Fine-Grained Pedagogical Intent Annotation

Kseniia Petukhova and Ekaterina Kochmar

LLMs Protégés: Tutoring LLMs with Knowledge Gaps Improves Student Learning Outcome

Andrei Kucharavy, Cyril Vallez and Dimitri Percia David

17:15 - 17:30 *Closing Remarks*

Large Language Models for Education: Understanding the Needs of Stakeholders, Current Capabilities and the Path Forward

Sankalan Pal Chowdhury¹, Nico Daheim², Ekaterina Kochmar³,
Jakub Macina¹, Donya Rooein⁴, Mrinmaya Sachan¹, Shashank Sonkar⁵
¹ETH Zurich, ²TU Darmstadt, ³MBZUAI, ⁴Bocconi University, ⁵Rice University
Alphabetical order of presenters, Correspondence to: mrinmaya.sachan@inf.ethz.ch

Motivation and Objectives: Recent advancements in Large Language Models (LLMs) have opened unprecedented opportunities in education but the current development goals of LLMs stand in contrast to the requirements of educational applications. This tutorial aims to bridge the gap between two major communities: Natural Language Processing (NLP) researchers and Artificial Intelligence in Education (AIED) practitioners. Our objectives are: (1) to help NLP researchers understand the requirements and challenges of education, enabling them to develop LLMs that align with educational needs, and (2) to enable educators and AIED practitioners to gain a deeper understanding of the capabilities and limitations of current NLP technologies, fostering effective integration of LLMs in educational contexts. By facilitating cross-disciplinary dialog, we aim to uncover the potential of LLMs in education.

First, we identify several critical challenges: *LLMs must be aligned to complement established pedagogical theories and educational practices*, incorporating principles such as scaffolding (Macina et al., 2023b; Sonkar et al., 2024a) or Socratic questioning (Shridhar et al., 2022), effective feedback mechanisms (Daheim et al., 2024), and cognitive load management (Settles and Meeder, 2016). This ensures that AI systems enhance rather than undermine learning processes. We emphasize that *LLMs need to be integrated with existing AIED technologies*, including knowledge tracing models and intelligent tutoring systems (ITS). As highlighted by UNESCO (Miao and Cukurova, 2024), we also need to explore *human-AI collaboration* to preserve human agency while leveraging the benefits of LLMs. The use of LLMs also raises ethical concerns about *data privacy and security* and *fairness* for students, necessitating robust safeguards. Finally, *AI literacy* among educators, students, and policymakers is important for ensuring that stakeholders understand their potential and limitations.

1 Tutorial Overview and Structure

1. LLMs meet AIED (60 min)

Intro to LLMs (20 min)
Learning science, AIED foundations (20 min)
Misalignment b/w LLMs & AIED (20 min)

2. Case Studies & Coffee Break (120 min)

Intelligent Tutoring Systems (30 min)
Coffee break (30 min)
Automated feedback & assessment (20 min)
Content (e.g. problem) generation (20 min)
Student modeling and adaptivity (20 min)

3. Closing Discussion (30 min)

LLM development for education
Human, ethical and societal aspects
Closing remarks

We will begin with an introduction of key LLM technologies and AIED usecases, focussing on the needs of stakeholders in education, such as pedagogy, and opportunities to harness LLMs for education applications. Then, we will outline how these needs stand in contrast with current LLM development which instead focusses on solving correctness. Afterwards, we will delve into a series of case studies that highlight how LLMs can be adapted for: (1) robust, personalized, and scalable conversational tutoring systems; (2) adaptive and personalized content generation of educational material, lesson plans, and assessments; (3) grading and delivery of detailed and personalized feedback on student work. We will examine the current capabilities of LLMs in these domains, discussing recent research findings and practical applications. The tutorial will interweave the applications with critical challenges such as pedagogical alignment, ethical considerations, and human factors in AI-assisted education. We finally conclude with a discussion of LLM development for education that emphasizes human, ethical, and societal aspects.

2 LLMs Meet AIED

LLM Training & AIED Requirements LLMs offer significant potential in education but require careful tuning to align with pedagogical goals. For instance, LLMs tend to provide direct answers instead of scaffolding learning which can hinder learning (Macina et al., 2023b; Sonkar et al., 2024a). We will first discuss how LLMs are trained using supervised fine-Tuning (SFT) (Wei et al., 2022), instruction tuning, and reinforcement-learning-based optimization methods (Ziegler et al., 2019; Rafailov et al., 2023). Connected to this, we also highlight the shortcomings of current benchmarks (Hendrycks et al., 2020; Cobbe et al., 2021; Hendrycks et al., 2021) that are used to evaluate LLMs, mainly for solving accuracy. Evaluation of AIED systems is different from this, as pedagogical factors play a large role and have dominated the development of educational systems (Graesser et al., 2005). We highlight these educational needs from different perspectives and show how LLM development goals do not align to them. For example, students require space to think and learn, also by making mistakes (Macina et al., 2023a; Sonkar et al., 2024a), and teachers require flexible student simulations (Markel et al., 2023).

Human Factors & Ethical Considerations: Integrating LLMs into educational contexts brings several human-centered challenges that must be addressed to ensure effective and ethical use. For example, teachers are often not included in the development loop (Shankar et al., 2024), but gaining their trust, also through model explainability (Cortez et al., 2024) is important. We will discuss how instructors can be included effectively, for example, to decide, when and which NLP models to use or which inputs to give to the models. We will also discuss how they can modify the generated outcomes as needed (Lu et al., 2023) and prompt architectures to provide responses to MCQs based on student simulations (Lu and Wang, 2024).

The application of LLMs in schools also raises ethical considerations related to attribution, plagiarism, and the potential for AI-generated content to be presented as original work. To address these issues, universities and educational authorities must strengthen and enforce academic integrity policies while educating students about responsible AI use. Promoting awareness and developing guidelines is essential in maintaining the integrity of academic work in the age of GenAI (Okaiyeto et al., 2023).

3 LLMs for Educational Applications

3.1 Intelligent Tutoring Systems (ITSs)

ITSs have long been the focus of AIED developments including systems such as AutoTutor-based (Nye et al., 2014), example-tracing tutors (Alevan et al., 2009) or Cognitive tutor (Anderson et al., 1997). However, they require extensive human authoring. While LLMs hold great promise to overcome this and enable applications like student tutoring (Chen et al., 2024) or teacher training (Gregorcic et al., 2024; Markel et al., 2023). Yet, they still face limitations, such as generating factually incorrect responses or not offering sufficient pedagogy (Sonkar et al., 2023).

In this tutorial, we will cover a range of works that attempt to alleviate these shortcomings, for example, such that use LLMs within structured dialogs (Schmucker et al., 2024; Pal Chowdhury et al., 2024), data-driven approaches to adding scaffolding capabilities (Macina et al., 2023a; Sonkar et al., 2023; Jurenka and et al., 2024), and mitigating hallucinations by adding intermediate reasoning steps for prompted LLMs (Wang et al., 2024b; Daheim et al., 2024). As large amounts of dialog tutoring data can be hard to collect, we will also discuss synthetic data creation methods (Wang et al., 2024a; Chevalier et al., 2024).

Finally, we will touch upon evaluation protocols that, ideally, should include relevant stakeholders and evaluate learning effectiveness. Such studies include using LLMs in real classrooms, for example, for computer science (Nie et al., 2024) or math education (Cheng et al., 2024), or using LLMs as student simulations to evaluate the effectiveness of automatic dialog tutors (Macina et al., 2023a). Such student simulations can also be effective for teacher training (Gregorcic et al., 2024; Wang and Demszky, 2023) and training teaching assistants (Markel et al., 2023).

3.2 Automated Feedback and Assessment

Hint and Feedback mechanisms play an important role in determining learning outcomes. We will discuss studies that show both the potential and limitations of LLMs in generating quality feedback. (McNichols et al., 2024) show fine-tuned LLMs have limited generalization capabilities. Contrarily, (Dai et al., 2024) find GPT-4 outperforms human instructors in important aspects of effective feedback dimensions such as feeding-up, feeding-forward, and process level. However, student dynamics are

complex; (Nazaretsky et al., 2024) highlights a preference for human-generated feedback when students know its source. We will discuss solutions to overcome these challenges such as reinforcement learning (Scarlatos et al., 2024) and LLM-based student simulation models (Phung et al., 2024).

Another important aspect of feedback is its emotional and motivational impact on students. We will discuss the importance of affective feedback (Li et al., 2024a; Baral et al., 2023). We will also explore how LLMs can be used to provide not just cognitive but also emotional support, offering praise (Thomas et al., 2023) and addressing negative self-talk (Thomas et al., 2024). Additionally, we will touch on ongoing efforts to integrate AI-driven emotional assessment in educational settings (Vistorte et al., 2024) to create empathetic learning environments.

Finally, we'll shift our focus to **automatic assessment**. We will review their performance in Automated Short/Long Answer Grading (Kortemeyer, 2023a; Sonkar et al., 2024b) and Automated Essay Grading (AEG) (Mizumoto and Eguchi, 2023), referencing open-source benchmarks (Ruseti et al., 2024; Dzikovska et al., 2013; Blanchard et al., 2013) for these tasks. Next we will summarize some findings on the real-world deployment of LLMs for grading, which show promise despite certain limitations. We will start with studies on math grading (Morris et al., 2024; Gandolfi, 2024) including those which involve handwritten recognition (Liu et al., 2024a). We will also expand the analysis to other subjects like physics (Kortemeyer, 2023b), computer science (Nilsson and Tuvsedt, 2023), and biology (Mackey et al., 2023) to highlight their capabilities and limitation across domains. We will also explore hybrid grading strategies that incorporate human oversight to enhance reliability (Kaya and Cicekli, 2024).

3.3 Educational Content Generation

LLM-generated content serves teachers (e.g., for curating lessons and exercises) and students (e.g., for writing essays and problem-solving). We will examine studies that use controllable generation to adapt LLMs to diverse learners based on difficulty, grade level, and readability score (Rooein et al., 2023; Kew et al., 2023). We will also discuss LLMs in controlled content generation, focusing on readability scores (Imperial and Tayyar Madabushi, 2023) and novel prompting techniques for difficulty assessment (Rooein et al., 2024).

We will also explore strategies to control and align generated questions with students' abilities, expert requirements, and question taxonomies like Bloom's (Elkins et al., 2024; Hwang et al., 2023). We will mention studies on improving adaptability in question generation (Scaria et al., 2024; Wang et al., 2022) and cover methods like PFQS (Li and Zhang, 2024) for improved control by generating answer outlines before question generation. Evaluation of generated educational questions typically involves expert assessments (Scaria et al., 2024; Biancini et al., 2024), while tools like SQUET (Moore et al., 2024) offer automated quality evaluation. However, challenges remain, as studies show GPT models underperforming in evaluating the pedagogical quality of generated questions (Bulathwela et al., 2023).

Finally, we will also discuss multimodal and multilingual LLMs in education – research has demonstrated the effectiveness of multimodal learning in enhancing educational outcomes, e.g., in science (Bewersdorff et al., 2024). These findings are supported by learning theories emphasizing the cognitive benefits of integrating multiple modes of information, such as combining multimodal representations like text and images (Mayer, 2024).

3.4 Adaptivity and Personalization

In this section, we will discuss personalized learning's potential to address diverse student needs, based on educational theories emphasizing tailored learning experiences. We discuss knowledge space theory (Doignon and Falmagne, 1985), Vygotsky's Zone of Proximal Development (Vygotsky, 1978), and Ebbinghaus's memory model (Ebbinghaus, 1913), which have influenced applications like Duolingo's spaced repetition (Settles and Meeder, 2016) and ETS's assessments (Carlson and von Davier, 2017). We then introduce Knowledge Tracing (KT) techniques, from basic Rasch models (Rasch, 1960) and Item Response Theory (IRT) (Lord, 1980) to advanced Bayesian Knowledge Tracing (Corbett and Anderson, 1994) and Deep Knowledge Tracing (Piech et al., 2015).

Traditionally, KT models have focused on question IDs rather than textual content due to dataset limitations. However, the attention mechanism is well-suited for sequence modeling tasks like knowledge tracing. We will cover models such as MC-QStudentBert (Parsa Neshaei et al., 2024), AKT (Ghosh et al., 2020), SAKT (Pandey and Karypis, 2019), Dtransformer (Yin et al., 2023), and SAINT

(Choi et al., 2020), which leverage attention mechanisms to capture complex relationships between knowledge components and student interactions. The emergence of datasets with auxiliary information, like XES3G5M (Liu et al., 2024b), has facilitated the application of pre-trained LLMs in KT, as explored in works like (Lee et al., 2024).

LLMs have also expanded the scope of KT by enabling adaptive exercise generation (Cui and Sachan, 2023; Srivastava and Goodman, 2021) and domain-specific modifications to transformer architecture, e.g. SparseKT (Huang et al., 2023) which models student behaviors like forgetting (Im et al., 2023). LLMs have also been used in student simulation models like OKT (Liu et al., 2022), which predicts actual student textual responses. Despite these advances, challenges remain, such as LLMs' limited context windows which hinder capturing long-range learning trajectories (Li et al., 2024b).

4 Vision and Path Forward

AI in education offers significant opportunities but requires careful technical, ethical, regulatory, and pedagogical consideration. Requirements include balancing technology with human agency, inclusion, and diversity (Miao and Cukurova, 2024), addressing privacy (Baraniuk, 2024; Leitner et al., 2019; O'Hara and Straus, 2022) and transparency (Holmes et al., 2022), promoting AI literacy (Su et al., 2023; Su and Yang, 2023), but also developing LLMs that meet pedagogical goals. We aim to build a common ground between various stakeholders, namely policymakers, educators, developers, and researchers, which can form a basis for human-centered AI development in education.

5 Diversity & Inclusion considerations

Our tutorial aims to bring together NLP, LS and AIED researchers as well as practitioners. The tutorial is designed to be understandable to an audience with a range of backgrounds. Our group of presenters is made up of diverse backgrounds, seniority-levels, genders, and affiliations.

6 About the Speakers

Sankalan Pal Chowdhury is a second year PhD student in the ETH-EPFL Joint Doctoral Program for Learning Science, advised by Mrinmaya Sachan and Tanja Käser. His research focuses on improving tutoring abilities of LLMs. His work has been published in EMNLP, TACL and L@S.

Nico Daheim is a third year ELLIS PhD student advised by Iryna Gurevych and Mrinmaya Sachan. He works on making LLMs equitable dialog tutors that provide students with personalized opportunities to learn. His works have been published at EMNLP, NAACL, EACL, ICLR and ICML.

Ekaterina Kochmar is an Assistant Professor at the NLP Department at MBZUAI, where she conducts research at the intersection of AI, NLP, and ITSs. She is the current President of SIGEDU and has been involved in organizing BEA since 2013.

Jakub Macina is a fourth year PhD at ETH advised by Mrinmaya Sachan and Manu Kapur. His research focuses on understanding and improving generative models' reasoning and pedagogical capabilities. His work has been published in venues such as ACL, EMNLP, and RecSys.

Donya Rooein is a Postdoc at Bocconi University; her work revolves around leveraging NLP for Education. She explores the synergy between machine learning, linguistics, and practitioner insights to enhance education systems. Her work has been published in different ML, NLP, and AIED venues, including NAACL, WWW, and EdMedia.

Mrinmaya Sachan is an Assistant Professor at ETH Zurich, focusing on NLP and its interface with Education. His group has published relevant research on the challenges of Pedagogy and LLMs, Educational Chatbots and Tutors, Student Modeling and Assessment across various NLP and Education-focused venues.

Shashank Sonkar is a final-year PhD student at Rice University advised by Richard G. Baraniuk. His work focuses on pedagogical alignment of LLMs, learner modeling, and intelligent assessment. His work has been published in EMNLP, COLING, AIED, EDM, and LAK.

7 Type of Tutorial & Target Audience

The tutorial will be **introductory** and present research from the fields of NLP, AIED and learning sciences. We will discuss seminal as well as recent papers to build a common ground for participants. Therefore, we welcome participants from any of these backgrounds. While it is helpful to have knowledge of either NLP / ML or learning sciences, it is not a requirement. The tutorial will be self-contained and welcomes an estimated 50-100 attendees based on recent BEA iterations. We will recommend the attendees a small reading list comprising of papers listed in the appendix.

References

- Vincent Aleven, Bruce M McLaren, Jonathan Sewall, and Kenneth R Koedinger. 2009. Example-tracing tutors: A new paradigm for intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 19(2):105–154.
- John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1997. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207.
- Sami Baral, Anthony F Botelho, Abhishek Santhanam, Ashish Gurung, John Erickson, and Neil T Heffernan. 2023. Investigating patterns of tone and sentiment in teacher written feedback messages. In *International Conference on Artificial Intelligence in Education*, pages 341–346. Springer.
- Richard G. Baraniuk. 2024. [Mid-scale ri-2: Safeinsights: A national research infrastructure for large-scale learning science and engineering](#). NSF Award Number 2153481.
- Arne Bewersdorff, Christian Hartmann, Marie Hornberger, Kathrin Seßler, Maria Bannert, Enkelejda Kasneci, Gjergji Kasneci, Xiaoming Zhai, and Claudia Nerdel. 2024. Taking the next step with generative artificial intelligence: The transformative role of multimodal large language models in science education. *arXiv preprint arXiv:2401.00832*.
- Giorgio Biancini, Alessio Ferrato, and Carla Limongelli. 2024. Multiple-choice question generation using large language models: Methodology and educator insights. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 584–590.
- 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.
- Sahan Bulathwela, Hamze Muse, and Emine Yilmaz. 2023. Scalable educational question generation with pre-trained language models. In *International Conference on Artificial Intelligence in Education*, pages 327–339. Springer.
- James E Carlson and Matthias von Davier. 2017. Item response theory. *Advancing human assessment: The methodological, psychological and policy contributions of ETS*, pages 133–178.
- Eason Chen, Jia-En Lee, Jionghao Lin, and Kenneth Koedinger. 2024. Gptutor: Great personalized tutor with large language models for personalized learning content generation. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 539–541.
- Li Cheng, Ethan Croteau, Sami Baral, Cristina Heffernan, and Neil Heffernan. 2024. Facilitating student learning with a chatbot in an online math learning platform. *Journal of Educational Computing Research*, 62(4):907–937.
- Alexis Chevalier, Jiayi Geng, Alexander Wettig, Howard Chen, Sebastian Mizera, Toni Annala, Max Jameson Aragon, Arturo Rodríguez Fanlo, Simon Frieder, Simon Machado, et al. 2024. Language models as science tutors. *arXiv preprint arXiv:2402.11111*.
- 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.
- 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*.
- Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. In *User modeling and user-adapted interaction*, volume 4, pages 253–278. Springer.
- 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.
- Peng Cui and Mrinmaya Sachan. 2023. Adaptive and personalized exercise generation for online language learning. *arXiv preprint arXiv:2306.02457*.
- 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](#). *Preprint*, arXiv:2407.09136.
- 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*, page 100299.
- Jean-Paul Doignon and Jean-Claude Falmagne. 1985. Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, 23(2):175–196.
- 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 *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, volume 2, pages 263–274.

- Hermann Ebbinghaus. 1913. *Memory: A Contribution to Experimental Psychology*. Teachers College, Columbia University, New York.
- 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.
- Alberto Gandolfi. 2024. Gpt-4 in education: Evaluating aptness, reliability, and loss of coherence in solving calculus problems and grading submissions. *International Journal of Artificial Intelligence in Education*, pages 1–31.
- 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.
- Arthur C Graesser, Patrick Chipman, Brian C Haynes, and Andrew Olney. 2005. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618.
- Bor Gregorcic, Giulia Polverini, and Andreja Sarlah. 2024. Chatgpt as a tool for honing teachers’ socratic dialogue skills. *Physics Education*, 59(4):045005.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- 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*.
- Wayne Holmes, Kaska Porayska-Pomsta, Ken Holstein, Emma Sutherland, Toby Baker, Simon Buckingham Shum, Olga C Santos, Mercedes T Rodrigo, Mutlu Cukurova, Ig Ibert Bittencourt, et al. 2022. Ethics of ai in education: Towards a community-wide framework. *International Journal of Artificial Intelligence in Education*, pages 1–23.
- 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.
- Kevin Hwang, Sai Challagundla, Maryam Alomair, Lujie Karen Chen, and Fow-Sen Choa. 2023. Towards ai-assisted multiple choice question generation and quality evaluation at scale: Aligning with bloom’s taxonomy. In *Workshop on Generative AI for Education*.
- Yoonjin Im, Eunseong Choi, Heejin Kook, and Jongwuk Lee. 2023. Forgetting-aware linear bias for attentive knowledge tracing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3958–3962.
- Joseph Marvin Imperial and Harish Tayyar Madabushi. 2023. Flesch or fumble? evaluating readability standard alignment of instruction-tuned language models. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 205–223, Singapore. Association for Computational Linguistics.
- Irina Jurenka and et al. 2024. Towards responsible development of generative ai for education: An evaluation-driven approach. *Preprint*, arXiv:2407.12687.
- Mustafa Kaya and Ilyas Cicekli. 2024. A hybrid approach for automated short answer grading. *IEEE Access*.
- Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. 2023. Bless: Benchmarking large language models on sentence simplification. *arXiv preprint arXiv:2310.15773*.
- Gerd Kortemeyer. 2023a. Performance of the pre-trained large language model gpt-4 on automated short answer grading. *arXiv preprint arXiv:2309.09338*.
- Gerd Kortemeyer. 2023b. Toward ai grading of student problem solutions in introductory physics: A feasibility study. *Physical Review Physics Education Research*, 19(2):020163.
- Unggi Lee, Jiyeong Bae, Dohee Kim, Sookbun Lee, Jaekwon Park, Taekyung Ahn, Gunho Lee, Damji Stratton, and Hyeoncheol Kim. 2024. Language model can do knowledge tracing: Simple but effective method to integrate language model and knowledge tracing task. *arXiv preprint arXiv:2406.02893*.
- Philipp Leitner, Markus Ebner, and Martin Ebner. 2019. Learning analytics challenges to overcome in higher education institutions. *Utilizing learning analytics to support study success*, pages 91–104.
- Hai Li, Wanli Xing, Chenglu Li, Wangda Zhu, and Neil Heffernan. 2024a. Positive affective feedback mechanisms in an online mathematics learning platform. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 371–375.
- Kunze Li and Yu Zhang. 2024. Planning first, question second: An llm-guided method for controllable question generation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4715–4729.
- Xueyi Li, Youheng Bai, Teng Guo, Zitao Liu, Yaying Huang, Xiangyu Zhao, Feng Xia, Weiqi Luo, and Jian Weng. 2024b. Enhancing length generalization for attention based knowledge tracing models with

- linear biases. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 5918–5926. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Naiming Liu, Zichao Wang, Richard G Baraniuk, and Andrew Lan. 2022. Gpt-based open-ended knowledge tracing. *arXiv preprint arXiv:2203.03716*.
- Tianyi Liu, Julia Chatain, Laura Kobel-Keller, Gerd Kortemeyer, Thomas Willwacher, and Mrinmaya Sachan. 2024a. Ai-assisted automated short answer grading of handwritten university level mathematics exams. *arXiv preprint arXiv:2408.11728*.
- Zitao Liu, Qiongqiong Liu, Teng Guo, Jiahao Chen, Shuyan Huang, Xiangyu Zhao, Jiliang Tang, Weiqi Luo, and Jian Weng. 2024b. Xes3g5m: A knowledge tracing benchmark dataset with auxiliary information. *Advances in Neural Information Processing Systems*, 36.
- Frederic M Lord. 1980. *Applications of item response theory to practical testing problems*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Xinyi Lu, Simin Fan, Jessica Houghton, Lu Wang, and Xu Wang. 2023. Readingquizmaker: a human-nlp collaborative system that supports instructors to design high-quality reading quiz questions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18.
- 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*, pages 16–27.
- 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, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023b. **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.
- Brendan P Mackey, Razmig Garabet, Laura Maule, Abay Tadesse, James Cross, and Michael Weingarten. 2023. Evaluating chatgpt-4 in medical education: an assessment of subject exam performance reveals limitations in clinical curriculum support for students.
- 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.
- Richard E Mayer. 2024. The past, present, and future of the cognitive theory of multimedia learning. *Educational Psychology Review*, 36(1):8.
- 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*.
- Fengchun Miao and Mutlu Cukurova. 2024. *AI competency framework for teachers*. UNESCO, Paris. Foreword by Stefania Giannini, UNESCO Assistant Director-General for Education. Includes bibliography.
- 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.
- Steven Moore, Eamon Costello, Huy A Nguyen, and John Stamper. 2024. An automatic question usability evaluation toolkit. In *International Conference on Artificial Intelligence in Education*, pages 31–46. Springer.
- Wesley Morris, Langdon Holmes, Joon Suh Choi, and Scott Crossley. 2024. Automated scoring of constructed response items in math assessment using large language models. *International Journal of Artificial Intelligence in Education*, pages 1–28.
- Tanya Nazaretsky, Paola Mejia-Domenzain, Vinitra Swamy, Jibril Frej, and K Käser. 2024. Ai or human? evaluating student feedback perceptions in higher education. In *European Conference on Technology Enhanced Learning, ECTEL 2024*.
- Allen Nie, Yash Chandak, Miroslav Suzara, Ali Malik, Juliette Woodrow, Matt Peng, Mehran Sahami, Emma Brunskill, and Chris Piech. 2024. The gpt surprise: Offering large language model chat in a massive coding class reduced engagement but increased adopters’ exam performances. Technical report, Center for Open Science.
- Filippa Nilsson and Jonatan Tuvstedt. 2023. Gpt-4 as an automatic grader: The accuracy of grades set by gpt-4 on introductory programming assignments.
- 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.
- Amy O’Hara and Stephanie Straus. 2022. Privacy preserving technologies in us education. *International Journal of Population Data Science*, 7(3).
- Samuel Ariyo Okaiyeto, Junwen Bai, and Hongwei Xiao. 2023. Generative ai in education: To embrace it or not? *International Journal of Agricultural and Biological Engineering*, 16(3):285–286.

- 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.
- Shalini Pandey and George Karypis. 2019. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*.
- Seyed Parsa Neshaei, Richard Lee Davis, Adam Hazimeh, Bojan Lazarevski, Pierre Dillenbourg, and Tanja Käser. 2024. Towards modeling learner performance with large language models. *arXiv e-prints*, pages arXiv–2403.
- 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.
- Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.
- Georg Rasch. 1960. *Probabilistic models for some intelligence and attainment tests*. Danish Institute for Educational Research, Copenhagen.
- Donya Rooein, Amanda Cercas Curry, and Dirk Hovy. 2023. Know your audience: Do llms adapt to different age and education levels? *arXiv preprint arXiv:2312.02065*.
- 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.
- Stefan Ruseti, Ionut Paraschiv, Mihai Dascalu, and Danielle S McNamara. 2024. Automated pipeline for multi-lingual automated essay scoring with reader-bench. *International Journal of Artificial Intelligence in Education*, pages 1–22.
- 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 *International Conference on Artificial Intelligence in Education*, pages 165–179. 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.
- 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.
- Burr Settles and Brendan Meeder. 2016. A trainable spaced repetition model for language learning. In *Proceedings of the 54th annual meeting of the association for computational linguistics*, volume 1, pages 1848–1858.
- Shashi Kant Shankar, Gayathri Pothancheri, Deepu Sasi, and Shitanshu Mishra. 2024. Bringing teachers in the loop: Exploring perspectives on integrating generative ai in technology-enhanced learning. *International Journal of Artificial Intelligence in Education*, pages 1–26.
- Kumar Shridhar, Jakub Macina, Mennatallah El-Assady, Tanmay Sinha, Manu Kapur, and Mrinmaya Sachan. 2022. [Automatic generation of socratic subquestions for teaching math word problems](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4136–4149, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- 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.
- Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, and Richard G Baraniuk. 2024a. Pedagogical alignment of large language models. *arXiv preprint arXiv:2402.05000*.
- Shashank Sonkar, Kangqi Ni, Lesa Tran Lu, Kristi Kincaid, John S Hutchinson, and Richard G Baraniuk. 2024b. Automated long answer grading with ricechem dataset. In *International Conference on Artificial Intelligence in Education*, pages 163–176. Springer.
- Megha Srivastava and Noah Goodman. 2021. Question generation for adaptive education. *arXiv preprint arXiv:2106.04262*.
- Jiahong Su, Davy Tsz Kit Ng, and Samuel Kai Wah Chu. 2023. Artificial intelligence (ai) literacy in early childhood education: The challenges and opportunities. *Computers and Education: Artificial Intelligence*, 4:100124.
- Jiahong Su and Weipeng Yang. 2023. Artificial intelligence (ai) literacy in early childhood education: An

- intervention study in hong kong. *Interactive Learning Environments*, pages 1–15.
- Danielle Thomas, Xinyu Yang, Shivang Gupta, Ade-tunji Adeniran, Elizabeth Mclaughlin, and Kenneth Koedinger. 2023. [When the Tutor Becomes the Student: Design and Evaluation of Efficient Scenario-Based Lessons for Tutors](#). In *LAK23: 13th International Learning Analytics and Knowledge Conference*, LAK2023, page 250–261, New York, NY, USA. Association for Computing Machinery.
- Danielle R Thomas, Jionghao Lin, Shambhavi Bhushan, Ralph Abboud, Erin Gatz, Shivang Gupta, and Kenneth R Koedinger. 2024. Learning and ai evaluation of tutors responding to students engaging in negative self-talk. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 481–485.
- Angel Olider Rojas Vistorte, Angel Deroncela-Acosta, Juan Luis Martín Ayala, Angel Barrasa, Caridad López-Granero, and Mariacarla Martí-González. 2024. Integrating artificial intelligence to assess emotions in learning environments: a systematic literature review. *Frontiers in Psychology*, 15:1387089.
- Lev Semyonovich Vygotsky. 1978. *Mind in society: The development of higher psychological processes*. Harvard University Press, Cambridge, MA.
- 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 and virtual meeting. Association for Computational Linguistics.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024b. [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 E Wang and Dorottya Demszky. 2023. Is chatgpt a good teacher coach? measuring zero-shot performance for scoring and providing actionable insights on classroom instruction. *arXiv preprint arXiv:2306.03090*.
- Xu Wang, Simin Fan, Jessica Houghton, and Lu Wang. 2022. Towards process-oriented, modular, and versatile question generation that meets educational needs. *arXiv preprint arXiv:2205.00355*.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. [Finetuned language models are zero-shot learners](#). In *International Conference on Learning Representations*.
- Yu Yin, Le Dai, Zhenya Huang, Shuanghong Shen, Fei Wang, Qi Liu, Enhong Chen, and Xin Li. 2023. [Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer](#). In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 855–864, New York, NY, USA. Association for Computing Machinery.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Reading List

We plan to share the following list of representative papers spanning the topics covered in our tutorial to the attendees to generate interest. However, we do not plan to have this as a requirement for attendees:

1. 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*, page 100299
2. 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
3. Alberto Gandolfi. 2024. Gpt-4 in education: Evaluating aptness, reliability, and loss of coherence in solving calculus problems and grading submissions. *International Journal of Artificial Intelligence in Education*, pages 1–31
4. Wayne Holmes, Kaska Porayska-Pomsta, Ken Holstein, Emma Sutherland, Toby Baker, Simon Buckingham Shum, Olga C Santos, Mercedes T Rodrigo, Mutlu Cukurova, Ig Ibert Bittencourt, et al. 2022. Ethics of ai in education: Towards a community-wide framework. *International Journal of Artificial Intelligence in Education*, pages 1–23
5. Irina Jurenka and et al. 2024. [Towards responsible development of generative ai for education: An evaluation-driven approach](#). *Preprint*, arXiv:2407.12687

6. Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, and Richard G Baraniuk. 2024a. Pedagogical alignment of large language models. *arXiv preprint arXiv:2402.05000*
7. Allen Nie, Yash Chandak, Miroslav Suzara, Ali Malik, Juliette Woodrow, Matt Peng, Mehran Sahami, Emma Brunskill, and Chris Piech. 2024. The gpt surprise: Offering large language model chat in a massive coding class reduced engagement but increased adopters' exam performances. Technical report, Center for Open Science
8. Xinyi Lu, Simin Fan, Jessica Houghton, Lu Wang, and Xu Wang. 2023. Readingquiz-maker: a human-nlp collaborative system that supports instructors to design high-quality reading quiz questions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18
9. Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513
10. Fengchun Miao and Mutlu Cukurova. 2024. *AI competency framework for teachers*. UNESCO, Paris. Foreword by Stefania Giannini, UNESCO Assistant Director-General for Education. Includes bibliography

Comparing human and LLM proofreading in L2 writing: Impact on lexical and syntactic features

Hakyung Sung¹ Karla Csuros^{2,1*} Min-Chang Sung^{3,1*}

¹LCR-ADS Lab, Linguistics, University of Oregon

²West University of Timisoara

³Gyeongin National University of Education

Abstract

This study examines the lexical and syntactic interventions of human and LLM proofreading aimed at improving overall intelligibility in identical second language writings, and evaluates the consistency of outcomes across three LLMs (ChatGPT-4o, Llama3.1-8b, Deepseek-r1-8b). Findings show that both human and LLM proofreading enhance bigram lexical features, which may contribute to better coherence and contextual connectedness between adjacent words. However, LLM proofreading exhibits a more generative approach, extensively reworking vocabulary and sentence structures, such as employing more diverse and sophisticated vocabulary and incorporating a greater number of adjective modifiers in noun phrases. The proofreading outcomes are highly consistent in major lexical and syntactic features across the three models.

1 Introduction

The use of generative large language models (LLMs) in second language (L2) writing has gained popularity for providing real-time feedback on vocabulary, grammar, and style (e.g., Han et al., 2024; Meyer et al., 2024). These models offer immediate corrective suggestions, enhancing the precision and quality of L2 writing—a role once largely filled by human editors with expertise. As LLMs increasingly replace or supplement human intervention, questions arise about their impact on L2 writings.

While previous studies have concentrated on general error correction through LLM proofreading (e.g., Heintz et al., 2022; Su et al., 2023; Wu et al., 2023; Katinskaia and Yangarber, 2024), recent studies have shown that LLMs do not consistently outperform state-of-the-art supervised grammatical error correction models on minimal-edit benchmarks, often producing more fluency-oriented rewrites instead (Davis et al., 2024). This tendency stems in

part from the fact that LLMs, by default, generate transformative fluency corrections rather than minimal edits when processing ungrammatical text (e.g., Coyne et al., 2023; Fang et al., 2023; Loem et al., 2023). However, little research has examined how this generative rewriting behavior affects broader lexical and syntactic characteristics of L2 writing compared to human proofreading, especially when the proofreading goal extends beyond grammatical accuracy to overall intelligibility. Moreover, it remains unclear whether different LLMs yield consistent proofreading outcomes. This study addresses these gaps by posing three guiding questions: (1) What are the similarities and differences in lexical features between human proofreading and LLM proofreading of L2 writings? (2) What are the similarities and differences in syntactic features between human proofreading and LLM proofreading of L2 writings? (3) Do three different LLMs provide consistent proofreading outcomes in terms of lexical and syntactic features in L2 writing?

Our findings show that while both human and LLM proofreading enhance lexical and syntactic features, LLMs are more likely to make more extensive lexical and syntactic edits. By quantifying these changes through a range of lexical and syntactic indices, we reveal that LLMs favor more generative rewrites, which may improve fluency but risk altering nuance or inflating perceived proficiency.

2 Background

2.1 Proofreading in L2 writing

Proofreading is a complex issue in writing research, particularly for L2 writers, as it involves varying scopes of interventions. Traditional definitions of proofreading often restrict it to surface-level error correction that focuses on resolving orthographic and grammatical errors without altering content (Carduner, 2007; Hyatt et al., 2017). However, research shows that professional human proofreaders

*Contributed equally to the study.

occasionally restructure content to improve the logical flow of ideas and make the writing easier to understand (Salter-Dvorak, 2019). Noting these varying practices in proofreading, Harwood et al. (2009, p. 167) provided a quite general definition of proofreading as “[any] third-party interventions (entailing written alteration) on assessed work in progress.”

Previous studies have shown that human proofreading displays variability not just in scope, but also in quality. Harwood (2018) found that 14 proofreaders made between 113 and 472 changes to the same L2 learner essay, with some interventions improving clarity and others introducing new errors, leading to inconsistent quality. Similarly, Shafto (2015) argued that proofreading is a highly attention-dependent task, meaning that symptoms such as tiredness can heavily impact human proofreaders’ ability to detect and correct ungrammatical and unnatural expressions.

The debate surrounding the adequacy of L2 proofreading is also characterized by varying perspectives from stakeholders (i.e., students, faculty, researchers). While L2 students often seek proofreading services to improve their grades or enhance their writing skills, some faculty view such assistance as a form of academic dishonesty (Salter-Dvorak, 2019; Turner, 2011). Despite these divergent opinions, there is a general consensus that proofreaders can significantly enhance language accuracy and clarity in L2 writing, provided that the original authorial voice is maintained (Turner, 2024; Warschauer et al., 2023; Zou and Huang, 2024).

2.2 LLMs in L2 writing and proofreading

While automated written corrective feedback has been present in L2 classrooms for over a decade (cf. Wilson et al., 2014), recent research is now exploring how LLM assistants can be incorporated into holistic writing workflows (Zhao, 2024). Researchers examine the integration of the LLM in prewriting (Xiao, 2024) and postwriting stages (Osawa, 2024), as well as its role in fostering metacognitive skills through iterative revisions that include editing and proofreading (Su et al., 2023; Warschauer et al., 2023; Zou and Huang, 2024).

Among these LLM integrations, several studies have highlighted the capabilities of LLM proofreading (or more broadly, editing). For instance, Su et al. (2023) found that ChatGPT effectively assessed grammar, clarified meaning, and sug-

gested lexical and syntactic refinements. Similarly, Yan and Zhang (2024) observed that ChatGPT identified and corrected a range of linguistic errors—including lexical (e.g., word choice, idioms), grammatical (e.g., verb tense, articles), structural (e.g., run-on or fragmented sentences), mechanical (e.g., spelling, punctuation), and stylistic (e.g., formality) aspects.

Few studies have compared LLM proofreading directly to human revisions. For instance, Heintz et al. (2022) compared outputs edited by LLMs with those revised by human editors using sentences written by non-native English speakers. They found that while Wordvice AI¹ achieved near-human accuracy (77%) in correcting grammar and spelling errors, it lagged behind human editors in areas like vocabulary refinement and fluency adjustments. Similarly, Jiang et al. (2023) analyzed 2,197 T-units² and 1,410 sentences from weekly writing samples of 41 Chinese students in an online high school language program at a U.S. university. They found that ChatGPT-4 achieved high precision (88%) in correcting errors at the T-unit level (in comparison to human judgments), but sometimes overcorrected valid sentences or misinterpreted context-dependent issues, such as ambiguous word order and culturally embedded idioms.

2.3 Summary of findings and research gaps

To briefly summarize, previous research has demonstrated that proofreading in L2 writing is highly variable in both scope and quality, with interventions ranging from surface-level corrections to content restructuring. Recently, LLMs have been shown to offer performance comparable to, or even surpassing, that of human editors in L2 writing proofreading, although they exhibit limitations in context-sensitive judgment and cultural awareness.

Despite these insights, still little is known about the fine-grained linguistic interventions that could be made by LLMs compared to human proofreaders. Additionally, existing research has focused primarily on grammatical error detection and correction, overlooking broader language use. For example, although LLMs may facilitate vocabulary expansion, it remains unclear how their suggestions differ from those of human proofreaders, and detailed syntactic changes remain underexplored.

¹<https://wordvice.ai/proofreading>

²A T-unit is often defined as the minimal grammatical unit, comprising a single independent clause plus any subordinate clauses or dependent phrases attached to it (Lu, 2010).

Moreover, most studies have examined only one type of LLM, leaving open the question of whether these linguistic changes are specific to one model or generalizable across other LLMs.

3 Methods

3.1 Dataset

This study utilizes the ICNALE Edited Essays dataset, one of the publicly available corpora within the International Corpus Network of Asian Learners of English (ICNALE) project (Ishikawa, 2018, 2021). The dataset comprises 656 essays written by 328 L2 learners and their edited versions produced by professional native English-speaking proofreaders.

The L2 participants were college students learning English in ten regional contexts: Japan (JPN), Korea (KOR), China (CHN), Taiwan (TWN), Indonesia (IDN), Thailand (THA), Hong Kong (HKG), the Philippines (PHL), Pakistan (PAK), and Singapore (SIN). Each participant wrote two argumentative essays in response to the prompts: (1) “It is important for college students to have a part-time job” and (2) “Smoking should be completely banned at all restaurants”.

3.1.1 Rationale for dataset selection and representativeness

The ICNALE dataset was chosen for three main reasons. First, it provides paired original and professionally proofread versions, allowing for direct comparison with LLM-generated outputs. Second, it includes explicit L2 proficiency labels, facilitating stratified analyses across proficiency levels. Last, it offers balanced regional coverage across ten Asian countries or regions (see Table 1). However, we acknowledge that broad generalizations to other genres or demographic groups (e.g., narrative writing, younger learners) must be made with caution.

3.1.2 Proficiency band

All participants were classified into four L2 proficiency bands (linked to the Common European Framework of Reference for Languages) based on their recent scores in standardized English tests (e.g., TOEFL, TOEIC) or their performance in a standard receptive vocabulary test³ (Nation and

³The vocabulary test consists of 50 multiple-choice items designed to measure vocabulary knowledge within the 1,000–5,000 word range. A typical item (from the 4,000-word level) presents a short sentence containing a target word and asks

Beglar, 2007). Table 1 shows the proficiency distribution of each regional learner group.

Region	A2_0	B1_1	B1_2	B2_0	Total
JPN	10	10	10	10	40
KOR	10	10	10	10	40
CHN	10	10	10	10	40
TWN	10	10	10	10	40
IDN	10	10	10	3	33
THA	10	10	10	2	32
HKG	–	10	10	10	30
PHL	–	10	10	10	30
PAK	–	10	10	3	23
SIN	–	–	10	10	20
Total	60	90	100	78	328

Table 1: Distribution of participants by region and proficiency

3.1.3 Proofreading process and proofreader profiles

The ICNALE project recruited five experienced proofreaders with strong academic backgrounds and extensive experience in editing scholarly work. Their profiles are summarized in Table 2.

ID	Age	Sex	Degree	Experience (years)	L1 English
A	28	Female	BA	3	Canadian
B	32	Female	MS	5	Australian
C	27	Female	BS	3	American
D	38	Female	BS	10	British
E	31	Female	PhD	2	Australian

Table 2: Profiles of proofreaders in the ICNALE project

As documented in the ICNALE project, the professional proofreaders were tasked with editing errors and inappropriate wording to ensure that each essay became fully intelligible (Ishikawa, 2021, p. 496). No standardized rubric or adjudication mechanism was imposed at the original corpus compilation stage. All revisions were performed in MS Word using the Track Changes function, which allowed every edit, addition, or deletion to be recorded.

A calibration study in which all five proofreaders revised the same eight essays revealed substantial variability in editing behavior (cf. Ishikawa, 2018, p. 122). The number of edited word tokens ranged from 40.00 to 59.63—a difference of 19.63 tokens, or 40.97% of the average. Ishikawa (2021) attributed this variation to the inherent subjectivity of human editing, shaped by individual judgments of intelligibility.

test-takers to select the most appropriate definition.

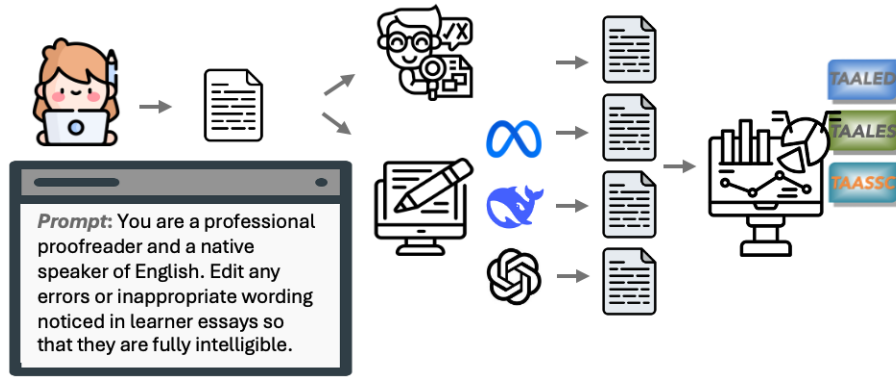


Figure 1: Overview of the experiment

3.2 LLM selection and prompt design

Figure 1 outlines the experiment. First, to compare the human proofreading in the ICNALE project with LLM proofreading, we selected three text-generating LLMs: GPT-4o (used in ChatGPT, accessed via OpenAI’s API; Achiam et al., 2023, hence we called them *Chatgpt-4o*), *Llama3.1-8b* (Touvron et al., 2023), and *Deepseek-r1-8b* (Guo et al., 2025). ChatGPT-4o was chosen due to its widespread accessibility, although its underlying parameter count and architecture remain proprietary. In contrast, both *Llama3.1-8b* and *Deepseek-r1-8b* are open models with 8 billion parameters that are lightweight enough for local installations, with *Deepseek-r1-8b* being a distilled version of *Llama3.1-8b*.

Each model was tasked with reading the original L2 writings and generating a proofread version based solely on a standardized prompt, with no access to additional learner information. The exact prompt used was as follows: “You are a professional proofreader and a native speaker of English. Edit any errors or inappropriate wording noticed in learner essays so that they are fully intelligible. Return only the final edited version of the essay. Do not include any explanations, comments, reasoning, or additional thoughts in your response.” This prompt was designed to align with the instructions given to ICNALE proofreaders—“They were asked to edit any error or inappropriate wording noticed in learner essays so that they could be fully intelligible. They were also required not to ‘rewrite’ the original texts, that is, not to add new content or to alter organization” (Ishikawa, 2021, p. 496)—ensuring consistency with the human proofreading protocol for fair comparison.

3.3 Lexical and syntactic analyses

The proofread-and-generated texts, along with the learner and edited texts in the ICNALE dataset, were processed to extract lexical and syntactic features using the source codes of publicly available NLP tools: TAALED (cf. Kyle et al., 2024), TAALES (cf. Kyle et al., 2018) and TAASSC (cf. Kyle and Crossley, 2018). We measured lexical and syntactic aspects of the learner and proofread essays based on the concept of linguistic complexity, which provides a descriptive-analytic framework for L2 production (Bulté and Housen, 2012; Bulté et al., 2024).

3.3.1 Lexical features

Lexical features were evaluated in terms of two aspects: diversity and sophistication. Lexical diversity indices reflect vocabulary variation and repetition, with higher scores indicating a broader vocabulary range and fewer repetitions. In this study, we employ common measures such as the number of unique words and the moving-average type-token ratio—the latter mitigating the impact of text length on traditional lexical diversity measures (Kyle et al., 2024).

Lexical sophistication indices, on the other hand, focus on measuring the use of advanced words (Laufer and Nation, 1995; Meara and Bell, 2001). They are typically assessed based on relative word frequency, semantic concreteness, and domain or register distinctiveness, with less frequent, less concrete, and more domain-specific words generally considered more sophisticated (Kyle et al., 2018). We also incorporate the concept of ngram sophistication by analyzing associations and dependency relations within bigrams (Kyle and Eguchi, 2021).

3.3.2 Syntactic features

Syntactic features can be examined from multiple perspectives. Traditional approaches, such as measuring the average length of T-units, focus on the overall length of syntactic structures and operate under the assumption that longer units generally indicate greater complexity (Lu, 2010, 2011).

In contrast, fine-grained syntactic complexity indices (Kyle and Crossley, 2018) provide a more nuanced analysis by capturing specific structural characteristics rather than relying on surface-level measures like sentence length. These indices are often categorized into clausal-level (e.g., nominal subjects per clause), phrasal-level (e.g., dependents per nominal, including adjectives and prepositions), and morphosyntactic-level features (e.g., use of past tense).

To the best of our knowledge, there is no consensus on which fine-grained indices reliably capture syntactic complexity as perceived by human judges. Nevertheless, L2 writing studies suggest that higher-proficiency learners (identified by human ratings) tend to use more elaborated noun phrases (e.g., Biber et al., 2011).

3.4 Statistical methods

3.4.1 Evaluating linguistic features across groups

Prior to statistical analyses, we confirmed that the five groups of texts (i.e., original [ORIG], human-proofread [EDIT], and the three LLM-proofread versions) were largely comparable in length.⁴ This comparability, with the exception of Deepseek-r1-8b, indicates that subsequent improvements in lexical and syntactic domains are not simply due to different text lengths.

We calculated a range of 49 lexical and 143 syntactic indices from every text in the five groups and identified features showing significant between-group variance in two stages. First, we conducted visual inspection of box plots to exclude the indices with a great number of outliers, little individual variance, and/or unnoticeable mean differences. Second, we applied a linear mixed-effects model to each index, using Group (e.g., ORIG, EDIT, ChatGPT-4o) as a categorical fixed effect with ORIG as the baseline. Proficiency was included as a fixed effect that interacted with Group,

⁴The differences in the number of word tokens relative to the original text were: EDIT: -1.02, ChatGPT-4o: +6.13, Llama3.1-8b: -3.38, and Deepseek-r1-8b: -15.11***.

and Participants were included as a random effect. We retained only those models that converged successfully to ensure reliable estimates. From these convergent models, we focused primarily on the main effect of the proofreading mode, while also examining whether any observed mode effects were moderated by Proficiency. These procedures yielded six lexical and nine syntactic indices. Detailed descriptions of each index are provided in Appendix A.

For each of these indices, we reported the results of four pairwise comparisons, between ORIG and human or LLM proofreading, from the linear mixed-effects models. To avoid a Type I error due to multiple comparisons, we applied a Bonferroni adjustment to the alpha level, reducing it from .05 to .0125.

3.4.2 Evaluating consistency across LLMs

The linear mixed-effects analyses informed us that the cross-model evaluation should exclude five more syntactic features, which showed multicollinearity or overlapping metrics. For the rest ten features,⁵ we calculated the standardized z-scores so that each metric contributed equally to a composite measure of overall lexical and syntactic complexity.

Next, we restructured the data so that each row represented an essay and each column contained the composite score derived from the output of a different model, treating these composite scores as “ratings” of the same essay. We then calculated the Pearson correlation coefficients between the ratings for every pair of models’ proofread output and computed Cronbach’s alpha (Cronbach, 1951) across these scores to assess their overall consistency. All datasets and code used for this analysis are available in the supplementary repository: https://osf.io/mhtpg/?view_only=13ce0959a80e4d498b6761aba197bc83.

4 Results

4.1 Lexical features

Table 3 summarizes the analysis of the selected lexical sophistication and diversity features. First, all proofreading modes, including human editing, led to significantly higher bigram mutual information (raw_bg_MI) scores. This finding suggests that

⁵Lexical features: *matr*, *b_concreteness*, *mcd*, *usf*, *cw_lemma_freq_log*, and *raw_bg_MI*; Syntactic features: *nonfinite_prop*, *amod_dep*, *nominalization*, and *be_mv*.

Index	EDIT	ChatGPT-4o	Llama3.1-8b	Deepseek-r1-8b
raw_bg_MI	+0.35 / 1.80***	+0.65 / 3.30***	+0.62 / 3.17***	+0.60 / 3.03***
usf	-1.37 / 0.15	-9.21 / 0.99***	-8.48 / 0.91***	-12.09 / 1.30***
b_concreteness	+0.00 / 0.02	-0.15 / 0.83***	-0.12 / 0.67***	-0.21 / 1.11***
cw_lemma_freq_log	-0.02 / 0.03	-0.30 / 0.54***	-0.26 / 0.47***	-0.37 / 0.67***
mattr	+0.01 / 0.18	+0.07 / 2.20***	+0.08 / 2.63***	+0.10 / 3.41***
ntypes	+0.63 / 0.05	+19.98 / 1.68***	+16.68 / 1.40***	+16.80 / 1.41***

Table 3: Lexical features compared; For each index, two numbers are shown: the value on the *left* indicates the unstandardized main effect coefficient, while the value on the *right* (following the backslash) represents the standardized coefficient, calculated as the ratio of the coefficient to the residual standard deviation of the dependent variable; Significance vs. ORIG is marked ($*p < 0.0125$, $**p < 0.0025$, $***p < 0.00025$); negative values are red and positive values are blue; interaction effects are omitted.

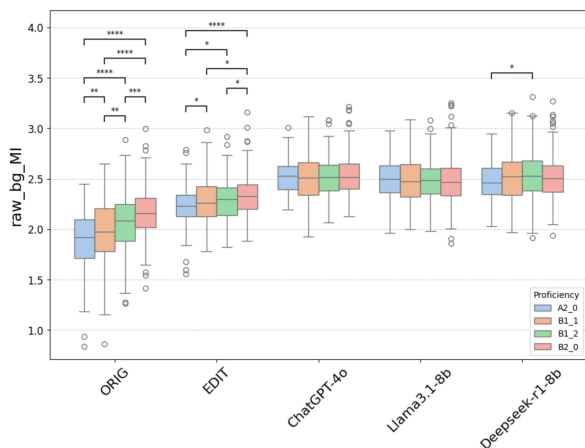


Figure 2: raw_bg_MI compared across ORIG, EDIT, and LLM-proofread texts by proficiency

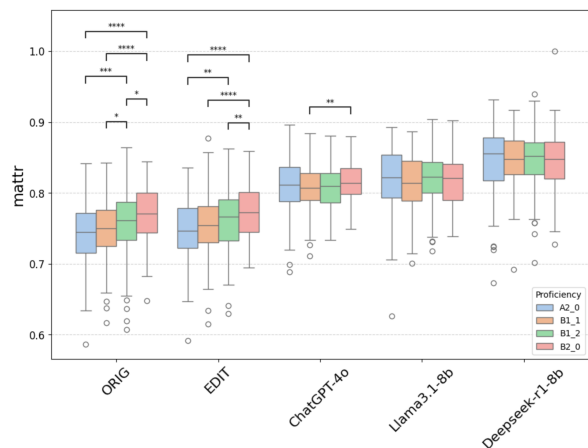


Figure 3: mattr compared across ORIG, EDIT, and LLM-proofread texts by proficiency

both human and LLM proofreading improved the lexical sophistication in terms of the coherence or contextual connectedness of adjacent words. However, LLM proofreading substantially increased raw_bg_MI to the extent that differences between lower and higher proficiency levels became less distinguishable (Figure 2).

In contrast, only the LLM-proofread texts showed significant changes in additional lexical sophistication measures, including a shift toward more contextually distinctive words (usf), less concrete words (b_concreteness), and lower-frequency content words (cw_lemma_freq_log). Human proofreading, by comparison, did not produce significant differences in these measures.

As for lexical diversity, significant improvements were observed only in the LLM-proofread texts, with increases in metrics such as mattr (Figure 3) and ntypes, indicating a broader range of vocabulary use.

4.2 Syntactic features

Table 4 summarizes the analysis of the selected syntactic features. Regarding the mean length of T-units (mltu), neither human nor LLM proofreading produced a consistent pattern: human proofreading (EDIT) and ChatGPT-4o tended to reduce T-unit length, while Llama3.1-8b and Deepseek-r1-8b tended to increase it, suggesting no uniform effect on the length of minimal grammatical units.

At the clause level, all LLM-proofread texts showed a significant increase in the total number of clauses (all_clauses) compared to the original learner essays, with Deepseek-r1-8b exhibiting the largest effect. Moreover, LLM-proofread texts contained a higher proportion of nonfinite clauses (nonfinite_prop), whereas human editing resulted in a slight reduction in this index.

At the phrase level, LLM proofreading increased the number of noun phrases (np), along with a rise in noun phrase dependencies (np_deps). This

Index	EDIT	ChatGPT-4o	Llama3.1-8b	Deepseek-r1-8b
mltu	-115.49 / 0.31	-105.73 / 0.28	+44.26 / 0.12	+118.42 / 0.31
all_clauses	+15.55 / 0.10	+133.76 / 0.84 ^{***}	+99.12 / 0.62 ^{***}	+179.00 / 1.12 ^{***}
nonfinite_prop	-1.33 / 0.29	+2.01 / 0.44 ^{***}	+2.63 / 0.57 ^{***}	+5.52 / 1.20 ^{***}
np	-21.30 / 0.08	+91.96 / 0.36 ^{**}	+41.27 / 0.16	+194.91 / 0.76 ^{***}
np_deps	-35.03 / 0.08	+79.21 / 0.17	+91.91 / 0.20	+217.81 / 0.47 ^{**}
amod_dep	+17.54 / 0.01	+137.65 / 0.75 ^{***}	+127.44 / 0.70 ^{***}	+204.54 / 1.12 ^{***}
nominalization	+58.12 / 0.40 ^{**}	+152.04 / 1.05 ^{***}	+102.85 / 0.71 ^{***}	+213.63 / 1.47 ^{***}
be_mv	+10.37 / 0.12	-56.53 / 0.63 ^{***}	-41.60 / 0.47 ^{**}	-84.02 / 0.94 ^{***}
past_tense	-15.80 / 0.29	-17.38 / 0.32	-17.77 / 0.32	-19.31 / 0.35 ^{**}

Table 4: Syntactic features compared; Interpretation of the table follows the same conventions described in Table 3

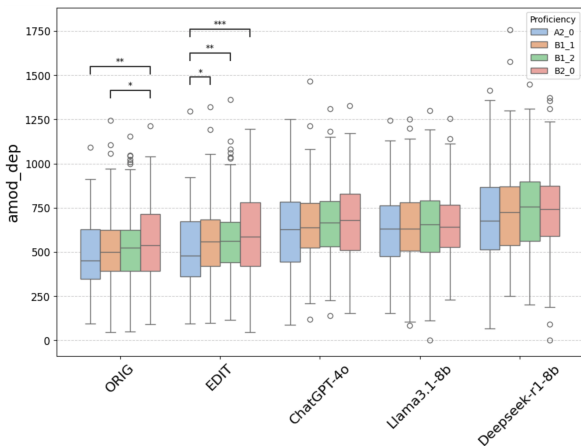


Figure 4: amod_dep compared across ORIG, EDIT, and LLM-proofread texts by proficiency

suggests that LLM proofreading not only added more noun phrases but also enriched their internal structure. In particular, the marked increase in adjective modifier dependencies (amod_dep; e.g., “various jobs”) suggests that LLM outputs favor more descriptive noun phrases (Figure 4).

At the morphological-syntactic level, both human and LLM proofreading showed significant increases in nominalization, but the increases were more pronounced in the LLM outputs (Figure 5). In contrast, the non-auxiliary use of the main verb “be” declined significantly under LLM proofreading, while human proofreading showed only a slight increase (be_mv). Additionally, all proofreading modes consistently reduced the use of past tense (past_tense).

4.3 Cross-model consistency

Based on the features that demonstrated meaningful group differences—and after removing indices with multicollinearity and conceptual overlap—we

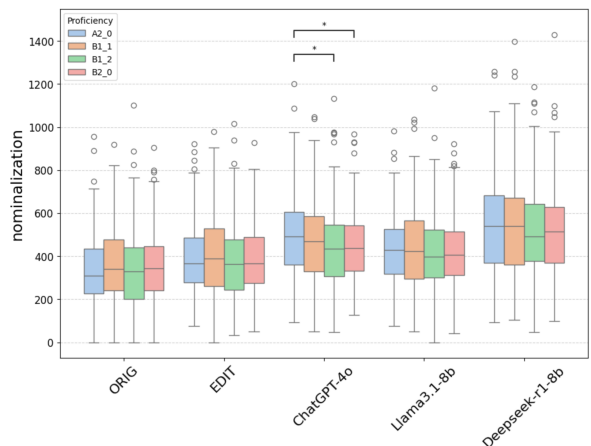


Figure 5: nominalization compared across ORIG, EDIT, and LLM-proofread texts by proficiency

Pair	Lexical	Syntax
ChatGPT-4o – Llama3.1-8b	0.70	0.62
ChatGPT-4o – Deepseek-r1-8b	0.60	0.53
Llama3.1-8b – Deepseek-r1-8b	0.56	0.65

Table 5: Pairwise Pearson correlations for lexical and syntactic features across LLMs

selected ten lexical or syntactic features. The composite lexical and syntactic scores exhibit strong internal consistency across the LLMs, with Cronbach’s alpha values of 0.83 and 0.81, respectively.

Table 5 presents the pairwise Pearson correlations among the three LLM proofreading models. For lexical features, ChatGPT-4o and Llama3.1-8b correlate at 0.70, while Deepseek-r1-8b correlates at 0.60 with ChatGPT-4o and 0.56 with Llama3.1-8b. For syntactic features, the corresponding correlations are 0.62, 0.53, and 0.65. These findings suggest that, despite minor variations, particularly with Deepseek-r1-8b, the LLMs tended to modify

vocabulary and syntactic structures in a relatively consistent manner when proofreading L2 writings, as measured by our selected indices.

5 Discussions

We compared the lexical and syntactic features of original L2 writings with those of texts that were proofread by human and LLMs. We also evaluated the consistency of LLM proofreading across different models.

Lexical features We found significant increases in bigram association strength, a ngram-level index of lexical sophistication, across all the proofreading modes. However, only LLM-proofread texts demonstrated notable changes in both word-level sophistication and diversity. Together, these results suggest that while both human and LLM proofreading improved the natural sequence of vocabulary—thus, enhancing the intelligibility of L2 writings—LLM proofreading provided an additional boost in lexical diversity and sophistication. In fact, this boost sometimes reduced or even eliminated typical differences between proficiency levels. Given that lexical sophistication and diversity are important constructs when evaluating L2 writing proficiency (Kyle et al., 2018, 2021), texts produced using LLM proofreading may obscure learners’ true writing abilities and artificially inflate their advanced language skills, ultimately undermining accurate assessment and long-term development.

We also observed that LLMs often replaced repeated words with alternative expressions—even when such changes are unwarranted—calling for caution. For example, “I often can smell” became “I often catch a whiff”, altering the intended meaning. Consequently, L2 writers using LLM proofreading should be mindful of unintended shifts in meaning or style and double-check suggested edits.

Syntactic features Compared with the marked lexical shifts, syntactic edits were subtler but still distinct pattern of edits. First, both human and LLM proofreading consistently reduced past-tense verbs, favoring present or neutral tense—a pattern often associated with factual, persuasive prose (Burrough-Boenisch, 2003; Fang and Maglio, 2024).

However, LLMs made more extensive structural modifications, including a higher proportion of non-finite clauses (e.g., “Because the company that need worker will ask the job experiences” → “Compa-

nies looking to hire often require prior work experience”) and a marked increase in adjective modifier dependencies (e.g., “become the social problem” → “become a significant social problem”). They also introduced more nominalizations (e.g., “we should...” → “(our) primary responsibility”) and reduced the non-auxiliary use of the main verb “be” (e.g., “is not the first” → “should not take precedence”).

Meanwhile, although the increase in overall noun complexity following LLM proofreading was not statistically robust (dp_deps), the gains were primarily driven by the insertion of adjective modifiers rather than by broader grammatical restructuring. For example, the structural complexity of noun phrases involving prepositional phrases (e.g., “disadvantages of works”) or coordination (e.g., “advantages and disadvantages”) remained largely unchanged.

Cross-model consistency We found that the three LLMs exhibit generally consistent proofreading performance in terms of the major lexical and syntactic features. We speculate that this consistency arises from fundamental similarities in how they are trained and optimized for language generation tasks. Consequently, while different LLMs may produce distinct outputs, their overall patterns of lexical enhancement and syntactic restructuring remain comparable.

6 Conclusions

Our study shows that while both human and LLM proofreading improve lexical and syntactic features in L2 writing, LLMs typically implement more generative edits, reworking vocabulary and sentence structures to a greater extent. Although these changes may enhance clarity and style, they risk overshadowing the original meaning or authorial voice and potentially inflate apparent language proficiency.

This finding has important implications for L2 writing practice. Acknowledging the great similarities in proofreading outcomes across different LLMs, more attention should be given to the question of “how to use LLM-proofreading effectively” rather than “what LLM to use for proofreading.” This key question can be addressed in reference to the observations that we have reported above, such as non-mandatory lexical substitution and excessive syntactic restructuring. Being aware of these tendencies in LLM-proofreading, L2 writers can

better maintain control over their writing process while strategically making use of LLMs for linguistic improvements.

Limitations

This study has several limitations. First, the same proofreading directive may be interpreted differently by human and LLM proofreaders, potentially affecting the nature and extent of the modifications.

Second, the analysis lacks qualitative comparisons between original and edited texts, which could reveal subtler aspects of the revisions. As one reviewer noted, LLM-proofread essays may appear more sophisticated but sometimes sacrifice coherence or introduce unintended nuances, making them harder to read. A more systematic qualitative analysis (ideally supported by human perception data comparing human- and LLM-proofread texts) would clarify whether LLM edits genuinely improve writing quality or simply enhance surface-level features.

Third, the task effects and proficiency-level constraints limit generalizability: our analysis focused solely on argumentative writing by Asian university-level students who already possess a certain level of L2 English proficiency. Consequently, these findings may not extend to other types of writing or to L2 groups with different backgrounds.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. [Gpt-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Douglas Biber, Bethany Gray, and Kornwipa Poonpon. 2011. [Should we use characteristics of conversation to measure grammatical complexity in l2 writing development?](#) *Tesol Quarterly*, 45(1):5–35.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. [Concreteness ratings for 40 thousand generally known english word lemmas](#). *Behavior Research Methods*, 46:904–911.
- Bram Bulté, Alex Housen, and Gabriele Pallotti. 2024. [Complexity and difficulty in second language acquisition: A theoretical and methodological overview](#). *Language Learning*.
- Bram Bulté and Alex Housen. 2012. [Defining and operationalising l2 complexity](#). In Alex Housen, Folkert Kuiken, and Ineke Vedder, editors, *Dimensions of L2 Performance and Proficiency: Complexity, Accuracy and Fluency in SLA*, pages 21–46. John Benjamins Publishing Company, Amsterdam.
- Joy Burrough-Boenisch. 2003. [Examining present tense conventions in scientific writing in the light of reader reactions to three dutch-authored discussions](#). *English for Specific Purposes*, 22(1):5–24.
- Jessie Carduner. 2007. [Teaching proofreading skills as a means of reducing composition errors](#). *Language Learning Journal*, 35(2):283–295.
- Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. [Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction](#). *arXiv preprint arXiv:2303.14342*.
- Lee J Cronbach. 1951. [Coefficient alpha and the internal structure of tests](#). *Psychometrika*, 16(3):297–334.
- Christopher Davis, Andrew Caines, O 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.
- David Fang and Sam J Maglio. 2024. [Time perspective and helpfulness: Are communicators more persuasive in the past, present, or future tense?](#) *Journal of Experimental Social Psychology*, 110:104544.
- Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. [Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation](#). *arXiv preprint arXiv:2304.01746*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*.
- Jieun Han, Haneul Yoo, Junho Myung, Minsun Kim, Hyunseung Lim, Yoonsu Kim, Tak Yeon Lee, Hwajung Hong, Juho Kim, So-Yeon Ahn, and Alice Oh. 2024. [LLM-as-a-tutor in EFL writing education: Focusing on evaluation of student-LLM interaction](#). In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 284–293, Miami, Florida, USA. Association for Computational Linguistics.
- Nigel Harwood. 2018. [What do proofreaders of student writing do to a master’s essay? differing interventions, worrying findings](#). *Written Communication*, 35(4):474–530.
- Nigel Harwood, Liz Austin, and Rowena Macaulay. 2009. [Proofreading in a uk university: Proofreaders’ beliefs, practices, and experiences](#). *Journal of Second Language Writing*, 18(3):166–190.
- Kevin Heintz, Younghoon Roh, and Jonghwan Lee. 2022. [Comparing the accuracy and effectiveness of wordvice ai proofreader to two automated editing tools and human editors](#). *Science Editing*, 9(1):37–45.
- Jon-Philippe K Hyatt, Elisa Jayne Bienenstock, and Jason U Tilan. 2017. [A student guide to proofreading and writing in science](#). *Advances in Physiology Education*, 41(3):324–331.
- Shin’ichiro Ishikawa. 2018. [The icnale edited essays: A dataset for analysis of l2 english learner essays based on a new integrative viewpoint](#). *English Corpus Studies*, 25:117–130.
- Shin’ichiro Ishikawa. 2021. [Asian learners’ knowledge and use of l2 english words and phrases: A corpus-based study on learners in china, japan, korea, and taiwan](#). In J. Szerszunowicz, editor, *Intercontinental Dialogue on Phraseology 4: Reproducible Language Units from an Interdisciplinary Perspective*, pages 493–510. University of Bialystok Publishing House.
- Zilu Jiang, Zexin Xu, Zilong Pan, Jingwen He, and Kui Xie. 2023. [Exploring the role of artificial intelligence in facilitating assessment of writing performance in second language learning](#). *Languages*, 8(4):247.
- Anisia Katinskaia and Roman Yangarber. 2024. [Gpt-3.5 for grammatical error correction](#).
- Kristopher Kyle, Scott Crossley, and Cynthia Berger. 2018. [The tool for the automatic analysis of lexical sophistication \(taales\): Version 2.0](#). *Behavior Research Methods*, 50:1030–1046.

- Kristopher Kyle and Scott A Crossley. 2018. [Measuring syntactic complexity in L2 writing using fine-grained clausal and phrasal indices](#). *The Modern Language Journal*, 102(2):333–349.
- Kristopher Kyle, Scott A Crossley, and Scott Jarvis. 2021. [Assessing the validity of lexical diversity indices using direct judgements](#). *Language Assessment Quarterly*, 18(2):154–170.
- Kristopher Kyle and Masaki Eguchi. 2021. [Automatically assessing lexical sophistication using word, bigram, and dependency indices](#). In Frances Blanchette and Constantine Lukyanenko, editors, *Perspectives on the L2 Phrasicon: The View from Learner Corpora*, pages 126–151. De Gruyter Brill, Berlin.
- Kristopher Kyle, Hakyung Sung, Masaki Eguchi, and Fred Zenker. 2024. [Evaluating evidence for the reliability and validity of lexical diversity indices in L2 oral task responses](#). *Studies in Second Language Acquisition*, 46(1):278–299.
- Batia Laufer and Paul Nation. 1995. [Vocabulary size and use: Lexical richness in L2 written production](#). *Applied Linguistics*, 16(3):307–322.
- 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.
- Xiaofei Lu. 2010. [Automatic analysis of syntactic complexity in second language writing](#). *International Journal of Corpus Linguistics*, 15(4):474–496.
- Xiaofei Lu. 2011. [A corpus-based evaluation of syntactic complexity measures as indices of college-level esl writers' language development](#). *TESOL Quarterly*, 45(1):36–62.
- Paul Meara and Huw Bell. 2001. [P-lex: A simple and effective way of describing the lexical characteristics of short L2 tests](#). *Prospect*, 16(3):5–19.
- Jennifer Meyer, Thorben Jansen, Ronja Schiller, Lucas W Liebenow, Marlene Steinbach, Andrea Horbach, and Johanna Fleckenstein. 2024. [Using llms to bring evidence-based feedback into the classroom: Ai-generated feedback increases secondary students' text revision, motivation, and positive emotions](#). *Computers and Education: Artificial Intelligence*, 6:100199.
- Paul Nation and David Beglar. 2007. [A vocabulary size test](#). *The Language Teacher*, 31(7):9–13.
- Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. 1998. [The university of south florida word association, rhyme, and word fragment norms](#). *Behavior Research Methods, Instruments, Computers*, 36(3):402–407.
- Koji Osawa. 2024. [Integrating automated written corrective feedback into e-portfolios for second language writing: Notion and notion ai](#). *RELC Journal*, 55(3):881–887.
- Hania Salter-Dvorak. 2019. [Proofreading: How de facto language policies create social inequality for L2 master's students in uk universities](#). *Journal of English for Academic Purposes*, 39:119–131.
- Roland Schäfer and Felix Bildhauer. 2012. [Building large corpora from the web using a new efficient tool chain](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul, Turkey. European Language Resources Association (ELRA).
- Meredith A Shafto. 2015. [Proofreading in young and older adults: The effect of error category and comprehension difficulty](#). *International Journal of Environmental Research and Public Health*, 12(11):14445–14460.
- Yanfang Su, Yun Lin, and Chun Lai. 2023. [Collaborating with chatgpt in argumentative writing classrooms](#). *Assessing Writing*, 57:100752.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. [Llama: Open and efficient foundation language models](#). *arXiv preprint arXiv:2302.13971*.
- Joan Turner. 2011. [Rewriting writing in higher education: The contested spaces of proofreading](#). *Studies in Higher Education*, 36(4):427–440.
- Joan Turner. 2024. [Afterword: Revisiting the boundaries of editing and proofreading](#). In Nigel Harwood, editor, *Proofreading and Editing in Student and Research Publication Contexts: International Perspectives*, pages 221–233. Routledge, London.
- Mark Warschauer, Waverly Tseng, Soobin Yim, Thomas Webster, Sharin Jacob, Qian Du, and Tamara Tate. 2023. [The affordances and contradictions of ai-generated text for writers of english as a second or foreign language](#). *Journal of Second Language Writing*, 62:101071.
- Joshua Wilson, Natalie G Olinghouse, and Gilbert N Andrada. 2014. [Does automated feedback improve writing quality?](#) *Learning Disabilities: A Contemporary Journal*, 12(1):93–118.
- Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. [Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark](#). *arXiv preprint arXiv:2303.13648*.
- 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*, pages 51–56.

- Da Yan and Shuxian Zhang. 2024. L2 writer engagement with automated written corrective feedback provided by chatgpt: A mixed-method multiple case study. *Humanities and Social Sciences Communications*, 11(1):1–14.
- Dan Zhao. 2024. The impact of ai-enhanced natural language processing tools on writing proficiency: An analysis of language precision, content summarization, and creative writing facilitation. *Education and Information Technologies*, 30(6):8055–8086.
- Min Zou and Liang Huang. 2024. The impact of chatgpt on l2 writing and expected responses: Voice from doctoral students. *Education and Information Technologies*, 29(11):13201–13219.

A Descriptions of the selected indices

Index	Description
Lexical indices	
n _{types}	Counts the number of unique words, taking into account their part-of-speech.
m _{attr}	Computes the type-token ratio over a 50-word sliding window.
b _{concreteness}	Uses psycholinguistic norms to assess word concreteness across categories based on large-scale ratings, indicating how tangible or abstract a word is perceived to be (Brysbaert et al., 2014).
usf	Measures the number of distinct stimuli that elicit a target word in a word association experiment; lower USF scores suggest the use of words that are more contextually distinct (Nelson et al., 1998).
cw _{lemma_freq_log}	Represents the logarithm of lemma frequencies for content words, computed with reference to an English web corpus (Schäfer and Bildhauer, 2012).
raw _{bg_MI}	Calculates raw bigram mutual Information to quantify the strength of association between consecutive words, with higher values indicating a stronger collocational relationship; this is measured against an English web corpus.
Syntactic indices	
m _{ltu}	Measures the average length of T-units, where a T-unit is defined as a main clause plus any subordinate clause(s) attached to it.
all _{clauses}	Counts the total number of clauses in the text (normed by 10,000 words).
nonfinite _{prop}	Computes the proportion of nonfinite clauses (e.g., gerunds, infinitives) relative to the total number of clauses.
np	Counts the total number of noun phrases, highlighting the nominal complexity within sentence structures (normed by 10,000 words).
np _{deps}	Counts the number of internal dependencies within noun phrases (e.g., adjectives, prepositions, coordinations) (normed by 10,000 words).
amod _{dep}	Measures the frequency of adjective modifier dependencies (normed by 10,000 words).
nominalization	Counts the frequency of nominalizations (i.e., words that convert verbs or adjectives into noun forms) identified by tokens containing predefined suffixes such as <i>-al</i> , <i>-ness</i> , among others (normed by 10,000 words).
be _{mv}	Measures the frequency of the verb “be” when used as a main verb (excluding its auxiliary function) (normed by 10,000 words).
past _{tense}	Measures the frequency of past tense verbs (normed by 10,000 words).

MateInfoUB: A Real-World Benchmark for Testing LLMs in Competitive, Multilingual, and Multimodal Educational Tasks

Dumitran Adrian Marius, Theodor-Pierre Moroianu and Bucă Mihnea-Vicentiu

University of Bucharest

Faculty of Mathematics and Computer Science

marius.dumitran@unibuc.ro {theodor.moroianu, mihneavicentiu}@gmail.com

Abstract

The rapid advancement of Large Language Models (LLMs) has transformed various domains, particularly computer science (CS) education. These models exhibit remarkable capabilities in code-related tasks and problem-solving, raising questions about their potential and limitations in advanced CS contexts. This study presents a novel bilingual (English–Romanian) multimodal (text and image) dataset of multiple-choice questions derived from a high-level computer science competition. A particularity of our dataset is that the problems are conceived such that some of them are easier solved using reasoning on paper, while for others writing code is more efficient. We systematically evaluate State of The Art LLMs on this dataset, analyzing their performance on theoretical programming tasks. Our findings reveal the strengths and limitations of current LLMs, including the influence of language choice (English vs. Romanian), providing insights into their applicability in CS education and competition settings. We also address critical ethical considerations surrounding educational integrity and the fairness of assessments in the context of LLM usage. These discussions aim to inform future educational practices and policies. To support further research, our dataset will be made publicly available in both English and Romanian. Additionally, we release an educational application tailored for Romanian students, enabling them to self-assess using the dataset in an interactive and practice-oriented environment.

1 Introduction

In recent years, LLMs have demonstrated revolutionary potential in natural language processing and code generation, enabling applications such as automated code writing systems and algorithmic problem-solving (Raihan et al., 2024; Rasheed et al., 2025). For instance, models like GPT-o3 exhibit remarkable proficiency in code generation

and problem-solving (OpenAI et al., 2025), yet their deployment in high-stakes domains remains constrained by efficiency and reliability challenges.

In the educational domain, LLMs exhibit considerable promise for enabling personalized learning and automating feedback; however, their capacity to manage complex, competition-level programming challenges—particularly in bilingual or non-English contexts—remains underexplored, with emerging critiques questioning their reliability in high-stakes scenarios, such as mathematical reasoning. Recent analyses, such as (Petrov et al., 2025; Mirzadeh et al., 2024; Hendrycks et al., 2021) reveal that LLMs frequently produce plausible-sounding but logically flawed solutions, raising concerns about their suitability for rigorous assessments. While benchmarks like HumanEval (Chen et al., 2021; Yu et al., 2024) and MBPP (Austin et al., 2021) evaluate general coding proficiency, they often neglect pedagogical dynamics, such as adaptive scaffolding for learners or ethical alignment with institutional values. Furthermore, studies caution that deploying LLMs in multilingual environments amplifies risks of semantic misinterpretation and cultural misalignment, necessitating rigorous scrutiny of their pedagogical robustness. (Rystrøm et al., 2025; Marchisio et al., 2024)

Our work aims to address this gap by conducting a rigorous evaluation of LLMs using a bilingual dataset, thus shedding light on their strengths, weaknesses, and the nuances of language-specific performance. Our dataset is uniquely comprised of multiple-choice questions that were originally administered as part of a pre-university exam for prospective students. This setting not only simulates a high-stakes assessment environment, but also provides rich insights into the performance of LLMs on tasks that require both theoretical knowledge and practical application. Our approach allows us to identify key strengths and limitations of state-of-the-art LLMs, highlighting scenarios

where additional context either bolsters performance or introduces redundancy and inefficiency. By dissecting performance variations across languages and problem types, we provide a nuanced understanding of how LLMs navigate complex educational assessments, such as those encountered in advanced computer science competitions and early university admissions. Moreover, our study raises important ethical considerations, as the use of automated assessments in educational settings must balance technological innovation with fairness and academic integrity.

Finally, to encourage further exploration and replication, the bilingual dataset¹ developed through this work will be made publicly available, offering a valuable resource for future research in both educational technology and competitive programming evaluation and an educational application² tailored for Romanian students, enabling them to self-assess using the dataset in an interactive and practice-oriented environment.

Main Contributions

The main contributions of our work can be summarized as follows:

- We introduce a novel **multimodal and bilingual dataset** comprising *Romanian* and *English*. The dataset includes **100 multiple-choice questions**, all enriched with extensive solutions in Romanian. This paper focuses specifically on benchmarking LLM performance on the Multiple Choice Question (MCQ) portion, including its multimodal aspects; the programming problems are provided as part of the dataset release for completeness and future research but are not evaluated here. We consider the evaluation of complex coding problems a distinct challenge requiring separate methodologies.
- Our dataset is uniquely designed so that multiple-choice problems can be solved through either mathematical and algorithmic reasoning or by generating executable Python code. Crucially, the benchmark tasks the LLMs with autonomously determining the most suitable approach—producing either direct answers or executable Python code.

- We provide an open-source **educational application** enabling students to interactively attempt and practice all problems included in our dataset, thereby facilitating practical engagement and learning.

Related Work

The evaluation of LLMs for code generation has advanced significantly, supported by benchmarks that measure functional correctness and problem-solving capability. Seminal datasets such as HumanEval (Chen et al., 2021; Yu et al., 2024) and MBPP (Mostly Basic Python Problems) (Austin et al., 2021) have become standard, focusing on generating standalone code from English-language prompts (Paul et al., 2024). While effective for assessing basic coding abilities, these benchmarks often emphasize isolated tasks, neglecting integrated reasoning, debugging, and pedagogical scaffolding (Fujisawa et al., 2024; Zhang et al., 2024). They also overlook ethical alignment (Abdulhai et al., 2024), which is critical in educational deployments.

Recent datasets attempt to address these gaps. APPS (Hendrycks et al., 2021) and CodeContests (Quan et al., 2025) introduce complex algorithmic problems from competitive programming, pushing models toward more advanced problem-solving. However, these datasets are monolingual and insufficiently capture linguistic diversity (Marchisio et al., 2024), despite growing evidence that non-English prompts introduce semantic errors and cultural misalignment (Rystrøm et al., 2025).

In educational contexts, systems for automated feedback (Sarsa et al., 2022) and personalized tutoring (Wu and Hu, 2023; Petrov et al., 2025) rarely engage with high-stakes scenarios such as programming competitions or university admissions. This leads to concerns about fairness (Mouselinos et al., 2023), academic integrity (Huang et al., 2025), and linguistic exclusion (Gao et al., 2024).

Multilingual benchmarks like DS-1000 (Lai et al., 2022) and MultiPL-E (Cassano et al., 2022) broaden the scope but primarily target English programming tasks rather than bilingual educational assessments. Studies reveal that language choice affects problem comprehension (Moumoula et al., 2025), with LLMs showing systematic bias in non-English settings and often generating plausible yet logically flawed responses (Petrov et al., 2025; Mirzadeh et al., 2024). As a result, emerging frameworks call for pairing benchmarks with fairness audits (Du et al., 2025) and cultural robustness

¹<https://huggingface.co/datasets/EHollower/MateInfoUB>

²<https://mateinfo-ub.github.io/>

evaluations (Rystrøm et al., 2025).

Several recent benchmarks have expanded beyond single-turn code generation to include interaction and feedback mechanisms. MINT introduces multi-turn tool use and natural language feedback (Wang et al., 2024), while InterCode and AppWorld emphasize coding with execution feedback and app-driven interaction (Yang et al., 2023; Trivedi et al., 2024). SciCode curates scientific computing tasks (Tian et al., 2024), and XCODEEVAL targets multilingual, multitask code understanding and generation (Khan et al., 2023). However, these benchmarks largely isolate competencies: tool use is decoupled from theoretical reasoning, and scientific or multilingual problems are rarely embedded in pedagogically structured tasks.

Unlike existing benchmarks that focus on isolated coding tasks, our dataset integrates theoretical understanding with practical implementation through hybrid problem formats. Each item in the dataset focuses on one or more core competencies: code synthesis, mathematical reasoning, and algorithmic thinking. This flexible format mirrors the diversity of real-world computer science assessments and addresses the "theoretical blind spots" highlighted by (Chan et al., 2024), where language models struggle when reasoning is detached from implementation. By evaluating symbolic manipulation alongside executable code generation, our dataset offers a more comprehensive measure of educational readiness.

2 Data Collection and Examples

The problem set used in our study is derived from **MateInfoUB**, an annual computer science contest specifically aimed at 12th-grade students. This contest also functions as an admission exam for the Faculty of Mathematics and Computer Science at the University of Bucharest. The competition is structured into two phases:

- **Phase 1:** An online round consisting of challenging multiple-choice questions. Students have access to a programming environment, but are restricted to using only publicly available resources. The use of forums, messengers, or Large Language Models (LLMs) is strictly prohibited.
- **Phase 2:** A live programming contest modeled after the International Olympiad in Informatics (IOI) format, featuring four programming problems. Students' solutions can earn

partial points based on correctness and efficiency.

Our work exclusively focuses on the first phase of the contest, and our dataset is obtained directly from the contest organizers in Romanian, currently also available online³. Extensive solutions accompanying each problem, manually written by the authors and by undergraduate students as part of their academic practice (*practică*) at the university are also available for reference and further research.

Some tasks are accompanied by an image containing a code snippet, a diagram, a graph or similar. For those tasks, we augment the statement with a clear textual description of the image's content.

Our final dataset is composed of the problems with statements and multiple choice answers in Romanian, as well as their direct translation in English. The translations are generated automatically, by using *Gemini 2.0 Flash* with very strict instructions enforcing a verbatim translation. The English translations are then manually checked for correctness.

In the following, we provide two examples that illustrate the characteristics of the dataset.

Example: Multimodal Problem Requiring Visual Analysis

Figure 1 presents a typical multiple-choice question from our dataset. The problem requires determining the number of distinct Minimum Spanning Trees (MSTs) present in the provided graph. Problems of this nature are challenging for LLMs, as solutions depend significantly on visual interpretation and structural observation of the graph. Previous studies have noted similar limitations in visual reasoning tasks performed by LLMs (Liu et al., 2023).

Figure 2 presents another multiple choice question from our dataset. Given a map that illustrates a river with two banks and four islands linked by eight bridges, the task asks for the minimum number of additional bridges that must be built so that a tourist can cross each bridge exactly once. Problems of this nature are challenging for LLMs because they require integrating visual-spatial reasoning with graph-theoretical concepts, such as identifying Eulerian paths, which are not explicitly stated, but must be inferred from the structure of the image or diagram.

³<https://mateinfo-ub.github.io/#/toate-datele>

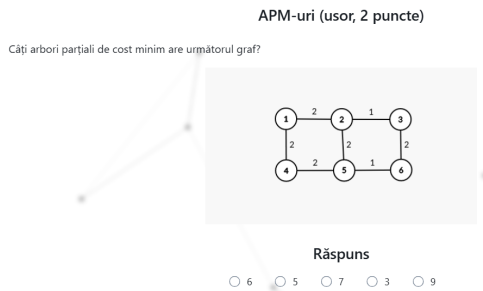


Figure 1: Example multiple-choice problem requiring visual analysis (in Romanian); English translation: "AMP-uri (easy, 2 points); How many minimum spanning trees does the following graph have?"

Koningsberg (easy, 2 points)

The adjacent map is given.

The map represents a river (blue), two banks and four islands (green), as well as eight bridges (black).

What is the minimum number of bridges that need to be built so that a tourist can cross all bridges exactly once?

Careful: the tourist can start his route wherever he wants (on a bank or on an island) and can also finish the route wherever he wants.

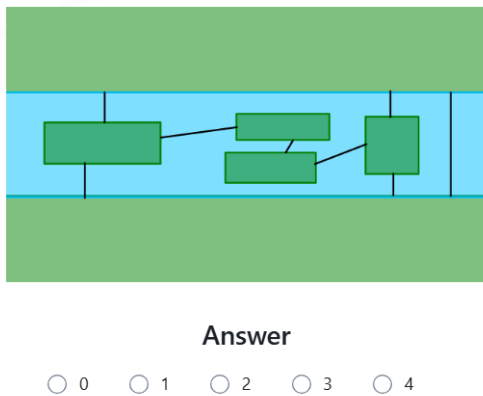


Figure 2: Example multiple-choice problem requiring visual analysis (in English).

3 Benchmarking

In this section, we present a comprehensive overview of our benchmarking strategy designed to evaluate various aspects of Large Language Models (LLMs) performance on our bilingual, multimodal dataset. The benchmarking aims to highlight differences in performance across multiple dimensions, including language, presentation modality, availability of multiple-choice options, and problem-solving approaches.

3.1 Methodology

Our evaluation is based on state-of-the-art LLM models from various vendors, namely *gemini-2.0-flash* and *gemini-2.5-pro-exp-03-25* from *Google AI Studio*, *mistral-large-latest* (April 2025) from *Mistral AI*, and *meta-llama/Llama-3.3-70B-Instruct-Turbo-Free*, *deepseek-ai/DeepSeek-R1* and *deepseek-ai/DeepSeek-V3* from *Together AI* (Together AI, 2025)

We use the models via the exposed API, starting a new chat instance for each task. We provide the models with the task's statement and the multiple-choice answers. We then instruct the models to provide reasoning steps, followed by either an answer or a Python script that computes the answer.

For minimizing benchmarking frictions, we clearly provide the models with the expected output format, which resembles XML. While very forgiving, in some instances, the models fail to adhere to it (if, for instance, their answer exceeds the API's length limit). In such situations, we consider the models' answers incorrect.

3.2 Evaluation Baseline

We evaluate the accuracy of our models on the original tasks. Due to the multiple choice nature of the tasks, verifying the correctness of the models' solutions is trivial. We run each model on each problem 3 times, for minimizing the randomness caused by the LLMs' seed selection. We chose to use the models' default API settings (i.e., without forcing temperature to 0) to reflect typical usage and obtain realistic levels of correctness, confidence, and creativity, as would be experienced by a standard user.

3.3 AI vs. Human Contestants

As our dataset comes from real contests, we compare the performance of the models with the results obtained during the 2021, 2022, 2023, and 2024 editions of the contest. We evaluated the models by measuring their percentile scores compared to the results of the students who qualified for the final stage of the contest.

3.4 Original Romanian Baseline vs. English Translations

LLMs have been notoriously bad at reasoning in languages other than English. By comparing our baseline benchmark with the performance of LLMs on English translations of the statements, we gain

insights about the model’s effectiveness in a language typically underrepresented in NLP research, as opposed to English.

A comparative analysis of English and Romanian benchmarks highlights language-specific challenges and differences in LLM capabilities between languages.

3.5 Original Multiple-Choice vs. No Multiple-Choice Variants

We investigate how the presence or absence of multiple choice answer options affects LLM performance. By removing the multiple-choice framework, we challenge the models’ capability to generate answers without guidance from predefined options.

3.6 Chain-of-Thought vs. Direct Answer

In our benchmarks, when prompting the models for an answer, we ask the models to provide a detailed description of the solution. The models thus respond with reasoning steps to solve the task, followed by the answer.

We measure how the performance of the models is impacted by the absence of the reasoning steps, by prompting the models to directly output the answer, without justifying it.

3.7 Answer-only vs. Hybrid Approach

Finally, we conduct experiments to compare LLM’s performance across two different reasoning strategies:

- **Hybrid approach:** The model autonomously chooses whether to solve the problem via code generation or direct reasoning (our baseline).
- **Think-only:** The model is restricted to providing direct theoretical or conceptual solutions, without the possibility of running *python* code.

We do not consider the third option (forcing the model to produce *python* code), as we experimentally see the model can write a trivial script printing a hard-coded answer, making the experiment uninteresting.

4 Results

In this section, we present the findings of our benchmarking evaluations.

Overall, our analyses suggest significant variations in LLMs performance across different scenarios.

We acknowledge that the outreach of benchmarks might be limited by the size relatively small of our dataset, but our measurements suggest the following trends:

- **Language Comparison:** Our measurements indicate various differences in model accuracy and problem-solving capabilities when problems are presented in Romanian versus English, with some models benefiting from a verbatim translation of the statements to English, a language they are more familiar with, while others perform better when exposed to the original statements.
- **Multiple-choice Contexts:** We observe that the availability of multiple-choice options slightly improves model accuracy compared to scenarios where these options are not provided.
- **Reasoning Strategies:** Benchmarks indicate a promising performance of hybrid strategies, where models autonomously select between code generation and direct reasoning, outperforming exclusive reasoning approaches.
- **Human Performance Comparison:** Comparing the performance of models with the performance of high school students taking part in the contest, we find that newer models consistently outperform most students.
- **Breadth of Capabilities:** Models demonstrate a broader range of problem-solving skills than originally anticipated, effectively addressing a very diverse set of exercises. In particular, of the 100 exercises in our dataset, all but three were solved by at least one model, highlighting their versatility and adaptability to different types of problems.

Detailed findings from each of our experiments are discussed in the following sections.

4.1 Baseline and comparison with human rankings

We find models quite capable, with newer models capable of solving most tasks. On average, models achieve a difficulty-weighted accuracy of 52%

(easy problems are worth 2 points, medium problems 3 points, and hard problems 5 points). A complete breakdown of the accuracy of the models is available in Table 1.

Model	Easy (%)	Medium (%)	Hard (%)
Gemini 2.5 Exp	96.7	85.6	76.7
Gemini 2.0 Flash	71.3	60.0	35.0
Llama 3.3 70B	51.3	31.1	18.3
DeepSeek R1	54.0	34.4	21.7
DeepSeek V3	72.0	63.3	30.0
Mistral Large	62.0	43.3	31.7

Table 1: Performance metrics for different models across difficulty levels in Romanian (larger is better).

One can see that *Gemini 2.5*, a reasoning-focused model, outperforms all others, including *DeepSeek R1*. However, upon closer examination, we found that *DeepSeek R1* frequently produces answers that exceed the API’s maximum response length, leading to truncation and, consequently, incorrect outputs.

In some cases, such as when attempting to manually solve complex counting problems, the model’s output becomes excessively long. We consider it most appropriate to adhere to the API vendor’s configured maximum response length and treat truncated or incomplete answers as incorrect.

As we have data on the scores obtained by students qualified in the *2021*, *2022*, *2023* and *2024* editions of the contest, we can compute the percentile (i.e. the percentage of students doing better than the model) of the qualified students. The results are available in Table 2.

Model	2021	2022	2023	2024
DeepSeek V3	38.22	27.12	26.15	9.95
Gemini 2.5 Exp	0.64	1.69	0.51	0.00
Gemini 2.0 Flash	50.96	55.93	56.41	1.05
Llama 3.3 70B	100.00	100.00	100.00	100.00
DeepSeek R1	100.00	100.00	79.49	100.00
Mistral Large	100.00	92.09	85.64	9.95

Table 2: Average percentiles of models compared to real students across different years (smaller is better).

Models show a constant improvement over the years, which we theorize can be explained by a combination of the following hypotheses:

- Older contests have more ad-hoc problems, which models tend to struggle with.
- Starting in 2022, the UK stopped its financial support for EU students, including Romania.

Thus, many students exploring alternative opportunities took part in the contest. As alternative abroad universities grew in popularity among high school students, interest in the contest could have decreased.

- The student participating in the contest in 2022, 2023 and 2024 were the most impacted by the *Covid-19* remote studying mandates during early high school.

We strongly believe that our dataset is not tainted (i.e., that models were not trained on it). While the statements were publicly available before we started our research, we are the first to compile a dataset that maps these problems to their corresponding answers.

In other words, while it is plausible that model training corpora may have included the raw statements, the solutions could not have been included, since all tasks are original, and our dataset is the first to pair them with verified multiple-choice answers.

4.2 Original language vs. English translation

Our experiments show that most models perform better on the Romanian version of the questions than on the English one. This gap likely arises from two factors. First, our English statements are verbatim translations of the original Romanian text, which can lose nuance and clarity and introduce artifacts (translationese) that impair model understanding.

Second, since the raw Romanian problems were publicly available before our work, it is plausible that models encountered those during training, whereas our English translations are novel; thus, they effectively function as a partial unseen validation set. We therefore consider the benchmark bilingual, while acknowledging that translation quality and prior exposure may both contribute to the observed performance drop.

The *DeepSeek* family of models sees a 10% gain in accuracy when solving the English variant, suggesting reduced multilingual abilities of the models.

The results of the experiment are available in Table 3.

4.3 Multiple-choice options provided vs. not provided

We observe a slight decline in performance when the multiple choice variants are not provided, which

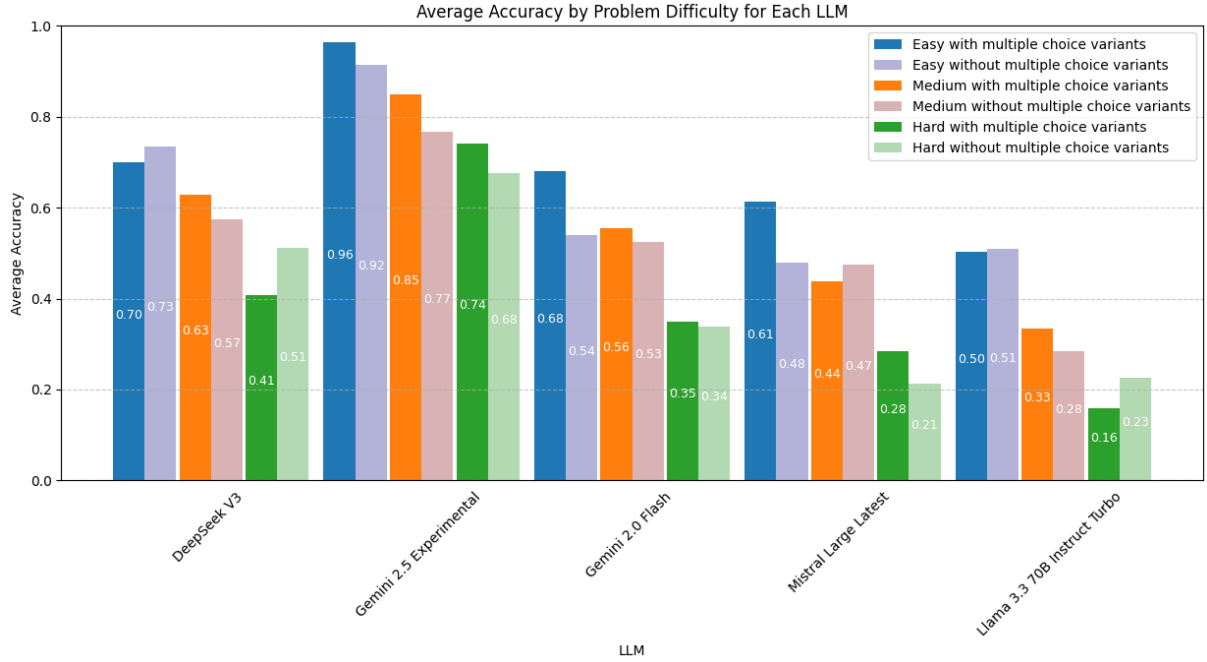


Figure 3: Performance of models when the multiple choice options are provided vs when they are not.

Model	English (%)	Romanian (%)
DeepSeek V3	0.61	0.55
Gemini 2.5 Exp	0.84	0.86
Gemini 2.0 Flash	0.50	0.55
Llama 3.3 70B	0.33	0.34
DeepSeek R1	0.52	0.37
Mistral Large	0.43	0.46
Overall Average	0.54	0.52

Table 3: Average scores for models across different languages.

aligns with the contests’ design goals of making the variants unhelpful for solving the tasks. Considering that models tend to *guess* an answer and hallucinate a justification when they cannot solve the task, we believe that the difference is caused by models having a higher chance of *guessing* the correct answer. The results can be seen in Figure 3.

4.4 Chain-of-Thought vs. Direct Answer

We observe a slight decline in performance when models are only prompted for the answer, as opposed to first providing a justification, or reasoning. While we expect *reasoning* models like *Gemini 2.5 Exp* and *DeepSeek-R1* to be invariant to the change (due to their own reasoning process), *DeepSeek-R1*’s internal chain-of-thought reasoning increases in length, which causes some of its answers to be truncated and invalidated. The full results are available in Table 4.

Model	Easy	Medium	Hard	Average
With Reasoning				
DeepSeek V3	0.70	0.63	0.41	0.58
Gemini 2.5 Exp	0.96	0.85	0.74	0.85
Gemini 2.0 Flash	0.68	0.56	0.35	0.53
Llama 3.3 70B	0.50	0.33	0.16	0.33
DeepSeek R1	0.65	0.43	0.26	0.45
Mistral Large	0.61	0.44	0.28	0.45
Overall Average				0.53
Without Reasoning				
DeepSeek V3	0.73	0.54	0.48	0.58
Gemini 2.5 Exp	0.95	0.81	0.79	0.85
Gemini 2.0 Flash	0.56	0.50	0.11	0.39
Llama 3.3 70B	0.53	0.27	0.19	0.33
DeepSeek R1	0.28	0.12	0.00	0.13
Mistral Large	0.48	0.35	0.15	0.33
Overall Average				0.43

Table 4: Comparison of average scores with and without reasoning for various models.

4.5 Answer-only vs. Hybrid Approach

In Figure 4, we can see the distribution of *python* answers and direct answers, when the models can choose unconstrained how to provide an answer for a given task (i.e., models can freely pick between providing a direct answer and providing a *python* code).

When *Python* is no longer allowed, the models perform unexpectedly worse, as more than half of their answers rely on executing *Python* code. We

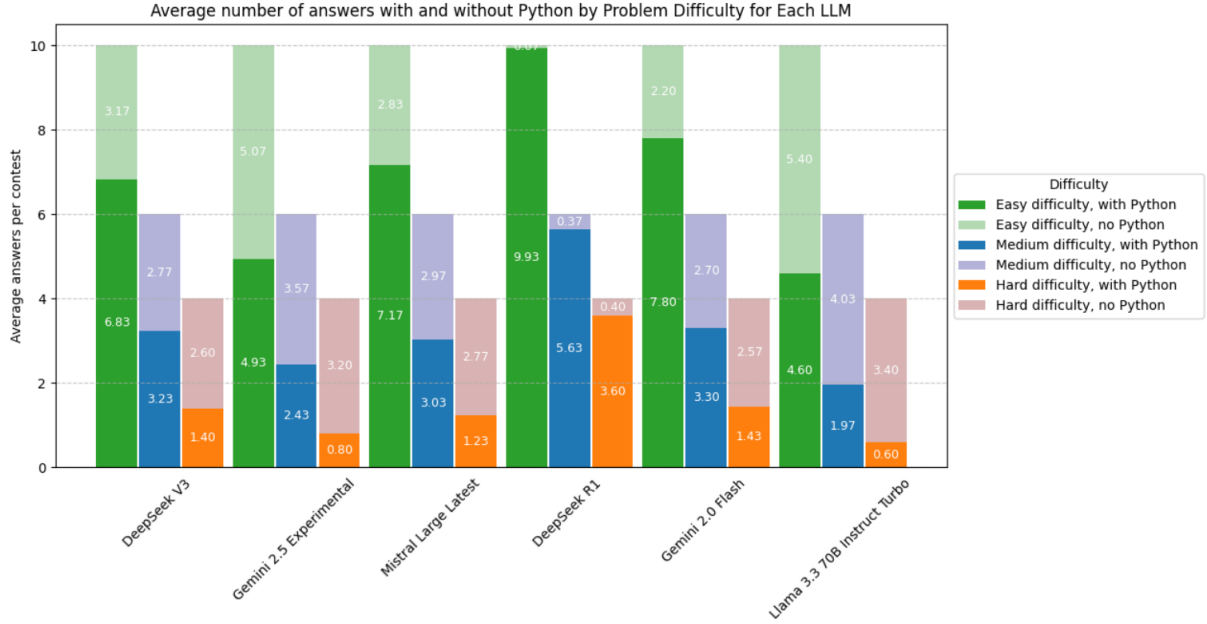


Figure 4: Percentage of direct answers vs *python* answers when the models are given the choice between the two. Columns are scaled the number of tasks of each difficulty (10 easy, 6 medium, 4 hard).

run the experiment on a subset of the models, and the results are available in Table 5.

Model	Easy	Medium	Hard
Python Code Allowed			
DeepSeek V3	0.70	0.63	0.41
Gemini 2.0 Flash	0.68	0.56	0.35
Python Code Not Allowed			
DeepSeek V3	0.65	0.48	0.28
Gemini 2.0 Flash	0.59	0.43	0.20

Table 5: Comparison of average scores for DeepSeek V3 and Gemini 2.0 Flash with and without Python code.

4.6 Discussion of Benchmark Results

Our findings have two key implications for computer-science education. First, assessments should combine tool-enabled tasks with those requiring scaffolding and manual reasoning to accurately gauge student mastery. Second, instructors and contest organizers should monitor for anomalous solution patterns—such as perfectly formatted code or implausibly high confidence scores—to detect unauthorized LLM use.

We also provide descriptive plots covering the experiments contained in our research. They are available in Appendix B.

5 Application

In parallel with our experiments, and inspired by our dataset, we develop a web-based application, which can be used as a training ground of students looking to compete in the contest.

The application is freely accessible online ⁴ and, among others, allows students to:

- Preview the statements of all editions of the contest.
- Simulate an edition of the contest.
- Automatically grade their attempt.

The application is implemented in *React*, and is hosted on *Github Pages*. Due to its limited functionalities, it does not require any kind of dynamic backend, and all of its assets, including statements, solutions, and images, can be packaged statically, making deployment easier.

Screenshots of the application and a description of its functionalities are available in the Appendix A.

6 Future Work

Several avenues for further research are highlighted by our preliminary findings and current limitations. Key directions include:

⁴<https://mateinfo-ub.github.io/>

- **Expanded Benchmarking:** Conducting extensive experiments involving additional competitive programming datasets (potentially using cross-validation) and further expanding multilingual analyses. This could also involve analyzing model performance over varying levels of intrinsic **problem complexity**.
- **Live Contest-Based Evaluation:** Introducing the second phase featuring a plethora of programming contests modeled after the International Olympiad in Informatics (IOI), with multiple problems graded on partial correctness and efficiency. This would enable a deeper analysis of LLMs' algorithmic reasoning and problem-solving capabilities in a structured, task-oriented environment.
- **Fine-Tuning and Contextual Support:** Investigating the impact of fine-tuning LLMs on domain-specific data, leveraging RAG methods, or exploring the effect of providing incremental **contextual hints or scaffolding** to guide model reasoning.
- **Model Efficiency and Scalability:** Exploring methods to optimize model inference times and computational efficiency for real-world educational deployment.
- **Enhanced Ethical Solutions:** Developing and evaluating robust technological and educational solutions that address challenges of academic integrity related to the use of LLM.

Pursuing these directions can deepen the understanding of LLM capabilities and limitations, contributing to their sustainable and ethical integration in education.

Limitations

Although our study provides valuable insights into Large Language Models' (LLMs) performance on bilingual educational assessments, several limitations must be acknowledged.

First, although our dataset features a diverse set of problems from a high-stakes computer science competition, the scope remains limited to the Romanian educational context. Generalization of our findings to other linguistic or educational settings may require additional validation.

Second, our dataset and benchmarks currently focus primarily on the immediate accuracy of LLM-generated solutions. Future work should explore

complementary evaluation metrics, including efficiency, robustness to variations in problem presentations, and detailed error analyses, which would provide deeper insights into model performance and reliability in educational contexts.

Lastly, our benchmarking has not explored the impact of methods such as retrieval-augmented generation (RAG) or fine-tuning of LLMs. Future work incorporating these approaches could reveal further improvements in performance and greater adaptability to specific educational tasks and datasets.

Ethical Considerations

A central motivation for this study is assessing the current capabilities of Large Language Models (LLMs) due to significant ethical challenges posed by their increasing accessibility during online assessments, particularly in competitive contexts such as MateInfoUB. Our benchmarking explicitly aims to identify tasks and problem structures that LLMs struggle to solve reliably. The insights gained allow educators and contest organizers to structure future contests in a way that mitigates unfair advantages gained through unauthorized LLM use.

Although our current findings suggest it remains possible to maintain fairness in online competitions for now by emphasizing problems that LLMs find challenging, this strategy will likely become less effective as LLM capabilities rapidly improve. Therefore, it is increasingly important for educational institutions and competition organizers to proactively adopt technical solutions designed to uphold academic integrity. Such solutions could include software capable of capturing contestant screens, monitoring interactions, and verifying participant authenticity. Additionally, educational efforts should emphasize ethical awareness and responsible technology use, preparing students to navigate the evolving landscape of educational assessments responsibly.

At the same time, we acknowledge that releasing a dataset modeled after real pre-university exams introduces the risk of misuse, particularly fine-tuning LLMs to artificially boost exam performance without genuine understanding. Our benchmark is intended for controlled research and diagnostic evaluation, not as training material for high-stakes testing. Responsible use requires avoiding practices that could compromise the integrity and fairness of educational assessments.

References

- Marwa Abdulhai, Gregory Serapio-García, Clement Crepy, Daria Valter, John Canny, and Natasha Jaques. 2024. [Moral foundations of large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17737–17752, Miami, Florida, USA. Association for Computational Linguistics.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. [Program synthesis with large language models](#). *Preprint*, arXiv:2108.07732.
- Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. 2022. [Multipl-e: A scalable and extensible approach to benchmarking neural code generation](#). *Preprint*, arXiv:2208.08227.
- Jason Chan, Robert Gaizauskas, and Zhixue Zhao. 2024. [Rulebreakers challenge: Revealing a blind spot in large language models’ reasoning with formal logic](#). *Preprint*, arXiv:2410.16502.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. [Evaluating large language models trained on code](#). *Preprint*, arXiv:2107.03374.
- Yongkang Du, Jen tse Huang, Jieyu Zhao, and Lu Lin. 2025. [Faircoder: Evaluating social bias of llms in code generation](#). *Preprint*, arXiv:2501.05396.
- Ippei Fujisawa, Sensho Nobe, Hiroki Seto, Rina Onda, Yoshiaki Uchida, Hiroki Ikoma, Pei-Chun Chien, and Ryota Kanai. 2024. [Procbench: Benchmark for multi-step reasoning and following procedure](#). *Preprint*, arXiv:2410.03117.
- Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2024. [Ambiguity-aware in-context learning with large language models](#). *Preprint*, arXiv:2309.07900.
- 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](#). *Preprint*, arXiv:2103.03874.
- Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, Jian-Yun Nie, and Yang Liu. 2025. [A survey on large language models with multilingualism: Recent advances and new frontiers](#). *Preprint*, arXiv:2405.10936.
- Mohammad Abdullah Matin Khan, M Saiful Bari, Xuan Long Do, Weishi Wang, Md Rizwan Parvez, and Shafiq Joty. 2023. [xcodeeval: A large scale multilingual multitask benchmark for code understanding, generation, translation and retrieval](#). *Preprint*, arXiv:2303.03004.
- Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Scott Wen tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2022. [Ds-1000: A natural and reliable benchmark for data science code generation](#). *Preprint*, arXiv:2211.11501.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. [Visual instruction tuning](#). *Preprint*, arXiv:2304.08485.
- Kelly Marchisio, Wei-Yin Ko, Alexandre Berard, Théo Dehaze, and Sebastian Ruder. 2024. [Understanding and mitigating language confusion in LLMs](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6653–6677, Miami, Florida, USA. Association for Computational Linguistics.
- Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. [Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models](#). *Preprint*, arXiv:2410.05229.
- Micheline Bénédictte Moumoula, Abdoul Kader Kaboré, Jacques Klein, and Tegawendé F. Bissyande. 2025. [Evaluating programming language confusion](#). *Preprint*, arXiv:2503.13620.
- Spyridon Mouselinos, Mateusz Malinowski, and Henryk Michalewski. 2023. [A simple, yet effective approach to finding biases in code generation](#). *Preprint*, arXiv:2211.00609.
- OpenAI, :, Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, and Wenda Zhou. 2025. [Competitive programming with large reasoning models](#). *Preprint*, arXiv:2502.06807.

- Debalina Ghosh Paul, Hong Zhu, and Ian Bayley. 2024. [Benchmarks and metrics for evaluations of code generation: A critical review](#). *Preprint*, arXiv:2406.12655.
- Ivo Petrov, Jasper Dekoninck, Lyuben Baltadzhiev, Maria Drencheva, Kristian Minchev, Mislav Balunović, Nikola Jovanović, and Martin Vechev. 2025. [Proof or bluff? evaluating llms on 2025 usa math olympiad](#). *Preprint*, arXiv:2503.21934.
- Shanghaoran Quan, Jiayi Yang, Bowen Yu, Bo Zheng, Dayiheng Liu, An Yang, Xuancheng Ren, Bofei Gao, Yibo Miao, Yunlong Feng, Zekun Wang, Jian Yang, Zeyu Cui, Yang Fan, Yichang Zhang, Binyuan Hui, and Junyang Lin. 2025. [Codeelo: Benchmarking competition-level code generation of llms with human-comparable elo ratings](#). *Preprint*, arXiv:2501.01257.
- Nishat Raihan, Mohammed Latif Siddiq, Joanna C. S. Santos, and Marcos Zampieri. 2024. [Large language models in computer science education: A systematic literature review](#). *Preprint*, arXiv:2410.16349.
- Zeeshan Rasheed, Muhammad Waseem, Kai Kristian Kemell, Aakash Ahmad, Malik Abdul Sami, Jussi Rasku, Kari Systä, and Pekka Abrahamsson. 2025. [Large language models for code generation: The practitioners perspective](#). *Preprint*, arXiv:2501.16998.
- Jonathan Rystrom, Hannah Rose Kirk, and Scott Hale. 2025. [Multilingual != multicultural: Evaluating gaps between multilingual capabilities and cultural alignment in llms](#). *Preprint*, arXiv:2502.16534.
- Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. [Automatic generation of programming exercises and code explanations using large language models](#). In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, ICER 2022, page 27–43. ACM.
- Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. 2024. [Scicode: A research coding benchmark curated by scientists](#). *Preprint*, arXiv:2407.13168.
- Together AI. 2025. Together ai: The ai acceleration cloud. <https://www.together.ai>.
- Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjana Balasubramanian. 2024. [Appworld: A controllable world of apps and people for benchmarking interactive coding agents](#). *Preprint*, arXiv:2407.18901.
- Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. 2024. [Mint: Evaluating llms in multi-turn interaction with tools and language feedback](#). *Preprint*, arXiv:2309.10691.
- Yangjian Wu and Gang Hu. 2023. [Exploring prompt engineering with GPT language models for document-level machine translation: Insights and findings](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 166–169, Singapore. Association for Computational Linguistics.
- John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. 2023. [Intercode: Standardizing and benchmarking interactive coding with execution feedback](#). *Preprint*, arXiv:2306.14898.
- Zhaojian Yu, Yilun Zhao, Arman Cohan, and Xiaoping Zhang. 2024. [Humaneval pro and mbpp pro: Evaluating large language models on self-invoking code generation](#). *Preprint*, arXiv:2412.21199.
- Shudan Zhang, Hanlin Zhao, Xiao Liu, Qinkai Zheng, Zehan Qi, Xiaotao Gu, Xiaohan Zhang, Yuxiao Dong, and Jie Tang. 2024. [Naturalcodebench: Examining coding performance mismatch on humaneval and natural user prompts](#). *Preprint*, arXiv:2405.04520.

A Online Training Platform

In the image 5 one can see the user interface of the application during a simulation of the 2022 edition of the contest, and the image 6 shows the user interface after the contest's timer ends or the user manually stops it.

While simple, the application contains all the necessary features for an exam-like environment:

- A timer.
- A menu with all of the problems of the contest, ordered by difficulty and color-coded based on the answer provided (blue during the contest and green / red afterwards).
- A problem viewer, where users can read the statements and provide answers.

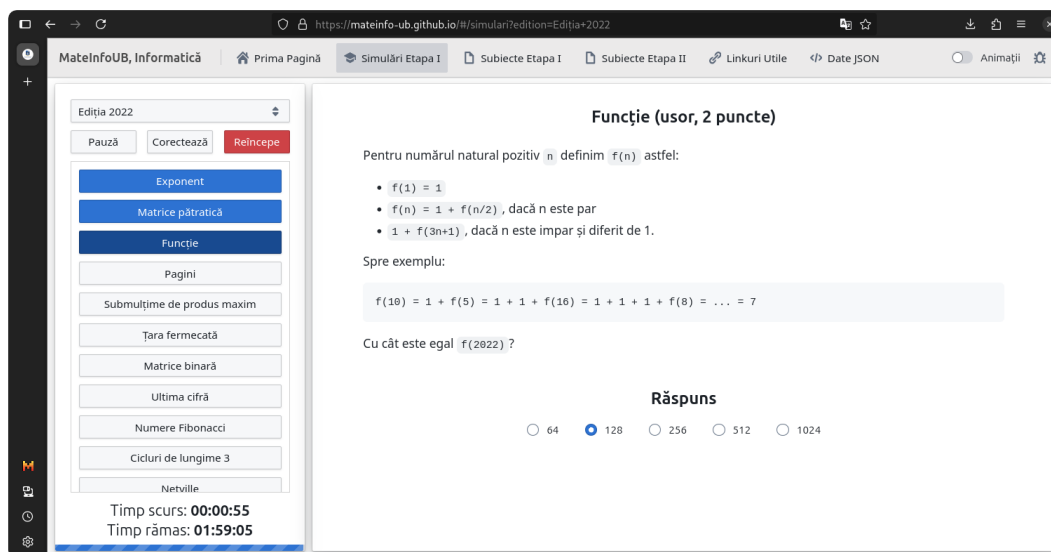


Figure 5: Screenshot of the web application while solving a contest.

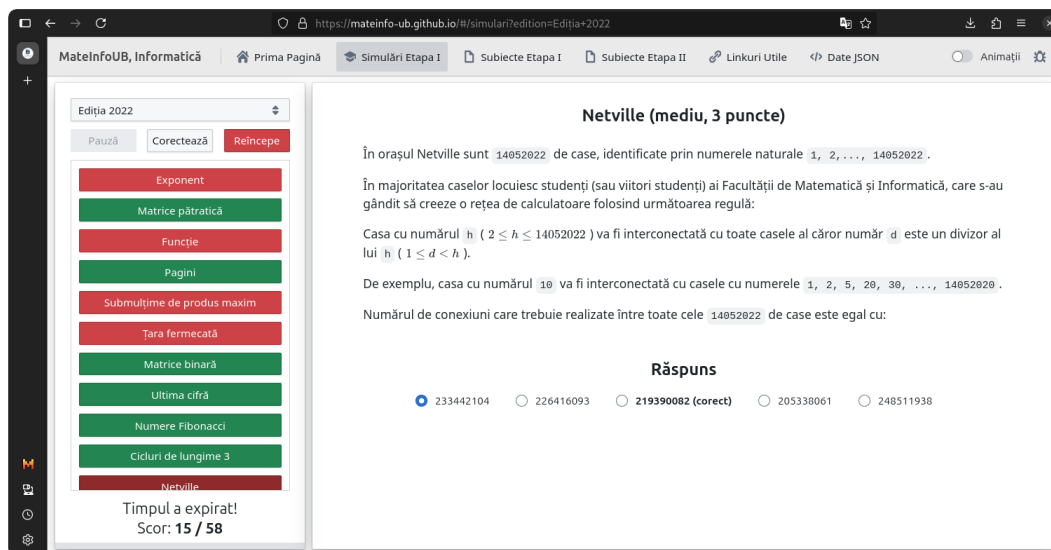


Figure 6: Screenshot of the web application correcting an attempt.

In addition to the simulation page, the application contains pages with the *pdf* statements, exactly as they were during the corresponding exams, and links to useful resources.

B Experiment Plots

In this section, we present visualizations of our experimental results.

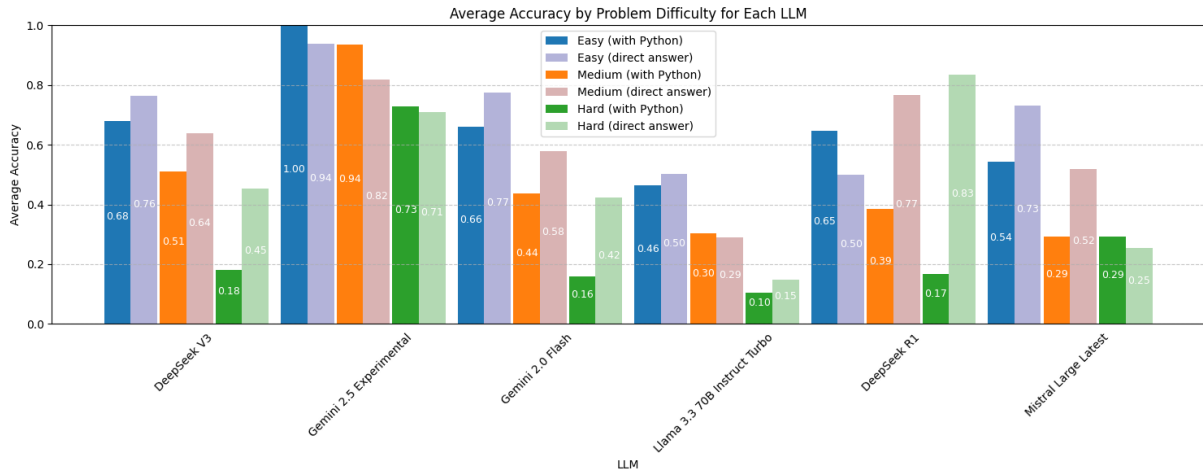


Figure 7: Accuracy of the unconstrained *Python* and direct answers of the models.

In Figure 7, we plot the accuracy of the models when providing an answer as *Python* code or as a direct answer. As the models can freely choose how to answer, we can see some interesting trends. For example, on hard problems, models are significantly more likely to get the right answer when providing *Python* code.

In Figure 8, we plot the percentile of the models when comparing their score with the scores of students advancing to the next phase of the contest, by year. For instance, *Gemini 2.5 Exp* ranks first for 3 out of the 4 years, while *Llama 3.3* ranks last all years.

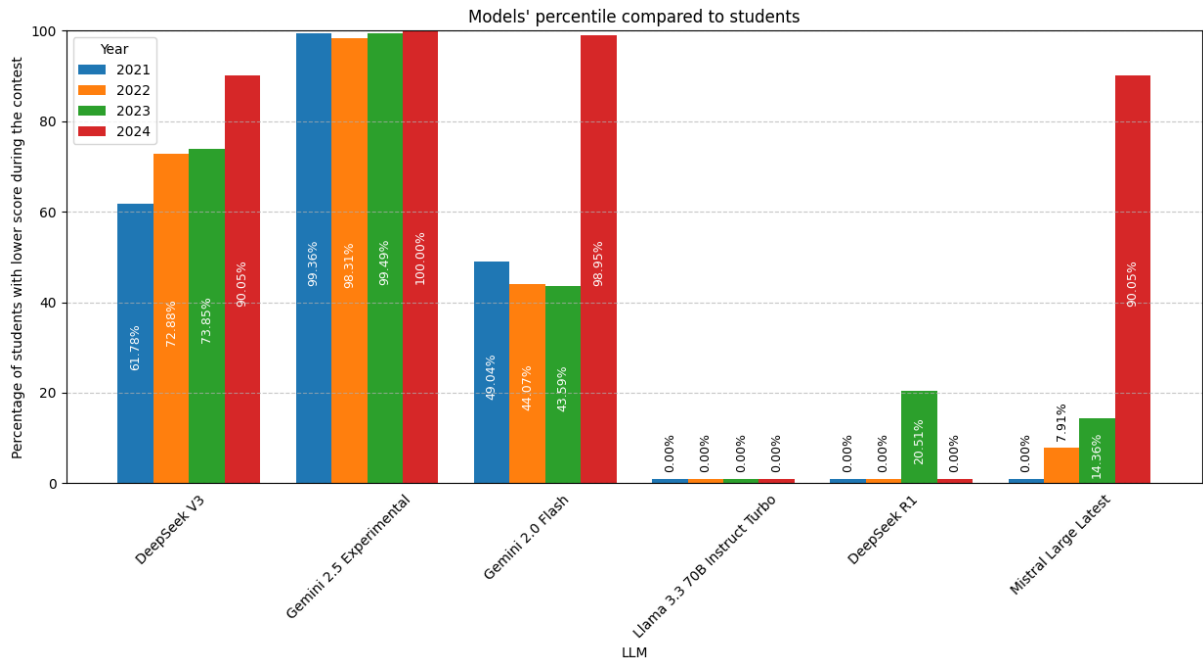


Figure 8: Percentage of students with a lower score than the models, by year.

Figure 9 plots the accuracy of models by language. Except for *DeepSeek-R1*, models tend to achieve a higher score in Romanian, the original language of the tasks.

When prevented to

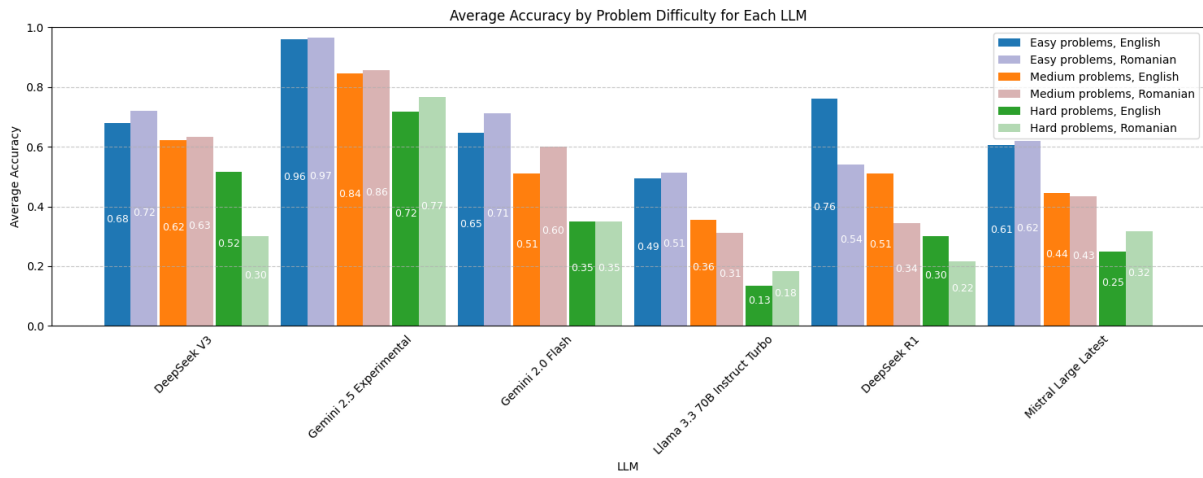


Figure 9: Performance of the models by language

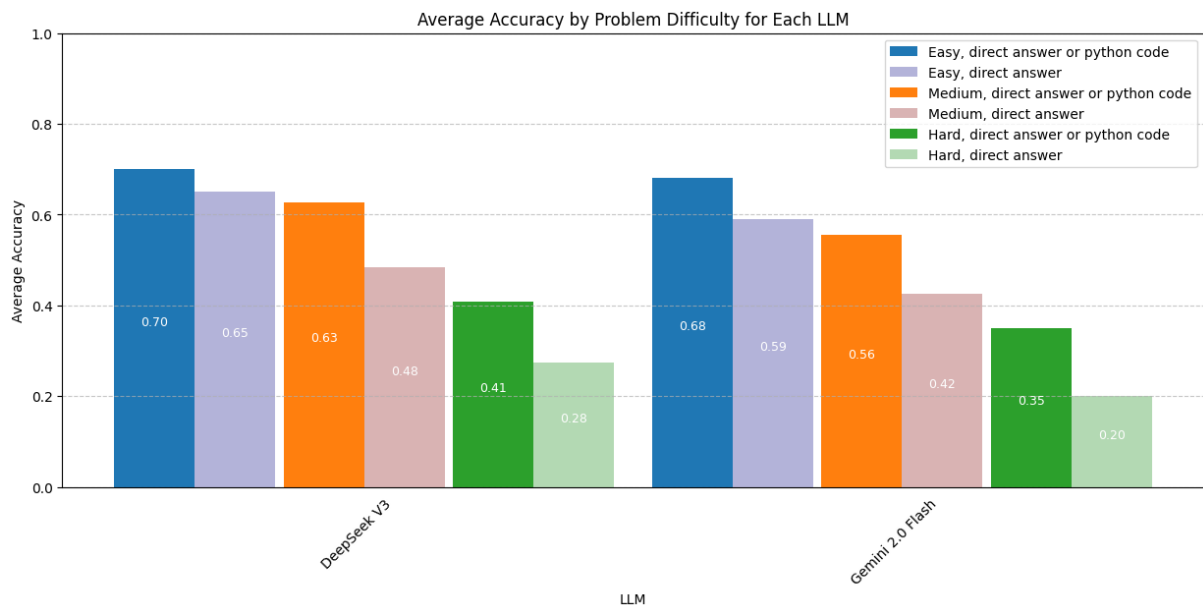


Figure 10: Performance of the models when they are allowed to produce *Python* code and when they have to provide the answer directly.

Unsupervised Automatic Short Answer Grading and Essay Scoring: A Weakly Supervised Explainable Approach

Felipe Urrutia^{1,2} Cristian Buc Roberto Araya² Valentin Barriere¹

¹Universidad de Chile, Department of Computer Science (DCC), Santiago, Chile

²Universidad de Chile, Centro de Investigación Avanzada en Educación (CIAE), Santiago, Chile
furrutia@dim.uchile.cl, cristian.buc@cenia.cl,
roberto.araya.schulz@gmail.com, vbarriere@dcc.uchile.cl

Abstract

Automatic Short Answer Grading (ASAG) refers to automated scoring of open-ended textual responses to specific questions, both in natural language form. In this paper, we propose a method to tackle this task in a setting where annotated data is unavailable. Crucially, our method is competitive with the state-of-the-art while being lighter and interpretable. We crafted a unique dataset containing a highly diverse set of questions and a small amount of answers to these questions; making it more challenging compared to previous tasks. Our method uses weak labels generated from other methods proven to be effective in this task, which are then used to train a white-box (linear) regression based on a few interpretable features. The latter are extracted expert features and learned representations that are interpretable *per se* and aligned with manual labeling. We show the potential of our method by evaluating it on a small annotated portion of the dataset, and demonstrate that its ability compares with that of strong baselines and state-of-the-art methods, comprising an LLM that in contrast to our method comes with a high computational price and an opaque reasoning process. We further validate our model on a public Automatic Essay Scoring dataset in English, and obtained competitive results compared to other unsupervised baselines, outperforming the LLM. To gain further insights of our method, we conducted an interpretability analysis revealing sparse weights in our linear regression model, and alignment between our features and human ratings.¹

1 Introduction

Applications of Large Language Models (LLMs) are emerging in the field of education and have

taken complementary roles to those of teachers (Jeon and Lee, 2023). For instance, LLMs have been used, with mixed results, to train teachers to learn new strategies (Wang and Demszky, 2023). One aspect of education that can greatly benefit of automation is that of grading or scoring (Lan et al., 2024). Such automation could greatly improve the flexibility of teaching and target on the fly specific educational shortcomings of students.

In this work, we focus on two of these automations: (i) **Automatic Short Answer Grading (ASAG)** and (ii) **Automatic Essay Scoring (AES)**; both instances of automated scoring for open-ended questions. More specifically, ASAG focuses on grading short, open-ended responses. These responses are typically a few sentences to a paragraph long and are often fact-based, requiring concise answers. In contrast, AES evaluates longer, more complex pieces of writing, which typically contain an introduction, body, and conclusion, and involve argumentation, analysis, and critical thinking. AES is one of the earliest research problems in natural language processing (Page, 1966, 1967).

One crucial aspect of automated grading on open-ended questions is the ability to interpret the grade. The machine learning community has prioritized increasing explainability in models, leading to the emergence of Explainable AI. This area focuses on building tools to understand the decisions made by learning models (Gunning et al., 2019; Arrieta et al., 2020; Fel et al., 2022), or even advocates for the sole use of white-box models (Rudin, 2019). However, white-box models typically display poorer performances compared with black-box ones (Loyola-Gonzalez, 2019). Thus, in line with the explainable trend, recent methods have focused on developing novel tools to increase the performance of white-box models, sometimes up

¹Code available at [furrutia/unasages-bea2025](https://github.com/furrutia/unasages-bea2025)

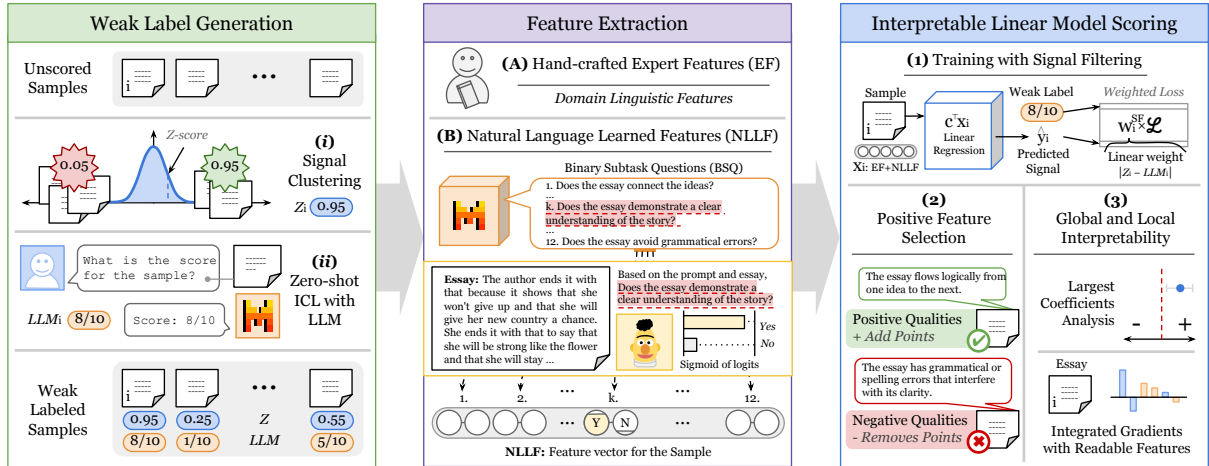


Figure 1: Full process. **Phase 1**, Generation of Weak Labels using Unsupervised methods: Signal Clustering (Chen et al., 2010a) or through an LLM (Jiang et al., 2023b). **Phase 2**, domain Expert Features (EFs) extraction and Natural Language Learned Features (NLLFs) obtained from answers to Binary Subtask Questions (BSQs) (Urrutia et al., 2023). **Phase 3**, feature selection, interpretable model training and analysis.

to that of black-box models (Urrutia et al., 2023).

Finally, most studies rely on supervised learning with annotated datasets (Takano and Ichikawa, 2022; Bonthu et al., 2023; Zhang et al., 2022), where a few items are associated to many annotations. A situation that is barely encountered in real-life scenarios. Moreover, only a few works in this area focus on non-English language (Latif et al., 2024). The majority of them are restrained to English, and none of them in (Latin-American) Spanish.

Motivation and Contributions. In this work, we tackle the issues raised above in a single framework (see Figure 1). First, we propose a method that allows us to reach high performance in ASAG and AES tasks in an unsupervised way. Second, we show the potential of our model to create interpretable white-box predictions based on sparse features, in a setting where strong generalization abilities are required because of highly diverse questions with a few answers.

Therefore our contributions are as follows: (i) we present a novel Non-English language dataset that is particularly challenging for ASAG systems, as it involves many questions with few answers, (ii) we propose a novel framework that unifies unsupervised and supervised methods into a single ASAG/AES system. In particular, we use weak labels from opaque unsupervised methods for supervised learning in white-box models, (iii) we propose a way to maximize the impact of the best-labeled training examples by weighting the loss

function regarding the degree of consensus between each weak label, (iv) we compare our method with strong ASAG and AES baselines on two distinct datasets of different languages, and show that our method significantly outperforms previous white-box models, and falls barely short to LLM-based ASAG or to SOTA AES, (v) we run a thorough analysis on the AES dataset to demonstrate the interpretability of our method by: looking at our model’s sparse weights, comparing it with SOTA using their integrated gradients but also showing our features are aligned with humans scores.

2 Related Work

In the context of ASAG, several methods have been proposed. Recent work has focused on generating understandable scoring by decomposing items (i.e., questions and responses to math problems) into rubrics whose validity can be inferred with language models (Hellman et al., 2023). Similar work have focused on directly fine-tuning pre-trained language models for ASAG (Takano and Ichikawa, 2022; Bonthu et al., 2023; Zhang et al., 2022), or training language models only based on student responses (Steimel and Riordan, 2020). Some works developed a hybrid ASAG system that evaluates answers to mathematical questions based on deterministic methods and the quality of explanations using text-based scoring methods (Cahill et al., 2020). Note that many semi-supervised (Brooks et al., 2014; Weegar and Idestam-almquist, 2024; Basu et al., 2013) or similarity-based methods (Bexte

et al., 2023) allow to use less labels, but they still need some of them.

In the context of AES, Taghipour and Ng (2016) were pioneers in training neural networks for AES, using a CNN-LSTM on the Automated Student Assessment Prize (ASAP; Hamner et al. 2012) dataset. Even though supervised models remain the most efficient (Yang et al., 2020), unsupervised methods like the one we are proposing show promising results. For instance, AESPrompt (Tao et al., 2022) obtains competitive results in one-shot essay scoring using continuous prompt learning. Wang et al. (2023) created a fully unsupervised approach using heuristic signals learning as a proxy task, as ultimate goal to train a BERT-based essay scorer, and obtained state-of-the-art performances on ASAP. Recent works have focused on the ability of LLMs to automatically score the proficiency of written essays on ASAP (Mansour et al., 2024; Lee et al., 2024). Stahl et al. (2024) even proposed prompting strategies for joint essay scoring and feedback generation to gain more interpretability.

Regarding general explainability, techniques that could be used for ASAG and AES such as Chain-of-Thought (CoT) (Wei et al., 2022) can provide a superficial level of explanation but are prone to structural biases in the text that put in question their fidelity (Turpin et al., 2023; Paul et al., 2024). Moreover, these techniques are fragile as pre-trained language models show lack of robustness on adversarial or unusual writing (Lottridge et al., 2023). Importantly, these writing types are often present in the answers of young children like in the ASAG dataset of this study.

3 Methods

The task of automatically assigning scores to short answers/essays involves finding a model M that assigns a score \hat{y}_i between 1 and S_{\max} to each pair of question/answer or instruction/essay. **First**, we use unsupervised methods to create weak labels. **Second**, we represent every document using interpretable features. **Third**, we select features and train a non-negative linear regression model on the weak labels, using a special loss to maximize the weak labels quality. We show the model is both white-box, sparse and interpretable.

Weak-supervision We propose to train an unsupervised model M by leveraging high-level heuristic signals, or weak labels. Our method (see Figure 1, **Phase 1**) involves utilizing two distinct signals:

(i) scores derived from the unsupervised Signal Clustering method (SC; Chen et al. 2010b, see below) and (ii) scores obtained from an LLM using zero-shot in-context learning. For a given question-answer/instruction-essay pair (q_i, a_i) , we denote as Z_i the signal of the answer with SC or LLM_i the LLM-based signal. To weakly-supervise the training of M , we use $y_i = Z_i$ or $y_i = LLM_i$ in order to minimize the loss function $\mathcal{L}(\hat{y}_i, y_i)$.

Signal Clustering (or Z-score) Based on Chen et al. (2010a), this method is simple yet allows for surprisingly good results in unsupervised automatic essay scoring. Basically, it initialize each essay score with a simple value, and then iteratively propagates the scores to other samples in the same cluster. For their essay scoring task, the authors of the original paper used the number of unique terms in the answer as the initial score. It uses the following inductive formula:

Z_{i0} : Initial score for the i -th answer,

$$S_{it} = \sum_{j \neq i} \text{Sim}_{ij} \cdot Z_{i(t-1)},$$

$$Z_{it} = \frac{S_{it} - \frac{1}{N-1} \sum_{k \neq i} S_{kt}}{\sigma_t},$$

where S_{it} is the score for the i -th answer at step t , Sim_{ij} is the similarity between the i -th answer and the j -th answer, and Z_{it} is the Z -score of the i -th with σ_t the standard deviation of $S_{.t}$ at step t . We call Z_i the Z -score of the i -th answer at final step. We update Z_i until convergence.

Interpretable Features Following the work of Urrutia et al. (2023), we incorporated a set of expert-derived features (EF) coming from expert domain knowledge, and also high-level explainable features such as Natural Language Learned Features (NLLF; Urrutia et al. 2023). NLLFs encode answers to simpler-than-the-task binary questions, called **Binary Subtask Questions (BSQs)**, into a human-readable feature vector. It allows the model to represent each sample as a vector of probabilities on other interpretable simpler sub-tasks, like "Is the answer written clearly and concisely?". More details are available in (Urrutia et al., 2023) and in Appendix B. We also use the concatenation of both type of features (EF+NLLF). For EF, we use in ASAG/AES a list of 36/14 hand-crafted features, to describe the answers to math questions/essays (Table 6/7 in Appendix). Figure 1 shows the feature

Question	Answers	Score
Don Antonio bought 3 boxes of cereal at \$673 each. The seller charged him \$2100. Is what they charged him correct? Explain in your own words.	If Don Antonio bought 3 boxes, it's fine. No, because he should be charged less. It's not wrong, I got 2019.	{2, 3} {4, 3} {6, 7}

Figure 2: Examples of a Question, Answers and Scores from our ASAG dataset. Translated from Spanish.

extraction in **Phase 2**, with an example of BSQ and the NLLF vector for an essay.

Interpretable Model: Linear Regression We trained a linear regression on two types of weak labels (see Figure 1, **Phase 3**).

Signal Filtering We propose a method to maximize the impact of well-labeled examples through the weighting of the loss function with respect to the degree of consensus among weak labels (see Figure 1, **Phase 3**). Basically, we compute linear weights utilizing the difference between the predicted scores generated by the LLM and the ones derived from the Signal Clustering method, both of which obtained in a unsupervised way. For a given question-answer pair (q_i, a_i) and weak-label $y_i \in \{Z_i, \text{LLM}_i\}$, the weighted loss is $w_i^{\text{SF}} \cdot \mathcal{L}(\hat{y}_i, y_i)$, where:

$$w_i^{\text{SF}} = 1 - \frac{|Z_i - \text{LLM}_i|}{S_{\max} - S_{\min}}$$

Feature Selection In order to keep our model interpretable, we used two tricks (see Figure 1, **Phase 3**). First, we only chose BSQs formulation that were positively correlated with the score of the student² i.e., describing events that were seen as positive by the teacher. Second, we forced the linear regression model to learn only positive weights (Slawski and Hein, 2013) as they are applied on features that are positives w.r.t. the score. Section 5 shows that this setting allows for sparsity in the parameters space of the linear regression model.

4 Experiment and Results

4.1 Datasets and Evaluation Metrics

We ran experiments on two distinct tasks using two datasets in different languages. The first set of experiments (Section 4.1.1) tackles ASAG in Spanish while the second set of experiments (Section 4.1.2) tackles AES in English.

²using weak labels

Task	Genre	Avg. Length	Score Range	# Essays
1		350	2-12	1783
2	PER	350	1-6	1800
3		150	0-3	1726
4		150	0-3	1772
5	SDE	150	0-4	1805
6		150	0-4	1800
7	NAR	250	0-30	1569
8		650	0-60	723

Table 1: Properties of the different tasks in the AES dataset called ASAG. Genre: PER (persuasive), SDE (source-dependent), NAR (narrative).

4.1.1 Automatic Short Answer Grading in Spanish

The dataset comprises written answers from fourth-grade students to mathematics questions. The question-answers pairs were collected using the online e-learning platform ConectaIdeas, which is currently deployed and use by teachers and students in Chile. Its data was already used in past scientific studies (Urrutia Vargas and Araya, 2023; Urrutia and Araya, 2023). It encompasses a total of 63,612 answers to 1,248 unique questions collected across two academic years. The answers were obtained from a total of 3,463 fourth-grade students, with 231 for the 2017 period and 3,232 for the 2022 period. The answers have on average a total of 50 characters. Each question has on average a total of 52 answers per question for 2022 and 30 for 2017.

The data are annotated based on the scoring of answers for one academic year (2017). Answers from the unlabeled academic year are utilized to train automatic systems, while those from the labeled academic year serve as a test set for evaluating the performance of these systems. Annotation was conducted by two elementary mathematics teachers, assigning scores ranging from 1 to 7 (i.e., from insufficient to excellent). Only the scores from one teacher were utilized as the ground-truth, while the scores from the other teacher were utilized to analyze human performance, in this sense we can make a model that predicts the grading behavior of one teacher. We calculate the agreement between their scores and obtained a Correlation of

Method	Weak Signal	Signal Filt.	Text	EF	NLLF	EF + NLLF
Length	None	-	.2734	-	-	-
Jaccard Sim.	None	-	.2758	-	-	-
Cosine Sim.	None	-	.3759	-	-	-
ULRA	LF	-	.5112	-	-	-
	EF	-	.4264	-	-	-
	EF+LF	-	.4218	-	-	-
Z-score	None	-	.5104	-	-	-
LLM	None	-	.5727	-	-	-
LLM-CoT	None	-	.4778	-	-	-
Linear Regression	Z-score	✗	-	.4937	.3853	.5096
	LLM-based signal	✗	-	.4815	.3538	.4312
	Z-score	✓	-	.5018	.3899	.5450
	LLM-based signal	✓	-	.4974	.3712	.4791
BERT	Z-score	✗	.5220	-	-	-
	LLM-based signal	✗	.5085	-	-	-
	Z-score	✓	.5280	-	-	-
	LLM-based signal	✓	.5430	-	-	-
Human	None	-	.7568	-	-	-

Table 2: Results of the ASAG models using Pearson correlation: the cheap baselines using similarity, the ULRAs using different weak linguistics signals, the Z-score and LLM predictions, and our weakly supervised models. For the weakly supervised models, the linear model utilizes all combinations of two feature sets (EF and NLLF), while the BERT model is trained on text data.

.76. Figure 2 shows an example of the dataset.

4.1.2 Automatic Essay Scoring in English

We ran experiments using the Automated Student Assessment Prize³ (ASAP) dataset (Hamner et al., 2012). This dataset has been widely used in several AES studies (Xie et al., 2022; Jiang et al., 2023b; Muangkammuen and Fukumoto, 2020; Mansour et al., 2024; Mathias and Bhattacharyya, 2018). For instance, it has been used by Wang et al. (2023) to assess the ULRA model for an unsupervised AES task. It is composed of 12,978 essays divided into 8 different sets. Each of the sets corresponds to a specific essay task or prompt, which can be seen as domain. The tasks are of different genres: persuasive, source-dependent response, and narrative. The statistics of the dataset is shown in Table 1.

As a validation metrics, we report Quadratic Weighted Kappa (QWK) in order to compare the different models, generally utilized to measure the agreement between groundtruth scores and predicted scores on this dataset and in AES research.

4.2 Baselines

Dummy Baseline We use a regression model based on the answer length in terms of number of characters.

Similarity Measures We calculate the similarity between the question and the answer to assess

its correctness based on the shared information between them. We use two methods: Jaccard Similarity on sparse embeddings (Bag-of-Words; (Harris, 1954)), and cosine similarity with dense vectors obtained from the [CLS] token of a multilingual Sentence Transformer (Reimers and Gurevych, 2019).

Signal Clustering (Z-score) Based on Chen et al., we use answer length as the initial scoring and assessed answer similarity based on the shared terms between two answers.

Mixtral We used a recent LLM to address the task in a zero-shot format (Jiang et al., 2023a), using a simple prompt containing the definition of the task. More details in Appendix C.

ULRA We implemented the unsupervised ULRA method of Wang et al. (2023) which showed state-of-the-art results on Automated Essay Scoring, which is close to ASAG. This model consists of using multiple quality signals obtained from heuristics as the pseudo-groundtruth, and then training a neural model by learning from the aggregation of these signals. The idea is that the final score should depend on an aggregation of these simple signals. For the ASAG task, we adapt the method translating the original Linguistic Features (LF) to Spanish, and by using our own Expert Features (EF) as pseudo-groundtruth. For the AES task, we utilized the LF from the original paper on the same task. Note that ULRA was also considered as a

³<https://www.kaggle.com/c/asap-aes>

Method	Weak Signal	Signal Filt.	Text	EF	NLLF	EF + NLLF
Length	None	-	.3893	-	-	-
Jaccard Sim.	None	-	-.1821	-	-	-
Cosine Sim.	None	-	.0237	-	-	-
ULRA (Wang et al., 2023)	LF	-	.6423	-	-	-
Z-score (Chen et al., 2010b)	None	-	.5809	-	-	-
LLM (Jiang et al., 2023a)	None	-	.5119	-	-	-
LLM-CoT (Wei et al., 2022)	None	-	.4152	-	-	-
MTS [†] (Lee et al., 2024)	None	-	.550	-	-	-
Linear Regression	Z-Score	✗	-	.5528	.6083	.6141
	LLM-based signal	✗	-	.5762	.5720	.6385
	Z-score	✓	-	.5603	.6123	.6255
	LLM-based signal	✓	-	.5797	.5814	.6451
BERT	Z-Score	✗	.5728	-	-	-
	LLM-based signal	✗	.4418	-	-	-
	Z-score	✓	.5764	-	-	-
	LLM-based signal	✓	.4781	-	-	-
AES-Prompt [†] (one-shot) (Tao et al., 2022)	None	-	.639	-	-	-
R ² -BERT [†] (supervised) (Yang et al., 2020)	None	-	.794	-	-	-
Human	None	-	.7384	-	-	-

Table 3: Results of the models on the AES task using the average of the QWK over the different essay tasks. We report the cheap baselines using similarity, ULRAs using different weak linguistics signals, the Z-score and LLM predictions, and our weakly supervised models. Human scores were re-calculated here. [†] From original papers.

weak label generation method, but did not generate favorable results.

Weakly supervised BERT We evaluated different BERT models (Devlin et al., 2019) with a regression head on top of the [CLS] vector to predict the weak signals. For the ASAG task, we used BETO, a Spanish BERT transformer (Cañete et al., 2023). For the AES task, we used an English BERT.⁴

Multi-trait Specialization We compare with the work of Lee et al. (2024), who proposed an unsupervised method using LLMs to predict the quality of essays in a zero-shot way. Their method learns to decompose the writing proficiency into distinct traits, as some are known to be useful for judging global essay quality (Ke and Ng, 2019) such as *Position* and *Thesis Clarity*, *Organization and Structure* or *Supporting Details and Evidence*.

4.3 Experimental Protocol

The transformers library (Wolf et al., 2019) was used to access the pre-trained model and to train our models. We used BETO as backbone for NLLF generation, and the 4-bit version of Mixtral-8x7b⁵ as LLM. The linear regressions were trained using scikit-learn (Pedregosa et al., 2012). We standardized every features before the

⁴bert-base-cased, bert-base-spanish-wwm-cased

⁵mistralai/Mixtral-8x7B-Instruct-v0.1

logistic regression. For the ASAG task, Pearson correlation measured the correlation between predicted scores from automatic models and ground-truth scores from one teacher. We evaluated our model on the 1,315 manually annotated examples. For the AES task, we randomly split the data into a training, a validation and a test sets following the proportion 60/20/20 like Wang et al. (2023).

4.4 Results

4.4.1 Results on ASAG

Table 2 shows the results of the different baselines and models. It is notable that naive baselines like a linear regression using the answer length can reach a correlation of .27, and are surpassed by similarity between answer and question using a sentence-bert. Best machine results (.57) are obtained with an LLM, surprisingly without using the CoT mechanism, but still far away from human performances (.75). All our weakly supervised approaches benefit from the Signal Filtering method. Adding NLLF to our method helps when using Z-value or the LLM output as weak label, allowing to reach a score close to the one of the LLM, but with an interpretable white-box algorithm (contrary to BERT). ULRA methods, using general and/or domain expert features, tend to display lower scores when compared with Signal Clustering and remaining methods in this task. Finally, the scores of the BERT model trained with the weak-labels are im-

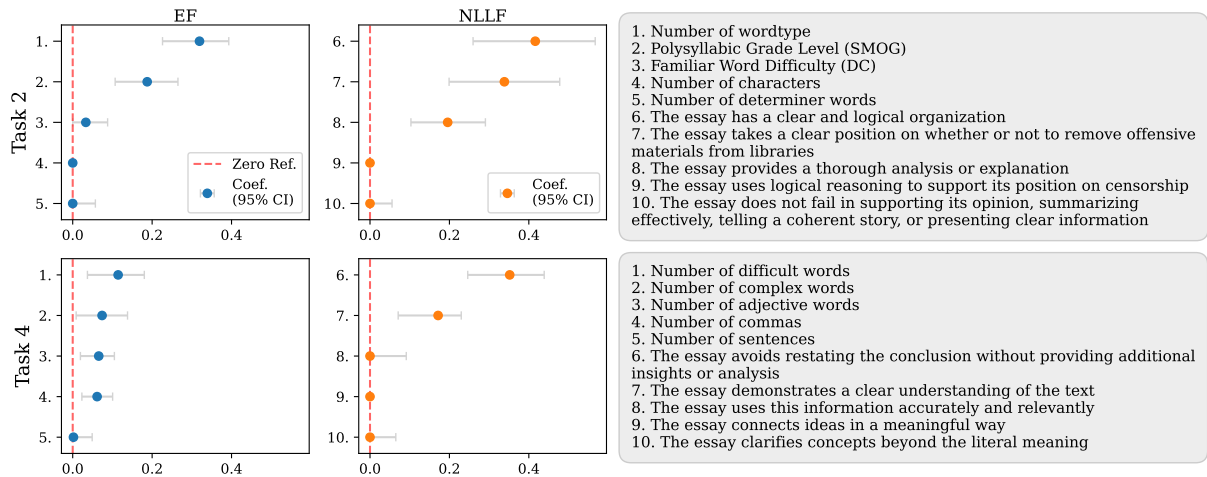


Figure 3: Highest coefficients of the Linear Regression with Signal Filtering using EF+NLLF on AES tasks 2 and 4. The box represents the 95% confidence interval. Biases are respectively of 3.37 and 1.35 for tasks 2 and 4.

proved when applying Signal Filtering, with the best one of .543 using an LLM-based signal as weak labels.

4.4.2 Results on AES

Table 3 shows the results of the different baselines and models on the AES task. Simple baselines achieve a moderate correlation (e.g., .4785 when using the answer length), while basic similarity measures, such as Jaccard and Cosine, perform poorly, with negative or near-zero correlations. Among the models, our method achieves the highest score (.645), outperforming other methods such as ULRA (.642), Z-score (.581), and LLM (.512), though all falling short of human-level performance (.738). Interestingly, the LLM with a CoT approach performs worse than the standard LLM, with a correlation of only .415, which is unexpected given the reported success of CoT in other contexts, specially for a task such as essay scoring in English. Notably, all of our weakly supervised models benefit significantly from the Signal Filtering method. Furthermore, adding the NLLF mechanism further enhances performance. Indeed, combining LLM-based labels, Signal Filtering, and NLLF reaches the highest performance, outperforming prompt engineering baselines such as MTS or AES-Prompt. Finally, the BERT models trained with the weak-labels display lower scores (highest BERT score of .577 using Z-score and Signal Filtering). As a way to cross-check our results, existing works assessed the capacity of various LLMs on this tasks and dataset (Mansour et al., 2024; Lee et al., 2024). The performances we obtained (QWK of 0.51), are

in line with the ones reported in Lee et al. (2024)⁶, but higher than the ones reported in Mansour et al. (2024).

5 Analysis

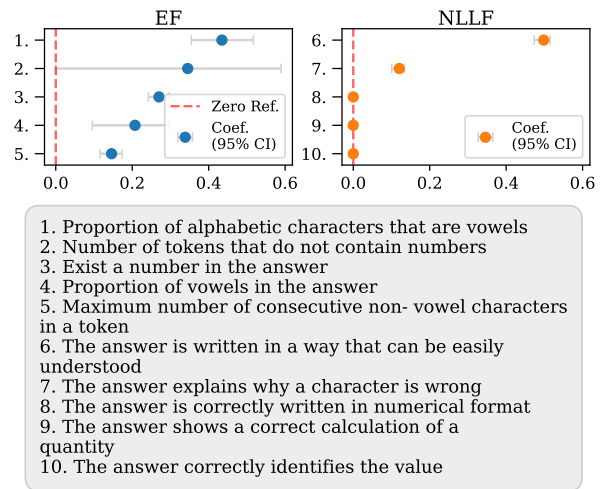


Figure 4: Highest coefficients of the Linear Regression with Signal Filtering using EF+NLLF features (Table 9). The box represents the 95% confidence interval. Bias is 4.65.

ASAG Coefficients Our best linear model uses a combination of only 6 coefficients: 4 hand-crafted features (EF) and 2 natural language learned features (NLLF). Figure 4 shows, from the most relevant features, that correct answers require a balanced use of vowels⁷ (Features 1 and 5) or numbers

⁶0.48 with a Mistral-7b-instruct

⁷Words with a balanced vowel-consonant structure, like the CVCVC pattern, are easier for children to process

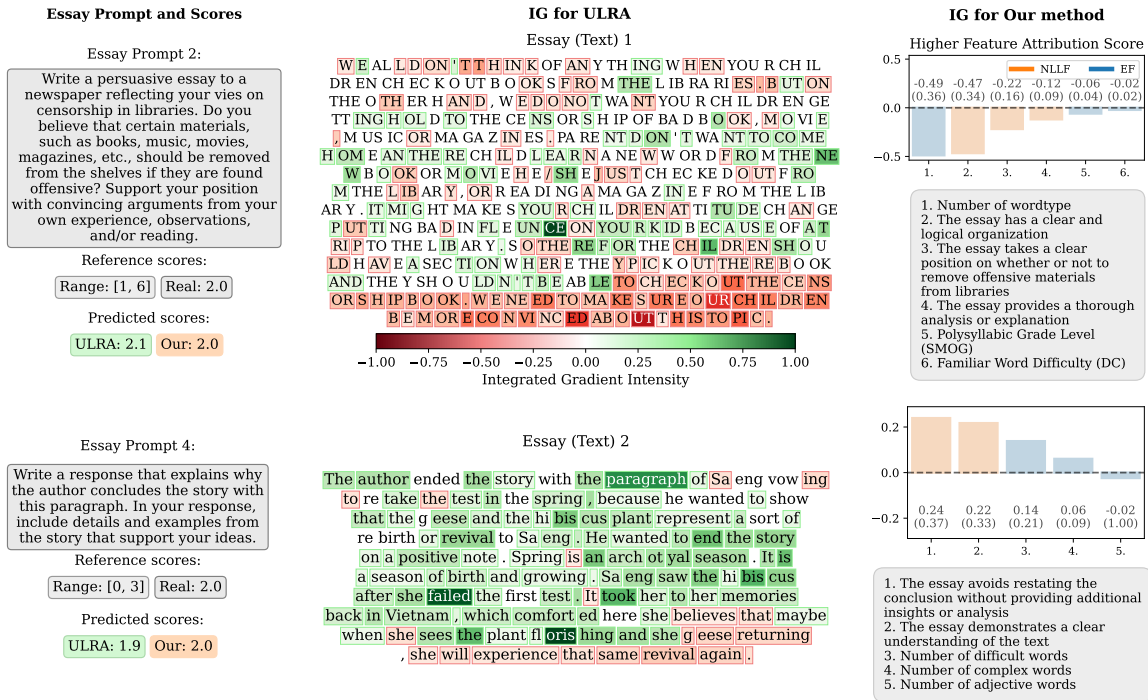


Figure 5: IG feature attribution examples from our method and ULRA on tasks 2 and 4 of the AES dataset.

(Feature 2, 3). In addition, NLLFs address common questions in which students are asked to explain if a character is making or not the right choice (Feat. 7) or just if the answer is clear (Feat. 6).

AES Coefficients Figure 3 show the coefficients of two of the eight linear regression models trained using NLLF+EF, respectively on tasks 2 and 4, persuasive and source-dependent genre, respectively. Most features have coefficients equal to zero, making the linear regression model very sparse, and leaving six usable features for each of the two models: 3 EF and 3 NLLF for the task 2, and 4 EF and 2 NLLF for the task 4. For the persuasive task, NLLFs are about argumentative techniques of the writer, whether or not it takes strongly position, and the structure of the essay.

AES IG Interpretability We claim that our system is white-box, but also interpretable. To back up our claim, we compare the two best performing models with a classical interpretation technique using Integrated Gradients (IG; Sundararajan et al. 2017) in order to attribute a score to each input feature. Figure 5 shows examples of feature attribution comparing our method and ULRA⁸. Whereas (Jiménez González and García, 1995; Brame, 1974) and help recognize proper words like a measure of coherence (Urrutia Vargas and Araya, 2023).

⁸For the linear regression, the integrated gradient is simply the product between the feature and its weight.

the attribution from the IG is complex to analyze in ULRA, our method offers two interesting advantages: (i) it is simple to interpret as it has only a few parameters which are all described in natural language, (ii) it identifies whether essays offer clear analyses or lack clear stances.

AES Human Interpretability We designed two experiments to manually validate our claim that NLLF values are coherent with humans judgments. First, we manually annotated 171 examples w.r.t the BSQ labels, in order to estimate the performances of the LLM and the NLLF Generator (NLLFG) in the subtasks. We find that both the LLM and the NLLFG obtain satisfying accuracies of .89 and .84, in concordance with the analysis of Urrutia et al. (2023). Second, for each BSQ, we selected pairs of examples based on deciles in the normalized distribution of the BSQ NLLF values. Each pair came from examples separated either by high (9 bins), medium (5 bins), or low (1 bin) distances in the distribution. We asked a human to annotate for each pair of examples, the one with highest NLLF value and the bin distance between the examples of the pair. This rendered a 6-class ordinal problem with 171 pairs. We obtained an accuracy of .44 (random is .16), an accuracy with a tolerance of 1 (Gaudette and Japkowicz, 2009) of .77 (random is .44) and a Krippendorff (2013)’s α

of 0.63 (random is 0). More details in Appendix F.

6 Conclusion and Future Work

In unsupervised ASAG of young students to diverse open ended questions in Spanish, and unsupervised AES in English, SoTA LLM-based methods are still far away from human performances. Moreover, the models trained in answer scores made with LLMs can be approximated by much simpler and interpretable models. Weak supervision on LLM labels but also on target values that are way simpler including Signal Clustering is a potential avenue of research for white-box model using several types of interpretable features such as the combination of linguistic-based expert-domain ones and compositionality-based learned ones. Future work should focus on more intensive search on the prompt space, as well as involve supervised learning (and not only weakly supervised learning) and out-of-distribution question analysis. Regarding the interpretability, the integrated gradients could be back-propagated up to the tokens in order to visualize the impact of each of them on each NLLF.

Limitations

Our work has been put in use in Spanish for a very specific type of questions that are from math exams, and in English essay with a higher quality of the text content. It would be interesting to try it in a multilingual setting, using multilingual LLMs. Future works would also imply weakly supervised multi-task learning, and more advanced prompt engineering such as the one of Lee et al. (2024), that allows for decomposing an essay into multiple traits to better score it using an LLM. Finally, it would be interesting to use manually crafted BSQs using the annotation guidelines instead of generating them, in order to see if it will improve the quality of the final model.

Ethics Statement

This work is in compliance with the ACL Ethics Policy as it allows to create models that might be more interpretable in a sensitive context such as young student education.

Acknowledgements

The authors would like to thank the Basal Funding for Centers of Excellence from ANID/PIA for their support through the Centro de Investigación

Avanzada en Educación (CIAE) with grant number FB0003.

References

- Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.
- Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. 2013. [Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading](#). *Transactions of the Association for Computational Linguistics*, 1:391–402.
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2023. [Similarity-Based Content Scoring - A more Classroom-Suitable Alternative to Instance-Based Scoring?](#) In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1892–1903.
- 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(April):106292.
- Michael K Brame. 1974. The cycle in phonology: stress in Palestinian, Maltese, and Spanish. *Linguistic Inquiry*, 5(1):39–60.
- Michael Brooks, Sumit Basu, Charles Jacobs, and Lucy Vanderwende. 2014. [Divide and correct: Using Clusters to Grade Short Answers at Scale](#). In *Learning@Scale*, pages 89–98.
- Aoife Cahill, James H. Fife, Brian Riordan, Avijit Vajpayee, and Dmytro Galochkin. 2020. [Context-based automated scoring of complex mathematical responses](#). *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 186–192.
- José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2023. Spanish pre-trained bert model and evaluation data. *arXiv preprint arXiv:2308.02976*.
- Yen Yu Chen, Chien Liang Liu, Chia Hoang Lee, and Tao Hsing Chang. 2010a. [An unsupervised automated essay-scoring system](#). *IEEE Intelligent Systems*, 25(5):61–67.
- Yen-Yu Chen, Chien-Liang Liu, Chia-Hoang Lee, Tao-Hsing Chang, et al. 2010b. [An unsupervised automated essay-scoring system](#). *IEEE Intelligent systems*, 25(5):61–67.

- Jacob Devlin, Ming-wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.
- Thomas Fel, Lucas Hervier, David Vigouroux, Antonin Poche, Justin Plakoo, Remi Cadene, Mathieu Chavidal, Julien Colin, Thibaut Boissin, Louis Bethune, Agustin Picard, Claire Nicodeme, Laurent Gardes, Gregory Flandin, and Thomas Serre. 2022. **Xplique: A Deep Learning Explainability Toolbox**. In *Workshop on Explainable Artificial Intelligence for Computer Vision (XAI4CV)*, pages 5–8.
- Lisa Gaudette and Nathalie Japkowicz. 2009. **Evaluation methods for ordinal classification**. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5549 LNAI:207–210.
- David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. 2019. XAI—Explainable artificial intelligence. *Science robotics*, 4(37):eaay7120.
- Ben Hamner, Jaison Morgan, Lynnvandev, Mark Shermis, and Tom Vander Ark. 2012. **The Hewlett Foundation: Automated Essay Scoring**. Kaggle.
- Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.
- Scott Hellman, Alejandro Andrade, and Kyle Habermehl. 2023. **Scalable and explainable automated scoring for open-ended constructed response math word problems**. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 137–147, Toronto, Canada. Association for Computational Linguistics.
- Jaeho Jeon and Seongyong Lee. 2023. **Large language models in education: A focus on the complementary relationship between human teachers and ChatGPT**. *Education and Information Technologies*, 28(12):15873–15892.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023a. **Mistral 7b**.
- Zhiwei Jiang, Tianyi Gao, Yafeng Yin, Meng Liu, Hua Yu, Zifeng Cheng, and Qing Gu. 2023b. **Improving Domain Generalization for Prompt-Aware Essay Scoring via Disentangled Representation Learning**. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 12456–12470.
- Juan E Jim  nez Gonz  lez and Carmen R Haro Garcia. 1995. Effects of word linguistic properties on phonological awareness in Spanish children. *Journal of Educational Psychology*, 87(2):193.
- Zixuan Ke and Vincent Ng. 2019. **Automated essay scoring: A survey of the state of the art**. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2019-Augus, pages 6300–6308.
- Klaus Krippendorff. 2013. **Content Analysis: An Introduction to Its Methodology**. In *Content Analysis: An Introduction to Its Methodology*.
- Yunshi Lan, Xinyuan Li, Hanyue Du, Xuesong Lu, Ming Gao, Weining Qian, and Aoying Zhou. 2024. Survey of natural language processing for education: Taxonomy, systematic review, and future trends. *arXiv preprint arXiv:2401.07518*.
- Ehsan Latif, Gyeong-Geon Lee, Knut Neuman, Tamara Kastorff, and Xiaoming Zhai. 2024. **G-SciEdBERT: A Contextualized LLM for Science Assessment Tasks in German**. pages 1–9.
- Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024. Unleashing Large Language Models’ Proficiency in Zero-shot Essay Scoring. In *Findings of ACL: EMNLP 2024*.
- Susan Lottridge, Chris Ormerod, and Amir Jafari. 2023. Psychometric considerations when using deep learning for automated scoring. *Advancing Natural Language Processing in Educational Assessment*, page 15.
- Octavio Loyola-Gonzalez. 2019. Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE access*, 7:154096–154113.
- Watheq Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. Can Large Language Models Automatically Score Proficiency of Written Essays? In *2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC-COLING 2024 - Main Conference Proceedings*, pages 2777–2786.
- Sandeep Mathias and Pushpak Bhattacharyya. 2018. ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 1169–1173.
- Panitan Muangkammuen and Fumiyo Fukumoto. 2020. **Multi-task Learning for Automated Essay Scoring with Sentiment Analysis**. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, (2015):116–123.
- Ellis B Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.
- Ellis B Page. 1967. Statistical and linguistic strategies in the computer grading of essays. In *COLING 1967 Volume 1: Conference internationale sur le traitement automatique des langues*.

- Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. [Making Reasoning Matter: Measuring and Improving Faithfulness of Chain-of-Thought Reasoning](#). In *ACL*, pages 15012–15032.
- 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. 2012. [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. *arXiv preprint arXiv:1908.10084*.
- 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.
- Martin Slawski and Matthias Hein. 2013. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization.
- Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. [Exploring LLM Prompting Strategies for Joint Essay Scoring and Feedback Generation](#). In *19th Workshop on Innovative Use of NLP for Building Educational Applications, Proceedings*.
- Kenneth Steimel and Brian Riordan. 2020. Towards Instance-Based Content Scoring with Pre-Trained Transformer Models. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, (Shermis):2015–2017.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *34th International Conference on Machine Learning, ICML 2017*, volume 7, pages 5109–5118.
- Kaveh Taghipour and Hwee Tou Ng. 2016. [A neural approach to automated essay scoring](#). In *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 1882–1891.
- Shunya Takano and Osamu Ichikawa. 2022. Automatic scoring of short answers using justification cues estimated by bert. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 8–13.
- Qiuyu Tao, Jiang Zhong, and Rongzhen Li. 2022. [AESPrompt: Self-supervised Constraints for Automated Essay Scoring with Prompt Tuning](#). In *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, pages 335–340.
- 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](#). *NeurIPS*, pages 1–14.
- Felipe Urrutia and Roberto Araya. 2023. [Who's the Best Detective? LLMs vs. MLs in Detecting Incoherent Fourth Grade Math Answers](#). *arXiv*.
- Felipe Urrutia, Cristian Calderon, and Valentin Barriere. 2023. [Deep natural language feature learning for interpretable prediction](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3736–3763, Singapore. Association for Computational Linguistics.
- Felipe Ignacio Urrutia Vargas and Roberto Araya. 2023. Automatic detection of incoherent written responses to open-ended mathematics questions of fourth graders. *MDPI Systems*, pages 0–35.
- Cong Wang, Zhiwei Jiang, Yafeng Yin, Zifeng Cheng, Shiping Ge, and Qing Gu. 2023. [Aggregating Multiple Heuristic Signals as Supervision for Unsupervised Automated Essay Scoring](#). *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1:13999–14013.
- 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](#). *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, (Bea):626–667.
- Rebecka Weegar and Peter Idestam-almquist. 2024. [Reducing Workload in Short Answer Grading Using Machine Learning](#). *International Journal of Artificial Intelligence in Education*, 34(2):247–273.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [HuggingFace's Transformers: State-of-the-art Natural Language Processing](#).
- Jiayi Xie, Kaiwei Cai, Li Kong, Junsheng Zhou, and Weiguang Qu. 2022. Automated Essay Scoring via Pairwise Contrastive Regression. In *Proceedings - International Conference on Computational Linguistics, COLING*, volume 29, pages 2724–2733.
- 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 Findings of ACL: EMNLP 2020*, pages 1560–1569.
- Mengxue Zhang, Sami Baral, Neil Heffernan, and Andrew Lan. 2022. [Automatic Short Math Answer Grading via In-context Meta-learning](#). *Proceedings of the 15th International Conference on Educational Data Mining, EDM 2022*.

A ASAG Dataset Statistics

We present a summary of the dataset in Table 4, including the total number of students, different questions and student answers. We added the average number of answers per question for each year.

Year	#Students	#Questions	#Answers	Avg. #Ans. per Question
2022	3,232	1,204	62,297	≈ 52
2017	231	44	1,315	≈ 30
Total	3,463	-	63,612	-

Table 4: Summary of students, total questions, total answers, and average answers per question across years.

B Features

B.1 EF

We manually designed linguistic features, detailed in Table 6, aimed at capturing structural, morphological, and statistical properties of student responses for ASAG task in Chilean Spanish. Given the unique characteristics of children’s writing in a mathematical context, we categorize EF into six groups: *morphological features*, which analyze the presence of numbers, digit counts, and the ratio of numerical to non-numerical tokens, essential for evaluating arithmetic-based responses; *syntactic features*, focusing on tokenization, negation length, and the distribution of non-numeric tokens, which help assess the sentence structure typical of early learners; *lexical features*, which measure character frequencies to detect common patterns in children’s spelling and word usage in Chilean Spanish; *structural features*, capturing answer length, repeated character sequences, and vowel/consonant distributions, which are indicative of fluency and coherence; *punctuation features*, which count and analyze punctuation marks, distinguishing between mathematical symbols (e.g., decimal points, equation signs) and non-mathematical punctuation that might indicate explanatory attempts; and *phonological features*, assessing vowel proportions relative to alphabetic characters to identify phonetic simplifications or spelling mistakes common in young learners.

For example, the phonological feature measuring the proportion of alphabetic characters that are vowels (Feature 1) distinguishes between responses like A1 (0.33) and A2 (0.52) to the same question, with A2 being more phonetically fluent (see Table 5). Similarly, syntactic complexity can be estimated through the number of tokens without

digits (Feature 2), where a detailed explanation (12 tokens, A2) correlates with a higher score than a brief response (2 tokens, A1). Morphological traits such as the binary presence of a number (Feature 3) allow us to capture relevant numerical grounding in an answer; for instance, A2 includes a number and scores higher. Phonological depth is further captured by vowel density (Feature 4), where answers with higher vowel proportion (0.31) exhibit better coherence than sparse ones (0.25). Finally, structural complexity, such as the maximum number of consecutive non-vowel characters in a token (Feature 5), helps detect unnatural or noisy tokens, e.g. A1 has a high value (5) due to “Hkflg”, suggesting incoherence, compared to A2’s more natural phrasing (value of 2).

B.2 NLLF

Following the method outlined in Urrutia et al., we utilize a selected roughly 12% subset of the train-set to generate the NLLF. We ask to a Mixtral to generate a diverse pool of Binary Subtask Questions (BSQ) for our ASAG/AES task. A member of our research team manually removes irrelevant BSQ. We chose 12 binary questions through automatic selection via Agglomerative Clustering, taking the centroid. We automatically answer the selected binary questions on the portion of the train-set with the same LLM to teach a Spanish/English BERT model in answering to all the selected binary questions. We generated a total of 24 features from the sigmoid of the logits of the trained BERT to provide *Yes* or *No* answers to the 12 binary questions (Table 6), i.e. two features per binary question.

C LLM

We used a simple prompt containing the definition of the task. For the AES task, initially, we use an unspecified prompt to score answers, yet observed a tendency for the model to assign notably low scores to answers containing kid misspelling errors. Subsequently, we refined our prompt specifying “not penalize for spelling mistakes and focus on the intended meaning conveyed by the student’s answer”. This adjustment yielded enhancements in the performance of the LLM.

D ULRA as Weak Signal

In the AES dataset, the LLM performances are outperformed by the ones obtained using the ULRA method, which is unsupervised but also black-box.

Feature	Question (Q) and Answer (A1-A2)	Feature Value	Score
Proportion of alphabetic characters that are vowels	Q: Si Jose multiplica 150 veces 1 ¿Cuál sería su resultado? Explica	-	-
	A1: 150x1 es 51 <i>(Low vowel ratio)</i>	0.33	3.0
	A2: sería 150 porque 150 veces 1 sería 150 <i>(Higher vowel ratio)</i>	0.52	7.0
Number of tokens without numbers	Q: José compró 4 cajas de leche a \$245 cada una. El vendedor le cobró en total \$950. ¿Está correcto lo que le cobró el vendedor? Explica.	-	-
	A1: está bien <i>(short, lacks analysis)</i>	2	2.0
	A2: está mal la respuesta es 980 se multiplica 245 x4 y el resultado es 980 <i>(detailed reasoning)</i>	12	7.0
Exist a number in the answer	Q: Paulina tiene 16 lápices para repartir entre 4 amigas. Su mamá le dice a Paulina que le va a dar 5 lápices a cada amiga. ¿Es correcto lo que le dice su mamá?	-	-
	A1: no es mal porque no <i>(no number)</i>	0	3.0
	A2: la mamá está mal porque son 4 lápices para cada amiga <i>(includes number)</i>	1	7.0
Proportion of vowels in the answer	Q: Una manzana pesa 0,35 g, otra 0,251 g y la última 0,51 g. ¿Cuánto pesan entre las tres?	-	-
	A1: 150x1 es 51 <i>(low vowel ratio)</i>	0.25	4.0
	A2: sumo todas las manzanas es 1,111. y sumé 0,35 más 0,251 más 0,51 y me dio ese resultado <i>(more fluent)</i>	0.31	7.0
Max. consecutive non-vowel characters	Q: ¿Cuál es el resultado de 501x2? Comenta cómo resolviste el ejercicio y explica qué es la multiplicación.	-	-
	A1: Hkflg <i>(noisy token)</i>	5	1.0
	A2: es 1 002, lo resolví con sumas y la multiplicación es una suma repetida <i>(coherent)</i>	2	7.0

Table 5: Examples of five expert features with their feature values for question/answer pairs in the ASAG task in Chilean Spanish (examples presented in their original language, Spanish).

For these reasons, we propose an additional experiments where we train a logistic regression model on our interpretable vector of expert and natural language learned representations, using the scores from ULRA as a weak label. The results are shown in Table 8. We can see that the use of a more accurate weak signal does not allow to improve the global performances.

E Prompt used for Zero-shot ICL with LLM

Figure 6 is the prompt used for ASAG dataset. The model is guided to assign grades while disregarding spelling errors and focusing on the content of the student’s response. Figure 7 is the prompt used for Task 1 and Task 3-8 of the AES dataset, where the range of possible scores is task-specific and highlighted in blue, where `[[MIN]]` is the minimum score and `[[MAX]]` the maximum. Figure 8 presents the prompt for Task 2 of the AES dataset, which is the only task differing from the other eight tasks in its focus on grading according to *Writing Applications*. Throughout all figures, the orange text represents the model’s initial response.

Instruction and Model answer

You are a teacher who assigns grades (between 1 and 7) to fourth grade students’ answers to open-ended math questions. Do not discount for spelling or grammatical errors. Focus on what the student is trying to say with his or her answer. Indicate the grade after the prefix ‘Score:’

Got it. Give me a question between <Q></Q> and an answer between <A>. Then ask me ‘What is the score for the answer to the question?’. I will then give you the note after the prefix ‘Score:’

Follow-up instruction

Question: <Q> `[[Q]]` </Q> Answer: <A> `[[A]]` What is the score for the answer to the question?

Figure 6: Prompt used for Zero-shot ICL with LLM on the ASAG dataset. Translated from Spanish. the orange text represents a model’s initial response.

Feature Name	Type of Feature
Exist a number in the answer	EF (Morphological)
Number of digits in the answer	EF (Morphological)
Number of numerical values in the answer	EF (Morphological)
The answer is composed of digits	EF (Morphological)
The answer is NaN (Not a Number)	EF (Morphological)
Proportion of digit characters in the answer	EF (Morphological)
Number of tokens in the answer	EF (Syntactic)
Number of tokens that do not contain numbers	EF (Syntactic)
Ratio of non-numeric tokens to the total number of tokens	EF (Syntactic)
Ratio of punctuation marks to the total number of tokens	EF (Syntactic)
Ratio of vowels to the total number of tokens	EF (Syntactic)
Length of the negation of the answer	EF (Syntactic)
Frequency of character 'x' in the answer	EF (Lexical)
Frequency of character 'y' in the answer	EF (Lexical)
Frequency of character 'g' in the answer	EF (Lexical)
Frequency of character 'h' in the answer	EF (Lexical)
Frequency of character 'j' in the answer	EF (Lexical)
Frequency of character 'k' in the answer	EF (Lexical)
Frequency of character 'w' in the answer	EF (Lexical)
Frequency of character 'ñ' in the answer	EF (Lexical)
Number of characters in the answer	EF (Structural)
Length of the longest number in the answer	EF (Structural)
Length of the longest sequence of repeated characters	EF (Structural)
Maximum number of consecutive vowels in a token	EF (Structural)
Maximum number of consecutive non-vowel characters in a token	EF (Structural)
Number of punctuation marks in the answer	EF (Punctuation)
Number of mathematical punctuation marks in the answer	EF (Punctuation)
Proportion of punctuation characters in the answer	EF (Punctuation)
Proportion of non-mathematical punctuation characters	EF (Punctuation)
Proportion of punctuation and digit characters in the answer	EF (Punctuation)
Proportion of non-digit and non-mathematical punctuation characters	EF (Punctuation)
Proportion of alphabetic characters that are vowels	EF (Phonological)
Proportion of vowels in the answer	EF (Phonological)
<hr/>	
The answer shows a correct calculation of a quantity	NLLF
The answer does not show a correct calculation of a quantity	NLLF
The answer explains why a character is wrong	NLLF
The answer does not explain why a character is wrong	NLLF
The answer is free of conceptual errors	NLLF
The answer contains conceptual errors	NLLF
The answer shows a correct understanding of the question	NLLF
The answer does not show a correct understanding of the question	NLLF
The answer correctly indicates a quantity	NLLF
The answer does not correctly indicate a quantity	NLLF
The answer is written in a way that can be easily understood	NLLF
The answer is not written in a way that can be easily understood	NLLF
The answer is written clearly and concisely	NLLF
The answer is not written clearly and concisely	NLLF
The answer is correctly written in numerical format	NLLF
The answer is not correctly written in numerical format	NLLF
The answer is accompanied by an explanation	NLLF
The answer is not accompanied by an explanation	NLLF
The answer is complete and does not lack any relevant information	NLLF
The answer is incomplete or lacks relevant information	NLLF
The answer addresses the question	NLLF
The answer does not address the question	NLLF
The answer correctly identifies the value	NLLF
The answer does not correctly identify the value	NLLF

Table 6: Expert features (EF) and Natural Language Learned Features (NLLF) for the ASAG task Everything was translated from Spanish.

Feature Name	Code
Long-Word Ratio	RIX
Polysyllabic Grade Level	SMOG
Complex Word Grade Level	GF
Familiar Word Difficulty	DC
Number of sentences	S
Number of adjective words	JJ
Number of unique words	UW
Number of preposition / subordinating - conjunction words	IN
Number of long words	LW
Number of determiner words	DT
Number of difficult words	DW
Number of complex words	CW
Number of noun words	NN
Number of commas	CO
Number of wordtype	WT
Number of non-basic words	NBW
Number of words	W
Number of characters	CH
Number of adverb words	RB

Table 7: Linguistic Features from Wang et al. (2023) for the AES task.

Method	SF	Text	EF	NLLF	EF + NLLF
ULRA	-	.6423	-	-	-
LR	✗	-	.5712	.6041	.6227
	✓	-	.5707	.6035	.6193

Table 8: Results of the Logistic Regression model using the scores of ULRA as a target during the weakly supervised learning. SF is Signal Filtering.

Instruction and Model answer

You are a teacher who assigns grades (between [[MIN]] and [[MAX]]) to essays from students ranging in grade levels from Grade 7 to Grade 10. You will help me break down the 'assign grade to student essay' task. To do this, I will give you a sample essay along with the assignment. Indicates the score after the prefix 'Score:'.

Got it. Give me a question between <A> and an essay between <E></E>. Then ask me 'What is the score for the essay?'. I will then give you the score after the prefix 'Score:'.

Follow-up instruction

Assignment: <A> [[A]] Essay: <E> [[E]] </E> What is the score for the essay?

Instruction and Model answer

You are a teacher who assigns grades (between 1 and 6) to essays from students ranging in grade levels from Grade 7 to Grade 10. You will help me break down the 'assign grade to student essay according to Writing Applications' task. To do this, I will give you a sample essay along with the assignment. Indicates the score after the prefix 'Score:'.

Got it. Give me a question between <A> and an essay between <E></E>. Then ask me 'According to Writing Applications, what is the score for the essay?'. I will then give you the score after the prefix 'Score:'.

Follow-up instruction

Assignment: <A> [[A]] Essay: <E> [[E]] </E> According to Writing Applications, what is the score for the essay?

Figure 8: Prompt used for Zero-shot ICL with LLM on the Task 2 of the AES dataset. The orange text represents a model's initial response.

Figure 7: Prompt used for Zero-shot ICL with LLM on the Task 1 and Tasks 3 to 8 of the AES dataset. The blue text highlights the range of values specific to each task, while the orange text represents a model's initial response.

Feature Name	Coef.	Std. err.	[0.025	0.975]
<i>Intercept</i>	4.65	0.01	4.64	4.67
The answer is correctly written in numerical format	0.00	0.00	0.00	0.00
The answer is written in a way that can be easily understood	0.50	0.01	0.47	0.52
The answer shows a correct calculation of a quantity	0.00	0.00	0.00	0.00
The answer correctly identifies the value	0.00	0.00	0.00	0.00
The answer shows a correct understanding of the question	0.00	0.00	0.00	0.00
The answer explains why a character is wrong	0.12	0.01	0.10	0.13
The answer is accompanied by an explanation	0.00	0.01	0.00	0.03
Exist a number in the answer	0.27	0.01	0.24	0.30
Frequency of character 'g' in the answer	0.00	0.00	0.00	0.00
Frequency of character 'h' in the answer	0.00	0.00	0.00	0.00
Frequency of character 'k' in the answer	0.00	0.00	0.00	0.00
Frequency of character 'w' in the answer	0.00	0.00	0.00	0.00
Frequency of character 'x' in the answer	0.03	0.01	0.02	0.04
Frequency of character 'y' in the answer	0.11	0.02	0.07	0.14
Number of characters of the answer	0.14	0.12	0.00	0.35
Number of tokens in answer	0.00	0.00	0.00	0.00
Length of the negation of the answer	0.00	0.06	0.00	0.21
Length of the longest number in the answer	0.11	0.01	0.08	0.13
Maximum number of consecutive non- vowel characters in a token	0.15	0.01	0.12	0.17
Number of digits in the answer	0.00	0.00	0.00	0.00
Number of mathematical punctuation marks in the answer	0.00	0.01	0.00	0.02
Number of tokens that do not contain numbers	0.35	0.18	0.00	0.59
Number of numerical values in the answer	0.00	0.00	0.00	0.00
Number of tokens in the answer	0.00	0.00	0.00	0.00
Proportion of alphabetic characters that are vowels	0.43	0.04	0.35	0.52
Proportion of punctuation characters in the answer	0.00	0.00	0.00	0.00
Proportion of punctuation and non- vowel characters in the answer	0.14	0.01	0.10	0.16
Proportion of vowels in the answer	0.21	0.05	0.10	0.30
Ratio of non-numeric tokens to the total number of tokens	0.00	0.02	0.00	0.05
Ratio of punctuation marks to the total number of tokens	0.00	0.00	0.00	0.00

Table 9: Coefficients of the Linear Regression with Signal Filtering using EF+NLLF features in the ASAG dataset. [0.025, 0.975] refers to the 95% confidence interval of the coefficient.

F Human validation of the NLLFs

F.1 NLLFG Classifiers

Here we analyze how accurate were the NLLF generated by the BERT-like model, and also the weak labels by the LLM. We took 190 examples from the validation set used to train the NLLFG of the ASAP task, and asked an expert to manually label them regarding the labels of a BSQ. More precisely, we manually annotated 10 examples sampled uniformly per BSQ having non-zero weights in the linear regressions (approximately 2-3 BSQs per essay set) across 8 essay sets. We compare the labeling of the expert with the outputs of the NLLFG and LLM models, using classical classification metrics such as precision, recall and F1-score.

The results for both the models are available in Table 10. The LLM obtained a better F1-score than the smaller transformer model, which was expected. It is interesting to note that the accuracy of the NLLFG model is 0.78, close to the ones of the LLM (0.86). The macro F1-scores are more divergent as the LLM reaches 0.84 and the NLLFG 0.74, which is still better than random.

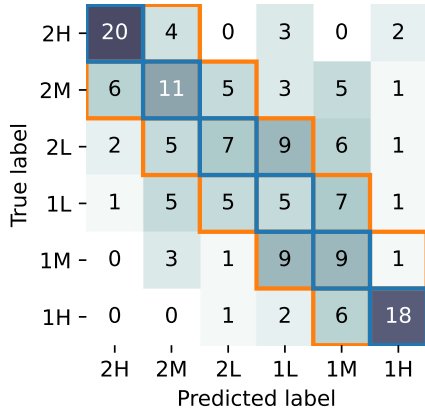
Model	Label	Prec.	Rec.	F1	Acc.
Mixtral	Yes	91	88	89	86
	No	76	82	79	
NLLFG	Yes	90	78	84	78
	No	57	76	65	

Table 10: Performance of NLLFG and Mixtral on a manually annotated set of 190 examples. The dataset consists of 10 uniformly selected examples per BSQ (approximately 2-3 BSQs per essay set) across 8 essay sets.

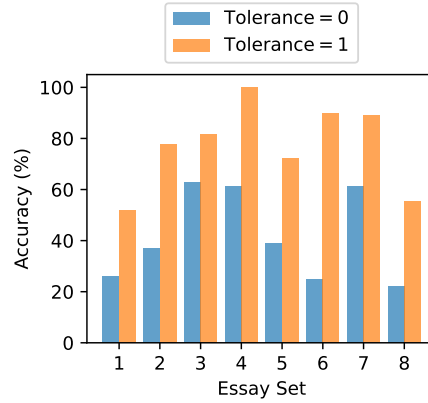
F.2 NLLFs Before the Linear Regression

We designed another experiment to assess the reliability of the NLLF with respect to human annotation, showing pairs of examples to a human, and asking which should have the highest value in NLLF and what is the distance in values between the examples of the pair. As the NLLF are normalized before the linear regression, hence each score depends on the whole group and becomes relative to the other examples (the best has a highly positive score and the worst has a highly negative score).

Pairs of examples with various distances in-



(a) Global Confusion Matrix



(b) Accuracies with tolerance 0 and 1

Figure 9: Metrics between the human annotation and the real values of the NLLFs, for the AES task.

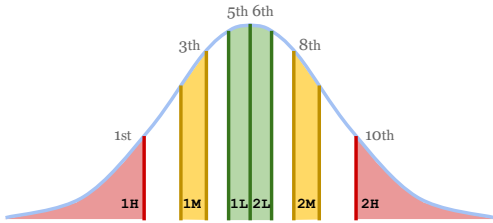


Figure 10: Examples were picked from bins with **high**, **medium** and **low** distance between each others. For a pair of examples, the annotator has to find which example has the highest value, and what is the distance between the examples.

between the examples were randomly selected regarding their places in the distributions: pairs from the first and last deciles of the distribution, pairs from the 3rd and 8th, and pairs from the 5th and 6th. We ask a human to tell for each pair, which example is the highest in the distribution, and how large is the distance between them. It gave us a classification problem with 6 ordinal classes: First-High (1H), First-Medium (1M), First-Low (1L), Second-Low (2L), Second-Medium (2M), Second-High (2H).

We focused on the 19 BSQs having non-zero weights in the linear regressions, and randomly selected 3 examples of High, Medium and Low distances between the pairs, which gave us a total of 171 pairs to annotate coming from 6 classes. Figure 10 shows the bins of the examples from the different categories.

The results overall are shown in Figure 9, with the confusion matrix and the We report a Krippendorff (2013)’s alpha of 0.63, an Accuracy of 0.43

Method	Weak Signal	Signal Filt.	Text	EF	NLLF	EF + NLLF
Length	None	-	0.0015	-	-	-
Jaccard Sim.	None	-	-0.1335	-	-	-
Jaccard Sim.	None	-	0.3170	-	-	-
ULRA	LF	-	0.4562	-	-	-
	EF+LF	-	0.3902	-	-	-
Z-score	None	-	0.4346	-	-	-
LLM	None	-	0.5629	-	-	-
LLM-CoT	None	-	0.4631	-	-	-
Linear Regression	Z-score	✗	-	0.4472	0.3627	0.4167
	LLM-based signal	✗	-	0.4212	0.2984	0.3772
	Z-score	✓	-	0.4471	0.3435	0.4915
	LLM-based signal	✓	-	0.3682	0.2925	0.4115
BERT	Z-score	✗	0.3965	-	-	-
	LLM-based signal	✗	0.3867	-	-	-
	Z-score	✓	0.2451	-	-	-
	LLM-based signal	✓	0.3848	-	-	-
Human	None	-	0.7403	-	-	-

Table 11: Results on ASAG using the QWK

(random is 0.17) and an accuracy with a tolerance of 1 (Gaudette and Japkowicz, 2009) of 0.77 (random is 0.44). This shows that human rank the examples in an order similar to the ones of the NLLF values 77% of the time using a tolerance of 1 in the ordinal classification.

G Others

Table 11 shows the results on the ASAG dataset using QWK. The results are very similar: LLM is better than our method, which is itself better than ULRA.

A Survey on Automated Distractor Evaluation in Multiple-Choice Tasks

Luca Benedetto, Shiva Taslimipoor, Paula Buttery

ALTA Institute, Dept. Computer Science and Technology, University of Cambridge
name.surname@cl.cam.ac.uk

Abstract

Multiple-Choice Tasks are one of the most common types of assessment item, due to their feature of being easy to automatically and objectively grade. A key component of Multiple-Choice Tasks are distractors – i.e., the wrong answer options – since *poor* distractors affect the overall quality of the item: e.g., if they are obviously wrong, they are never selected. Thus, previous research has focused extensively on techniques for automatically generating distractors, which can be especially helpful in settings where large pools of questions are desirable or needed. However, there is no agreement within the community about the techniques that are most suited to evaluate generated distractors, and the ones used in the literature are sometimes not aligned with how distractors perform in real exams. In this review paper, we perform a comprehensive study of the approaches which are used in the literature for evaluating generated distractors, propose a taxonomy to categorise them, discuss if and how they are aligned with distractors performance in exam settings, and what are the differences for different question types and educational domains.

1 Introduction

Multiple-Choice Tasks are a very popular form of students' assessment, due to their standardised format: they are easy to (automatically) grade and they remove subjectivity from the scoring process, and can thus be used to quickly and efficiently assess large numbers of students, in both high-stakes and low-stakes settings. A challenging step of curating high-quality Multiple-Choice Tasks – also referred to as Multiple-Choice Questions (MCQs) – is the generation of distractors, i.e., the incorrect options. Indeed, high-quality distractors must satisfy several properties (see §2.3), such as being incorrect but plausible, and consistent with the context but objectively wrong. The generation of high quality distractors has been shown to be challenging

even for human experts (Shin et al., 2019), and to target this issue and generate large quantities of distractors (which are needed for large pools of questions) recent research has explored many approaches to automatically generate distractors, as discussed in two recent surveys (Awalurahman and Budi, 2024; Alhazmi et al., 2024). According to the assessment and testing literature (Nunnally and Bernstein, 1994), the most reliable approach to evaluate distractors is pretesting: new MCQs are shown to students in exam settings, and their response patterns are used to assess the distractors. Unfortunately, pretesting is unfeasible when automatically generating large numbers of distractors and undesirable in some settings, e.g., due to exam security concerns (Ha et al., 2019); thus, automatically generated distractors are most commonly evaluated with static approaches or with manual evaluation. However, the best techniques to automatically evaluate generated distractors are not commonly agreed across the community and the ones used in practice are rarely aligned with the performance of distractors in real exam settings. Hence, in this paper, i) we perform a comprehensive review of the approaches used in the literature for automated distractor evaluation, ii) we propose a new taxonomy to categorise them, iii) we discuss which ones are the most aligned with pedagogical theory and with the performance of distractors in real exam settings (also focusing on different educational domain and question types), and iv) provide some guidelines for future research.

2 Related Work

2.1 Distractor Generation

Two very recent surveys provide a good overview of approaches to distractor generation and the trends in the literature (Awalurahman and Budi, 2024; Alhazmi et al., 2024). Similarly to many other domains, distractor generation has seen a

rapid shift in recent years: the majority of approaches are now based on (large) language models, in contrast with research pre-transformers which was primarily based on traditional machine learning. We refer to the two survey papers mentioned above for a detailed description of the different techniques used in distractor generation.

2.2 Distractor Evaluation

The task of distractor evaluation is much less studied than distractor generation, even though it is becoming increasingly relevant: indeed, with modern generative models it is very easy to experiment with different prompts and generate a large set of distractors, and it is thus crucial to have ways to automatically and reliably evaluate them. Unfortunately, neither of the survey papers mentioned above focused sufficiently on the techniques and metrics which are used to automatically evaluate distractors. Considering fully automated metrics, [Alhazmi et al. \(2024\)](#) only mention ranking-based (Precision, Recall, F1-score, Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), and Mean Average Precision (MAP)) and n-gram metrics (BLUE, ROUGE, and METEOR), while [Awalurahman and Budi \(2024\)](#) only mentions BLEU, ROUGE and METEOR. While these are all metrics that are indeed used in the literature, this list leaves out many others, which are very relevant and potentially more aligned with the performance of distractors in exam settings.

Few papers have distractor evaluation as main focus, proposing automated approaches for the task. [Pho et al. \(2015\)](#) work on distractors that are Named Entities in a knowledge graph, and propose an approach to evaluate them based on the syntactic and semantic relation between the distractors and the correct answer, and their relatedness in the graph. [Ghanem and Fyshe \(2023\)](#) generate “bad distractors” and train a model to estimate whether a given distractor is good or bad. Finally, [Raina et al. \(2023\)](#) propose an ensemble of three metrics which are meant to measure the incorrectness, plausibility, and diversity of distractors.

2.3 About Good Distractors

The educational literature is rich in recommendations and guidelines on how to create good distractors for MCQs. Ideally, these guidelines should be implemented within the models for automated distractor generation and evaluation, but our literature review suggests that in many cases the approaches

used for evaluating automatically generated distractors in the NLP and AI for Education communities are somewhat disconnected from them. It is important to note that there are differences between educational domains – e.g., guidelines for language learning and mathematics cannot be exactly the same – but there are many common aspects. Distractors that are too easy fail to assess students’ true understanding, while those that are too difficult or misleading can cause confusion and frustration; thus, distractors should be plausible, but objectively unacceptable ([Yeung et al., 2019](#)). Potentially, distractors should try to capture the common errors and misconceptions of students ([Lee et al., 2016](#); [Scarlatos et al., 2024](#)), which enables targeted interventions. Also, distractors should be independent from one another, otherwise one or more could be excluded with logical reasoning, thus hindering the quality of the question. Distractors should be semantically and grammatically coherent with the context ([Ghanem and Fyshe, 2023](#); [Gao et al., 2019](#)), and similar in length, style, and grammatical form to the correct answer ([Pho et al., 2015](#)). In language pedagogy literature, there is the recommendation that the target word and the distractors belong to the same word class ([Heaton, 1988](#)), ideally being “false synonyms” ([Goodrich, 1977](#)).

3 Taxonomy

Figure 1 presents the taxonomy we propose to categorise approaches from the previous literature. We group the different approaches based on the type of information that they use for evaluation. **Dynamic** approaches are based on learners’ answers, and **static** approaches leverage only the textual information from the distractors (and potentially the correct answer, the question, and the reading passage). Dynamic approaches (§4), and specifically *Traditional Distractor Analysis*, can be seen as the *gold standard*, since they are based on students’ responses and are an actual measurement of how distractors perform in exam settings; they can be further divided into approaches based on real students and the ones based on *responses from Question Answering (QA) models*. On the other hand, static approaches (§5) can be seen as an alternative to dynamic ones, as they can be used when it is unfeasible to obtain students’ responses. Static approaches can be further divided into three groups: i) *comparative* approaches evaluate generated distractors by comparing them to some refer-

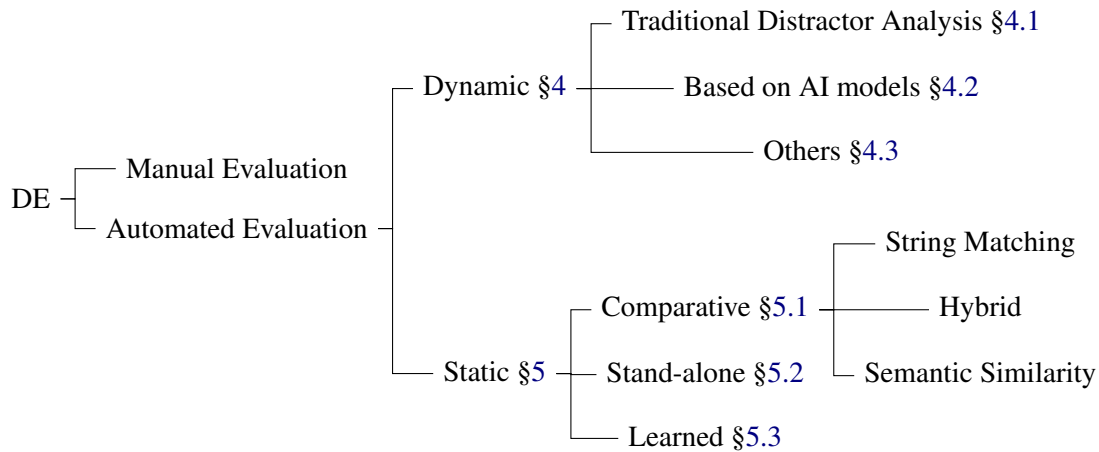


Figure 1: The taxonomy we propose to categorize the different approaches for Distractor Evaluation (DE).

ence ones, which are considered as gold standard, ii) *stand-alone* approaches consist in computing some measures of similarity between distractors and between distractors and the correct answer, and iii) *learned* approaches are machine learning models trained to predict the quality of generated distractors. From a practical point of view, there are notable similarities between distractor generation evaluation and difficulty estimation. In difficulty estimation, the gold standard is difficulty from pretesting – e.g., from Item Response Theory (Hambleton and Swaminathan, 2013) – but approaches have been proposed for difficulty estimation from text for when students’ responses are not available (Benedetto et al., 2022).

Previous approaches are described in Sections 4 and 5, and Table 1 provides an overview of all the papers we discuss in this survey, grouped according to the proposed taxonomy. The table also shows the educational domain which each paper worked on, whether manual evaluation is used in addition to automated evaluation metrics, and whether distractors are evaluated individually or as a set.

4 Dynamic Approaches

Dynamic approaches to distractor analysis use students’ responses to measure how well distractors perform in exams. They can be further divided into **traditional distractor analysis** §4.1 and **AI-based dynamic approaches** §4.2, depending on whether human or virtual students’ responses are used.

Traditional distractor analysis is most commonly used in the Education and Assessment literature: it studies how distractors perform in real exams, observing the response patterns of human students, and can thus be considered the *optimal* approach

to distractor evaluation. When it is unfeasible to use traditional distractor analysis due to cost, time constraints, or concerns about safety, AI-based dynamic approaches can be used. These are based on same techniques, but use the responses of QA models as a proxy for the responses from real students. Similar to difficulty estimation tasks, which are ideally performed via pretesting with real learners, research explored the possibility of using machine learning and AI to simulate it (Benedetto et al., 2022; AlKhuzayy et al., 2021). This includes the setting of virtual pretesting, which became more popular in recent years (Park et al., 2024; Uto et al., 2024; Benedetto et al., 2024).

Previous research also experimented with some approaches based on the responses of human learners but different from the ones used in traditional distractor analysis; they will be discussed in §4.3.

4.1 Traditional Distractor Analysis

Traditional distractor analysis is based on studying how often distractors are selected, and which is the (average) skill level of the learners selecting different distractors. Again, these metrics are based on how distractors perform in real exam settings, thus can be considered as the *optimal* ones.

Distractors that are never (or rarely) selected by students are *poor* distractors (Nunnally and Bernstein, 1994); the rule of thumb mentioned in several papers is that each distractor should be selected by at least 5% of the students (Haladyna and Downing, 1993), with the exception of very easy MCQs, which are correctly answered by more than 90% of the students (Gierl et al., 2017). Only three articles evaluate automatically generated distractors using the frequency with which participants select each

Paper	Manual Eval.	Single	Set	Dynamic §4		Static Comparative §5.1			Static Stand-Alone §5.2	Static Learned §5.3
				Traditional	QA Models	Lexical	Hybrid	Semantic		
(Pho et al., 2015)		X							X	
(Lee et al., 2016)	X	X		X						
(Gao et al., 2019)	X	X			X					
(Zhou et al., 2019)	X	X			X					
(Chung et al., 2020)	X	X	X							
(Qiu et al., 2020)	X	X			X					
(Maurya and Desarkar, 2020)	X	X					X			
(Offerijns et al., 2020)	X	X	X							
(Kalpakchi and Boye, 2021)	X	X	X					X		
(Rodríguez-Torrealba et al., 2022)		X							X	
(Xie et al., 2022)	X	X					X			
(Chiang et al., 2022)		X								
(Panda et al., 2022)	X	X								
(Qu et al., 2023)	X	X	X					X		
(Wang et al., 2023)	X	X							X	
(Raina et al., 2023)		X	X							X
(Yoshimi et al., 2023)		X	X							
(Login, 2024)		X					X			
(Zhou and Li, 2024)	X	X					X			
(Qu et al., 2024)		X	X						X	
(Taslimipour et al., 2024)	X	X	X						X	
(Lin et al., 2024)		X					X			
(Wang et al., 2025)	X	X	X						X	
(McNichols et al., 2023)		X								X
(McNichols et al., 2024)		X								X
(Feng et al., 2024)	X	X								X
(Scarlatos et al., 2024)	X	X								
(Fernandez et al., 2024)		X								
(Aldabe and Maritxalar, 2010)		X	X							
(Liang et al., 2018)		X					X			
(Zhang and VanLehn, 2021)		X	X							
(Dutulescu et al., 2024)	X	X					X			
(Mirkov and Ha, 2003)		X	X							
(Lee et al., 2025)	X	X	X							X
(Ren and Zhu, 2021)	X	X								
(Bitew et al., 2022)	X	X						X		
(Ghanem and Fyshe, 2023)	X	X								
(De-Fitero-Dominguez et al., 2024)	X	X							X	
(Yu et al., 2024)	X	X								
(Luo et al., 2024)	X	X	X							

Table 1: List of the papers discussed in this survey, grouped according to the proposed taxonomy.

of them: (Aldabe and Maritxalar, 2010; Zhang and VanLehn, 2021; Lee et al., 2016).

Another indication of distractor quality from the Education literature is the difference between the number of students selecting each distractor and the number of students selecting the correct answer: if a distractor is chosen more often than the correct answer, this is probably an indication of poor instructions or a misleading question (Nunnally and Bernstein, 1994). We did not find any paper evaluating generated distractors with this metric.

Lastly, since a good distractor is one that is selected by students who perform poorly and ignored by those who perform well (Gronlund, 1968), distractors that are selected by students that are (on average) of higher skill level than the students selecting the correct choice are poor distractors. We found only two papers using this factor to evaluate automatically generated distractors: Mitkov and Ha (2003) and Lee et al. (2025) divide students into a group of highly skilled students and a group of beginners, and label distractors that are selected by more students in the upper group than by students in the lower group as poor distractors.

4.2 AI-based Dynamic Approaches

Fundamentally, these use measurements similar to the ones from traditional distractor analysis, but based on the responses from QA models rather than human learners. Using machine learning models as a proxy of students, they should be validated accordingly. This is rarely done in the literature.

Chung et al. (2020) make the assumption that poor distractors will reduce the difficulty of the MCQ task for a QA model, thus use accuracy as an indicator of distractor quality, by comparing distractors generated with different models: the higher the accuracy, the worse the quality of the distractors. Similarly, Offerijns et al. (2020) study how the accuracy of a QA model changes when using manually-curated distractors rather than automatically generated ones: they observe that results are similar, thus claim that the generated distractors are on-par with the human-curated ones.

Guo et al. (2024) use the generated distractors to augment a dataset, which is then used to train a QA model. The quality of generated distractors is evaluated by measuring the QA accuracy on a separate test set: a better performance on the test set would indicate that the generated distractors were effective for training the model, and thus they are good distractors.

4.3 Others

Some papers use human responses for distractor evaluation, but in a setting different from traditional distractor analysis. Kalpakchi and Boye (2021) recruit participants on a crowd-sourcing platform and ask them to answer reading comprehension MCQs without providing them with reading passages. The authors claim that this approach can evaluate the *plausibility* of distractors by measuring how often they are selected. Luo et al. (2024) compare the response accuracy of three students on questions with distractors generated with different models, and claim that lower accuracy in responding to a question would indicate that there were better distractors. Yoshimi et al. (2023) evaluate distractors by measuring how the response accuracy of human annotators changes when using the original compared to generated distractors, aiming to make the accuracy as close as possible in the two settings. This is similar to the approach by Offerijns et al. (2020) but using humans rather than QA models.

5 Static Approaches

Static approaches evaluate distractors using only the content of the items, without considering learners' responses. Importantly, most of these approaches are not aligned *per se* with how distractors would perform in real exam settings, thus they should be validated (but often are not, in previous literature). They can be divided into *Comparative*, *Stand-alone*, and *Learned* approaches.

5.1 Comparative

Comparative approaches are based on a comparison between generated distractors and the reference ones available in the test dataset: this assumes that these reference distractors are of good quality and are the *only* distractors of good quality for a question. In other words, any generated distractor which is different from the reference ones is massively penalised. Both assumptions are somewhat problematic for distractor evaluation: experimental datasets often do not contain high-quality pretested questions (particularly the publicly available ones), and it might happen that other distractors are as effective, if not better, than the ones in the datasets. This disadvantage comes from the fact that most comparative approaches were not originally thought of for distractor evaluation, but rather for Machine Translation, and thus have fundamental issues when it comes to distractor eval-

uation (Rodriguez-Torrealba et al. (2022); Taslimipoor et al. (2024), inter alia). However, even with these major shortcomings, they are by far most commonly used approaches to evaluate new distractor generation models, due to their popularity and ease of implementation.

5.1.1 String Matching

String matching is the single most frequently used approach for distractor evaluation in the literature. Most papers used BLEU (Papineni et al., 2002) and/or ROUGE (Lin, 2004) to compare the generated distractors with reference ones in the experimental datasets (see Table 2 for the list of all papers). Other common metrics are Precision, Recall, F1-score, MRR, and NDCG (the list of papers is shown in Table 3). Notably, this distinction is also due to the fact that papers in the two tables mostly work on different types of questions: papers in 2 mainly work with reading comprehension questions with longer text answers, while papers in 3 mainly work with either cloze items or science tests with single word or named entity answers.

Paper	BLEU	ROUGE
(Gao et al., 2019)	X	X
(Zhou et al., 2019)	X	X
(Chung et al., 2020)	X	X
(Qiu et al., 2020)	X	X
(Maurya and Desarkar, 2020)	X	X
(Offerijns et al., 2020)	X	X
(Rodriguez-Torrealba et al., 2022)	X	X
(Xie et al., 2022)	X	X
(Qu et al., 2023)	X	X
(Login, 2024)	X	X
(Zhou and Li, 2024)	X	X
(Qu et al., 2024)	X	X
(De-Fitero-Dominguez et al., 2024)	X	X
(Luo et al., 2024)		X
(Lin et al., 2024)	X	X
(Taslimipoor et al., 2024)	X	
(Wang et al., 2025)	X	X

Table 2: List of papers using BLEU and/or ROUGE.

Other papers evaluated generated distractors using metrics based on string matching, but different from the metrics mentioned above. Liang et al. (2018) and Bitew et al. (2022) use Mean Average Precision, Luo et al. (2024) use Accuracy, and Kalpakchi and Boye (2021) measures the fraction of MCQs for which at least one generated distractor matches one of the reference ones.

McNichols et al. (2023); Feng et al. (2024); Fernandez et al. (2024), and McNichols et al. (2024) (all working on maths questions) define and use three *alignment-based* metrics: i) *partial match*

Paper	Precision	Recall	F1	MRR	NDCG
(Liang et al., 2018)	X	X		X	X
(Kalpakchi and Boye, 2021)		X			
(Ren and Zhu, 2021)	X		X	X	X
(Bitew et al., 2022)	X	X		X	
(Chiang et al., 2022)	X		X	X	X
(Panda et al., 2022)	X	X			
(Wang et al., 2023)	X	X	X		
(Yoshimi et al., 2023)			X		
(Dutulescu et al., 2024)	X		X	X	X
(Yu et al., 2024)	X	X	X	X	X

Table 3: List of papers using Precision, Recall, F1 score, Mean Reciprocal Rank, or NDCG for evaluation.

evaluates whether at least one of the generated distractors matches one of the reference ones, ii) *exact match* evaluates whether all the generated distractors match the reference ones, and iii) *proportional match* measures the proportion of generated distractors which match the reference ones. In addition to these three metrics, Scarlatos et al. (2024) define *weighted proportional*, which is a reinterpretation of the proportional match: it re-weights each “match” in the proportional metric giving more importance to reference distractors which are most commonly selected by students. Notably, considering all the evaluation metrics based on string matching, this *weighted proportional* is the only one which explicitly takes into consideration how well distractors perform in real exams.

5.1.2 Semantic Similarity

Several articles evaluate generated distractors by measuring their semantic similarity to the reference ones, using diverse techniques for capturing the semantic meaning of distractors and their distance from the reference ones. While this is arguably more reliable than string matching, it still relies entirely on the quality of distractors in the experimental dataset. The most common approach is BERTScore (Zhang et al., 2019), which is used by Login (2024); Qu et al. (2024, 2023) to compute the similarity between generated distractors and the reference ones. Other embedding techniques are used in other articles: Ren and Zhu (2021) use Word2Vec (Mikolov et al., 2013), Maurya and Desarkar (2020) use BERT (Devlin et al., 2019) embeddings, and more recently Taslimipoor et al. (2024) apply Sentence-BERT (Reimers and Gurevych, 2019) to compute similarity. Notably, no one of these papers give weights to how different reference distractors perform in real exams.

5.1.3 Hybrid lexical-semantic

As a middle-ground between the purely lexical string matching approach described in §5.1 and the semantic embeddings from §5.1.2, some papers used METEOR (Banerjee and Lavie, 2005) for evaluating the similarity between generated and reference distractors. Specifically, it was used by Login (2024); Zhou and Li (2024); Maurya and Desarkar (2020); Xie et al. (2022); Lin et al. (2024). This has the same limitations as the approaches described above, as it relies entirely on the quality of the reference distractors, and implies that those are the only good distractors for a given question.

5.2 Stand-alone Approaches

Stand-alone approaches are all the evaluation techniques which are based on textual information only and do not rely on reference distractors. As such, they are meant to detect high-quality distractors even when these do not match some reference ones, and are not susceptible to low-quality distractors in the reference data. Most of these evaluation metrics are meant to capture the *plausibility* and *diversity* requirements of good distractors.

5.2.1 Estimating plausibility

Pho et al. (2015) focus on the relatedness between the distractors and the correct answer option, primarily working on questions whose responses are named entities. The semantic similarity is then measured looking at the distance between the named entities of each distractor and the correct answer option in a taxonomy of named entities.

Plausibility is modelled as the cosine similarity between each generated distractor and the correct answer option in (Rodriguez-Torrealba et al., 2022; De-Fitero-Dominguez et al., 2024). The authors state that higher similarity to the correct answer option means better distractors and use such approach for evaluating the distractors. Still, they do not study the correlation between the results obtained with their evaluation metric and an evaluation based on students’ responses, thus this metric might reward distractors which are too close to the correct answer, and thus low quality.

A different take on plausibility is taken by Raina et al. (2023): they define *plausibility* as the sum of the confidence scores of a multiclass QA model for each of the distractors. This approach assumes that the confidence of a MCQA model is a good proxy of the confidence of real students, and evaluates this assumption by using a dataset which provides sta-

tistical information about how often distractors in the dataset are selected by real students (Mullooly et al., 2023); this is one of few works validating the metrics used for distractor evaluation.

5.2.2 Estimating diversity

More papers focused on studying the diversity of generated distractors, using Pairwise-BLEU, Distinct (Li et al., 2016), or other techniques. Pairwise-BLEU is used by Qu et al. (2023) and Wang et al. (2025), while Distinct is used by Qu et al. (2024) and Qu et al. (2023). Two different approaches are used by Raina et al. (2023), who use the BERT Equivalence Metric (BEM) (Bulian et al., 2022), and Taslimipoor et al. (2024), who use Sentence-BERT to measure the semantic similarity between different generated distractors. In all these papers the authors claim that high diversity is desirable, hence similarity between distractors should be low.

5.2.3 Others

Kalpakchi and Boye (2021) propose a set of evaluation metrics, including several stand-alone approaches different from all the approaches used by other papers. Most of them are filters which could actually be implemented within a DG model itself, and include measures such as i) the fraction of MCQs with two or more generated distractors which are equal, ii) the fraction of MCQs for which generated distractor match the correct answer, and others (we refer to the paper for the full list).

5.3 Learned Approaches

Learned evaluation metrics are machine learning models – with different architectures – specifically trained to evaluate the quality of generated distractors. Several approaches have been used in the literature, and they try to capture different characteristics that good distractors are expected to have. Notably, these approaches are on average the most recent of all the papers surveyed.

The first learned metric to evaluate generated distractors was proposed by Ghanem and Fyshe (2023), which is one of the few papers exclusively focusing on the evaluation of generated distractors. The proposed approach consists in automatically generating *bad distractors*, and training a model to estimate whether a distractor is good or bad (i.e. binary classification); the metric is validated with manual evaluation. A similar approach is used by Raina et al. (2023) and Qu et al. (2024). In the first paper, a model is trained to distinguish between

the correct answer option and the distractors, in a binary classification setting; the probability that such trained model assigns to each distractor (more specifically, $1 - P$) indicates *how incorrect* each distractor is.¹ In the latter, an Alberta model is trained to predict whether a given distractor is a correct answer to the corresponding question, and return a classification score in the range $[0, 100]$; the authors refer to this as *faithful score*.

Three papers focused on learned approaches to estimate the plausibility of generated distractors. In two of them (McNichols et al., 2023; Feng et al., 2024) the authors, who define plausibility as the likelihood of a distractor being selected by real students, compute it by training a BERT-based machine learning model on real students’ responses to predict the fraction of students selecting each distractor. The trained model assigns a probability score to each distractor, and these scores are then combined in two ways: i) by summing the selection probability of all distractors, and ii) by computing the entropy among them (to make sure that all are selected with reasonable frequency by students). In the third (Lee et al., 2025), the authors train a pairwise ranker to select, given a pair of distractors, the more plausible. Ground truth plausibility is estimated from students’ responses, thus this metric is aligned distractor performance in exam settings.

Finally, in one paper which performs distractor generation via reinforcement learning from preference feedback (Wang et al., 2025), the authors leverage the same reward model that was used in training during the reinforcement learning phase to then evaluate the generated distractors.

6 Discussion

6.1 Alignment with exam performance

Considering all the evaluation approaches described above, the only ones which are by definition aligned with how distractors perform in real exam settings are the techniques from traditional distractor analysis (§4.1), since they evaluate distractors based on the responses of real students. We argue that these approaches should be used whenever possible. Unfortunately, in most cases, that is not feasible, and some alternative approaches have to be used. In all these cases, it is important to validate the evaluation approach to ensure that they align with the exam performance of distractors, but this

¹The metric is validated using student response data from a publicly available dataset (Mullooly et al., 2023).

is rarely done in the literature. The main reason for this is that most of the publicly available datasets – e.g., RACE (Lai et al., 2017), SQuAD (Rajpurkar et al., 2016), or the MCQ dataset by Ren and Zhu (2021) – do not provide such information, thus it is impossible to properly validate the evaluation metrics on them and all evaluations are built upon weak foundations. One notable exception is the Cambridge MCQ Reading Dataset (Mullooly et al., 2023), which contains an indication of how often distractors are selected by students in real exam settings: the dataset contains both *good* and *bad* distractors, and can thus be used to validate different evaluation metrics. Similarly, private datasets, such as the Eedi dataset used by Scarlatos et al. (2024) and others, likely contain statistics about students’ responses, and thus provide the information needed to validate the evaluation metrics (as it is done for the weighted proportional metric described in §5.1.1). However, they are inaccessible for the wider research community.

6.2 Evaluating individual distractors and distractor sets

The taxonomy proposed in §3 categorises evaluation metrics based on the information used for evaluating generated distractors. However, another relevant dimension to consider is whether evaluation metrics work on individual distractors or distractors as a set of options. Indeed, distractors should ideally be evaluated with both, since they capture different aspects in relation to designing a good question item. The number of papers that evaluate distractors individually is an overwhelming majority in the literature, and only few use metrics that consider distractors as a set, as shown in Table 1.

All the comparative approaches in §5.1 focus on evaluating individual distractors. While this is very relevant, as it can help detect distractors which are too close to or too far from the correct answer option, it is a suboptimal evaluation. Indeed, in real exam settings distractors are shown to students in a set of (usually) four items (one being correct), and distractor evaluation metrics should also consider the similarity and differences between the distractors – thus evaluating *sets* of distractors. Notably, even considering the papers which perform a manual evaluation of the distractors, these are evaluated individually (e.g., annotators are asked to classify each of them as *acceptable* or *not acceptable* (although out of the main scope of this survey paper, we include an analysis of manual evaluation in the

appendix §A). From our analysis, a total of 15 papers (out of the 40 doing automated evaluation) use automated metrics that evaluate distractors as a set rather than as individual items.

6.3 Educational domains and question types

In the context of distractor generation and evaluation for MCQs, question types and educational domains play a crucial role in designing effective evaluation metrics. These factors influence the characteristics of distractors and the criteria used to assess their quality. The subject or educational domain influences the complexity, language, and knowledge required for distractor evaluation. For instance, in science and mathematics, evaluation metrics should check for scientific validity or in language learning, like in reading comprehension questions, evaluation should assess linguistic similarity and conceptual relevance. These aspects of evaluation have not been investigated explicitly in the literature, however we can see that for example almost all papers experimenting with the RACE dataset for reading comprehension, evaluate distractors using metrics from machine translation (see Table 2) while most distractor generations in the domain of science (Liang et al., 2018; Ren and Zhu, 2021; Bitew et al., 2022; Dutulescu et al., 2024) or with Cloze-style questions (where answers and distractors are single words or named entities) (Chiang et al., 2022; Panda et al., 2022; Wang et al., 2023; Yoshimi et al., 2023; Yu et al., 2024) are mainly evaluated using ranking based statistical measures (see Table 3).

6.4 About manual evaluation

Although not discussed in this survey, since our focus is on automated metrics which could be used in an automated generation and evaluation pipeline, manual evaluation is still used by the majority of papers (see Appendix A), sometimes in addition to the automated metrics and in other cases as the single evaluation approach. Annotators are domain experts, or the authors themselves, or recruited from crowd-sourcing platform – thus leading to annotations of varying reliability.

7 Conclusions

In this survey paper we have performed a comprehensive study of the metrics and techniques which are used to automatically evaluate generated distractors in the context of Multiple-Choice Tasks, and have proposed a taxonomy to categorise them.

We have seen that there is not a commonly agreed metric in the literature, and different authors and research groups tend to use different evaluation techniques. Most importantly, the metrics which are most commonly used in the literature (e.g., BLEU and ROUGE) are sub-optimal and arguably not aligned with how distractors actually perform in exams: indeed, they evaluate newly generated distractors by comparing them with some reference ones assuming that the references are i) of high quality and ii) the only distractors of high quality that can be created for the given question. Both assumptions are very strong, and not really supported by previous research, especially for publicly available datasets such as RACE (which is one of the most commonly used datasets).

Ideally, distractors should be evaluated with Traditional Distractor Analysis (i.e., with real learners) but, when this is not possible, the evaluation metrics used in its place should aim at being more aligned with how distractors perform in real exam settings and with the requirements that good distractors are expected to satisfy (according to vast literature from Education and Assessment), such as being consistent and coherent with the question and the correct option, and being plausible enough to *distract* learners. This highlights the need for validating the evaluation metrics which are used in distractor generation and evaluation settings and developing new, more aligned, ones. The development of such metrics should also take into consideration the differences between different educational domains, as the requirement might be different depending on the specific application scenario.

Limitations

When collecting the papers to review, we have performed several searches and used snow-balling to collect all the relevant publications which we could find. However, there is always a possibility that we might have missed some relevant research works. Also, we have highlighted the limitations of the current approaches to distractor evaluation, and this survey paper serves as motivation to focus more on the evaluation of distractors but, at this stage, we do not have an alternative approach to propose that might target these issues (yet).

Acknowledgments

This paper reports on research supported by Cambridge University Press & Assessment.

References

- Itziar Aldabe and Montse Maritxalar. 2010. [Automatic Distractor Generation for Domain Specific Texts](#). In *Advances in Natural Language Processing*, Lecture Notes in Computer Science, pages 27–38, Berlin, Heidelberg. Springer.
- Elaf Alhazmi, Quan Z. Sheng, Wei Emma Zhang, Munazza Zaib, and Ahoud Alhazmi. 2024. [Distractor Generation in Multiple-Choice Tasks: A Survey of Methods, Datasets, and Evaluation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14437–14458, Miami, Florida, USA. Association for Computational Linguistics.
- Samah AlKhuzaey, Floriana Grasso, Terry R. Payne, and Valentina Tamma. 2021. [A Systematic Review of Data-Driven Approaches to Item Difficulty Prediction](#). In Ido Roll, Danielle McNamara, Sergey Sosnovsky, Rose Luckin, and Vania Dimitrova, editors, *Artificial Intelligence in Education*, volume 12748, pages 29–41. Springer International Publishing, Cham.
- Jun Araki, Dheeraj Rajagopal, Sreecharan Sankaranarayanan, Susan Holm, Yukari Yamakawa, and Teruko Mitamura. 2016. [Generating Questions and Multiple-Choice Answers using Semantic Analysis of Texts](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1125–1136, Osaka, Japan. The COLING 2016 Organizing Committee.
- Halim Wildan Awalurahman and Indra Budi. 2024. [Automatic distractor generation in multiple-choice questions: A systematic literature review](#). *PeerJ Computer Science*, 10:e2441.
- Satanjeev Banerjee and Alon Lavie. 2005. [Meteor: An automatic metric for mt evaluation with improved correlation with human judgments](#). In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72. Association for Computational Linguistics.
- 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.
- Luca Benedetto, Paolo Cremonesi, Andrew Caines, Paula Buttery, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2022. [A survey on recent approaches to question difficulty estimation from text](#). *ACM Computing Surveys*, page 3556538.
- Semere Kiros Bitew, Amir Hadifar, Lucas Sterckx, Johannes Deleu, Chris Develder, and Thomas De-meester. 2022. [Learning to reuse distractors to support multiple-choice question generation in education](#). *IEEE Transactions on Learning Technologies*, 17:375–390.
- Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. 2022. [Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305. Association for Computational Linguistics.
- Darryl J Chamberlain and Russell Jeter. 2020. [Creating diagnostic assessments: Automated distractor generation with integrity](#). *Journal of Assessment in Higher Education*, 1(1):30–49.
- Shang-Hsuan Chiang, Ssu-Cheng Wang, and Yao-Chung Fan. 2022. [CDGP: Automatic Cloze Distractor Generation based on Pre-trained Language Model](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5835–5840, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ho-Lam Chung, Ying-Hong Chan, and Yao-Chung Fan. 2020. [A BERT-based Distractor Generation Scheme with Multi-tasking and Negative Answer Training Strategies](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4390–4400, Online. Association for Computational Linguistics.
- David De-Fitero-Dominguez, Eva Garcia-Lopez, Antonio Garcia-Cabot, Jesus-Angel Del-Hoyo-Gabaldon, and Antonio Moreno-Cediel. 2024. [Distractor Generation through Text-to-Text Transformer Models](#). *IEEE Access*, pages 1–1.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186. Association for Computational Linguistics.
- Andreea Dutulescu, Stefan Ruseti, Denis Iorga, Mihai Dascalu, and Danielle S. McNamara. 2024. [Beyond the Obvious Multi-choice Options: Introducing a Toolkit for Distractor Generation Enhanced with NLI Filtering](#). In *Artificial Intelligence in Education*, pages 242–250, Cham. Springer Nature Switzerland.
- Wanyong Feng, Jaewook Lee, Hunter McNichols, Alexander Scarlatos, Digory Smith, Simon Woodhead, Nancy Ornelas, and Andrew Lan. 2024. [Exploring Automated Distractor Generation for Math Multiple-choice Questions via Large Language Models](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3067–3082, Mexico City, Mexico. Association for Computational Linguistics.

- Nigel Fernandez, Alexander Scarlatos, Simon Woodhead, and Andrew Lan. 2024. [DiVERT: Distractor Generation with Variational Errors Represented as Text for Math Multiple-choice Questions](#). *Preprint*, arXiv:2406.19356.
- Yifan Gao, Lidong Bing, Piji Li, Irwin King, and Michael R. Lyu. 2019. [Generating distractors for reading comprehension questions from real examinations](#). 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, pages 6423–6430, Honolulu, Hawaii, USA. AAAI Press.
- Bilal Ghanem and Alona Fyshe. 2023. [DISTO: Evaluating Textual Distractors for Multi-Choice Questions using Negative Sampling based Approach](#). *Preprint*, arXiv:2304.04881.
- 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.
- Hugo Gonalo Oliveira, Igor Caetano, Renato Matos, and Hugo Amaro. 2023. [Generating and ranking distractors for multiple-choice questions in portuguese](#). In *12th Symposium on Languages, Applications and Technologies (SLATE 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- Hubbard C Goodrich. 1977. [Distractor efficiency in foreign language testing](#). *Tesol Quarterly*, pages 69–78.
- Norman Edward Gronlund. 1968. *Constructing Achievement Tests*.
- Qi Guo, Chinmay Kulkarni, Aniket Kittur, Jeffrey P Bigham, and Emma Brunskill. 2016. [Questimator: Generating Knowledge Assessments for Arbitrary Topics](#). page 3726–3732.
- Yingshuang Guo, Jianfei Zhang, Junjie Dong, Chen Li, Yuanxin Ouyang, and Wenge Rong. 2024. [Optimization Strategies for Knowledge Graph Based Distractor Generation](#). In *Knowledge Science, Engineering and Management*, pages 189–200, Singapore. Springer Nature.
- Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. [Predicting the Difficulty of Multiple Choice Questions in a High-stakes Medical Exam](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.
- Thomas M Haladyna and Steven M Downing. 1993. [How many options is enough for a multiple-choice test item?](#) *Educational and psychological measurement*, 53(4):999–1010.
- Ronald K Hambleton and Hariharan Swaminathan. 2013. *Item response theory: Principles and applications*. Springer Science & Business Media.
- John Brian Heaton. 1988. *Writing English language tests*. Longman.
- Shu Jiang and John SY Lee. 2017. [Distractor generation for Chinese fill-in-the-blank items](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148. Association for Computational Linguistics.
- Dmytro Kalpakchi and Johan Boye. 2021. [BERT-based distractor generation for Swedish reading comprehension questions using a small-scale dataset](#). In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 387–403, Aberdeen, Scotland, UK. Association for Computational Linguistics.
- Dmytro Kalpakchi and Johan Boye. 2024. [Generation and Evaluation of Multiple-choice Reading Comprehension Questions for Swedish](#). *Northern European Journal of Language Technology*, 10(1).
- Girish Kumar, Rafael E Banchs, and Luis Fernando D’Haro. 2015. [Revup: Automatic gap-fill question generation from educational texts](#). In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 154–161.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. [RACE: Large-scale ReAding Comprehension Dataset From Examinations](#). *Preprint*, arXiv:1704.04683.
- John Lee, Donald Sturgeon, and Mengqi Luo. 2016. [A CALL System for Learning Preposition Usage](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 984–993, Berlin, Germany. Association for Computational Linguistics.
- Yooseop Lee, Suin Kim, and Yohan Jo. 2025. [Generating Plausible Distractors for Multiple-Choice Questions via Student Choice Prediction](#). *Preprint*, arXiv:2501.13125.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119. Association for Computational Linguistics.
- Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C Lee Giles. 2018. [Distractor Generation for Multiple Choice Questions Using Learning to Rank](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

- Chen Liang, Xiao Yang, Drew Wham, Bart Pursel, Rebecca Passonneau, and C. Lee Giles. 2017. [Distractor Generation with Generative Adversarial Nets for Automatically Creating Fill-in-the-blank Questions](#). In *Proceedings of the 9th Knowledge Capture Conference, K-CAP '17*, pages 1–4, New York, NY, USA. Association for Computing Machinery.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text summarization branches out*, pages 74–81. Association for Computational Linguistics.
- Runfeng Lin, Dacheng Xu, Huijiang Wang, Zebiao Chen, Yating Wang, and Shouqiang Liu. 2024. [DGRC: An Effective Fine-Tuning Framework for Distractor Generation in Chinese Multi-Choice Reading Comprehension](#). In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 815–820.
- Nikita Login. 2024. [Wrong Answers Only: Distractor Generation for Russian Reading Comprehension Questions Using a Translated Dataset](#). *Journal of Language and Education*, 10(4):56–70.
- 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.
- Kaushal Kumar Maurya and Maunendra Sankar Desarkar. 2020. [Learning to Distract: A Hierarchical Multi-Decoder Network for Automated Generation of Long Distractors for Multiple-Choice Questions for Reading Comprehension](#). In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20*, pages 1115–1124, New York, NY, USA. Association for Computing Machinery.
- Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2023. [Exploring Automated Distractor and Feedback Generation for Math Multiple-choice Questions via In-context Learning](#). *Preprint*, arXiv:2308.03234.
- Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. [Automated Distractor and Feedback Generation for Math Multiple-choice Questions via In-context Learning](#). *Preprint*, arXiv:2308.03234.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. [Distributed Representations of Words and Phrases and their Compositionality](#). In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Ruslan Mitkov and Le An Ha. 2003. [Computer-Aided Generation of Multiple-Choice Tests](#). In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17–22.
- Andrew Mullooly, Øistein Andersen, Luca Benedetto, Paula Buttery, Andrew Caines, Mark J. F. Gales, Yasin Karatay, Kate Knill, Adian Liusie, Vatsal Raina, and Shiva Taslimipoor. 2023. [The Cambridge Multiple-Choice Questions Reading Dataset](#). Technical report, Cambridge University Press and Assessment.
- J.C. Nunnally and I.H. Bernstein. 1994. *Psychometric Theory*.
- Jeroen Offerijns, Suzan Verberne, and Tessa Verhoef. 2020. [Better Distractions: Transformer-based Distractor Generation and Multiple Choice Question Filtering](#). *Preprint*, arXiv:2010.09598.
- 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, Dublin, Ireland. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.
- Jae-Woo Park, Seong-Jin Park, Hyun-Sik Won, and Kang-Min Kim. 2024. [Large Language Models are Students at Various Levels: Zero-shot Question Difficulty Estimation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8157–8177, Miami, Florida, USA. Association for Computational Linguistics.
- Van-Minh Pho, Anne-Laure Ligozat, and Brigitte Grau. 2015. [Distractor Quality Evaluation in Multiple Choice Questions](#). In Cristina Conati, Neil Heffernan, Antonija Mitrovic, and M. Felisa Verdejo, editors, *Artificial Intelligence in Education*, volume 9112, pages 377–386. Springer International Publishing, Cham.
- Zhaopeng Qiu, Xian Wu, and Wei Fan. 2020. [Automatic Distractor Generation for Multiple Choice Questions in Standard Tests](#). *arXiv:2011.13100 [cs]*.
- Fanyi Qu, Hao Sun, and Yunfang Wu. 2024. [Unsupervised Distractor Generation via Large Language Model Distilling and Counterfactual Contrastive Decoding](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 827–838, Bangkok, Thailand. Association for Computational Linguistics.

- Fanyi Qu, Che Wang, and Yunfang Wu. 2023. [Accurate, Diverse and Multiple Distractor Generation with Mixture of Experts](#). In *Natural Language Processing and Chinese Computing*, Lecture Notes in Computer Science, pages 761–773, Cham. Springer Nature Switzerland.
- Vatsal Raina, Adian Liusie, and M.J.F. Gales. 2023. [Assessing Distractors in Multiple-Choice Tests](#).
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ Questions for Machine Comprehension of Text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- 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.
- Siyu Ren and Kenny Q. Zhu. 2021. [Knowledge-Driven Distractor Generation for Cloze-Style Multiple Choice Questions](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4339–4347.
- Ricardo Rodriguez-Torrealba, Eva Garcia-Lopez, and Antonio Garcia-Cabot. 2022. [End-to-End generation of Multiple-Choice questions using Text-to-Text transfer Transformer models](#). *Expert Systems with Applications*, 208:118258.
- 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](#). *Preprint*, arXiv:2405.05144.
- Jinnie Shin, Qi Guo, and Mark J. Gierl. 2019. [Multiple-Choice Item Distractor Development Using Topic Modeling Approaches](#). *Frontiers in Psychology*, 10:825.
- Katherine Stasaski and Marti A Hearst. 2017. [Multiple choice question generation utilizing an ontology](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312. Association for Computational Linguistics.
- Shiva Taslimipoor, Luca Benedetto, Mariano Felice, and Paula Buttery. 2024. [Distractor Generation Using Generative and Discriminative Capabilities of Transformer-based Models](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5052–5063, Torino, Italia. ELRA and ICCL.
- Masaki Uto, Yuto Tomikawa, and Ayaka Suzuki. 2024. [Question Difficulty Prediction Based on Virtual Test-Takers and Item Response Theory](#). In *Proceedings of the EvalLAC'24: Workshop on Automatic Evaluation of Learning and Assessment Content*.
- Hui-Juan Wang, Kai-Yu Hsieh, Han-Cheng Yu, Jui-Ching Tsou, Yu An Shih, Chen-Hua Huang, and Yao-Chung Fan. 2023. [Distractor Generation based on Text2Text Language Models with Pseudo Kullback-Leibler Divergence Regulation](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12477–12491, Toronto, Canada. Association for Computational Linguistics.
- Ruofan Wang, Yuru Jiang, Yuyang Tao, Mengyuan Li, Xia Wang, and Shili Ge. 2025. [High-Quality Distractors Generation for Human Exam Based on Reinforcement Learning from Preference Feedback](#). In *Natural Language Processing and Chinese Computing*, pages 94–106, Singapore. Springer Nature.
- Jiayuan Xie, Ningxin Peng, Yi Cai, Tao Wang, and Qingbao Huang. 2022. [Diverse Distractor Generation for Constructing High-Quality Multiple Choice Questions](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:280–291.
- Chak Yan Yeung, John SY Lee, and Benjamin K Tsou. 2019. [Difficulty-aware distractor generation for gap-fill items](#). In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 159–164. Australasian Language Technology Association.
- 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, Toronto, Canada. Association for Computational Linguistics.
- Han Cheng Yu, Yu An Shih, Kin Man Law, Kai Yu Hsieh, Yu Chen Cheng, Hsin Chih Ho, Zih An Lin, Wen-Chuan Hsu, and Yao-Chung Fan. 2024. [Enhancing Distractor Generation for Multiple-Choice Questions with Retrieval Augmented Pretraining and Knowledge Graph Integration](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11019–11029, Bangkok, Thailand. Association for Computational Linguistics.
- Lishan Zhang and Kurt VanLehn. 2021. [Evaluation of auto-generated distractors in multiple choice questions from a semantic network](#). *Interactive Learning Environments*, 29(6):1019–1036.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. [BERTScore: Evaluating text generation with bert](#). *arXiv preprint arXiv:1904.09675*.
- Hao Zhou and Li Li. 2024. [Qadg: Generating question-answer-distractors pairs for real examination](#). *Neural Computing and Applications*.
- Xiaorui Zhou, Senlin Luo, and Yunfang Wu. 2019. [Co-Attention Hierarchical Network: Generating Coherent Long Distractors for Reading Comprehension](#). *arXiv:1911.08648 [cs]*.

A On Manual Evaluation

Even though manual evaluation is not scalable to large amounts of distractors and cannot be used in a fully-automated content generation pipeline, it is still the most commonly used approach to evaluate distractors in distractor generation papers. From our analysis, a total of 23 papers out of 40 use manual evaluation in addition to automated evaluation; in addition to these, we also find 9 papers where the manual annotation is the only evaluation that is performed.

There are not commonly agreed guidelines on how to evaluate the distractors manually, and different papers follow different approaches and provide different labels, in some cases limiting the annotation to *good* and *bad* distractors, and in some other cases ranking on a Likert scale (e.g., from 1 to 5) some aspects of the distractors. In general, we observe that the annotators are either asked to provide an overall evaluation of the distractors (i.e., whether they are *good* distractors), or evaluate them according to the following aspects: plausibility (also referred to as distracting ability), fluency, coherence with the text (also referred to as validity), diversity (between the generated distractors), and being related to students' misconceptions. Notably, only two papers explicitly ask annotators to evaluate the diversity of the generated distractors – thus evaluating them as a set – and most of the papers perform an evaluation of individual distractors. Table 4 provides an overview of which of these aspects are considered in the different papers which perform manual evaluation of distractors.

Paper	Overall	Plausibility	Fluency	Misconception	Coherence	Diversity	Only Manual Eval
(Kumar et al., 2015)	X						X
(Guo et al., 2016)	X						X
(Lee et al., 2016)	X	X					
(Araki et al., 2016)		X			X		X
(Jiang and Lee, 2017)		X					X
(Liang et al., 2017)	X						X
(Stascki and Hearst, 2017)	X						X
(Zhou et al., 2019)		X	X		X		
(Gao et al., 2019)		X					
(Offerijns et al., 2020)	X				X		
(Chamberlain and Jeter, 2020)				X			X
(Qiu et al., 2020)		X	X		X		
(Maurya and Desarkar, 2020)		X			X		
(Ren and Zhu, 2021)		X			X		
(Kalpakchi and Boye, 2021)	X						
(Xie et al., 2022)			X		X	X	
(Bitew et al., 2022)	X						
(Panda et al., 2022)	X						
(Ghanem and Fyshe, 2023)	X						
(Gonçalo Oliveira et al., 2023)	X						X
(Wang et al., 2023)		X			X		
(Qu et al., 2023)		X	X		X	X	
(Zhou and Li, 2024)		X	X		X		
(Feng et al., 2024)		X			X		
(Yu et al., 2024)	X	X			X		
(Dutulescu et al., 2024)	X						
(Scarlatos et al., 2024)	X	X					
(Kalpakchi and Boye, 2024)	X						X
(Luo et al., 2024)		X			X		
(Taslimipoor et al., 2024)	X						
(Wang et al., 2025)	X	X			X		
(Lee et al., 2025)			X		X		

Table 4: List of papers using **manual evaluation**, with an indication of which characteristics of distractors the annotators are asked to evaluate.

Alignment Drift in CEFR-prompted LLMs for Interactive Spanish Tutoring

Mina Almasi and Ross Deans Kristensen-McLachlan
Department of Linguistics, Cognitive Science, and Semiotics
Aarhus University, Denmark
mina@cc.au.dk, rdkm@cc.au.dk

Abstract

This paper investigates the potentials of Large Language Models (LLMs) as adaptive tutors in the context of second-language learning. In particular, we evaluate whether system prompting can reliably constrain LLMs to generate only text appropriate to the student's competence level. We simulate full teacher-student dialogues in Spanish using instruction-tuned, open-source LLMs ranging in size from 7B to 12B parameters. Dialogues are generated by having an LLM alternate between tutor and student roles with separate chat histories. The output from the tutor model is then used to evaluate the effectiveness of CEFR-based prompting to control text difficulty across three proficiency levels (A1, B1, C1). Our findings suggest that while system prompting can be used to constrain model outputs, prompting alone is too brittle for sustained, long-term interactional contexts - a phenomenon we term **alignment drift**. Our results provide insights into the feasibility of LLMs for personalized, proficiency-aligned adaptive tutors and provide a scalable method for low-cost evaluation of model performance without human participants.

1 Introduction

The popularization of large language models (LLMs), particularly through the emergence of user-friendly interfaces such as ChatGPT, has led many stakeholders across society to consider how to use such technology effectively and safely to facilitate access to knowledge and education (Yan et al., 2024). Language education has not been immune to this hype, and with seemingly good cause, since LLMs show potential across a range of areas where they might enhance language learning.

One such area is their inherent *interactivity*. Interactive feedback is widely regarded as an important factor in second-language (L2) learning (Loewen and Sato, 2018). For L2 learners far removed from their target language community, op-

portunities for such interaction can be rare. With LLMs, though, learners appear to now have the opportunity to engage with a "speaker" of the target language freely and at their own pace (Kohnke et al., 2023). Other potential benefits include personalized teaching (Klimova et al., 2024) and reduced L2 anxiety (Hayashi and Sato, 2024).

These ideas build on decades of research on intelligent tutoring systems and computer-assisted learning (Psootka et al., 1992; Slavuj et al., 2015). In contrast to earlier rule-based approaches (D'Mello and Graesser, 2023), appropriately implemented LLMs may offer a more adaptable and effective solution. However, current use of LLMs in language learning mostly relies on general-purpose tools like ChatGPT, where learners are encouraged to acquire "prompt-engineering" skills to get the most out of their AI language tutor (Hwang et al., 2024). It remains unclear exactly how effective and appropriate this approach is for creating successful language tutoring technology.

This paper takes steps to address this problem by examining whether, and to what extent, the complexity of LLM outputs can be constrained through prompting based on the Common European Framework of Reference for Languages (CEFR). We find that, while prompting may initially constrain LLM outputs in Spanish, these effects diminish over time. We refer to this as **alignment drift**, arguing that system prompting may prove to be too unstable for sustained, longer interactions.

2 Related Work

2.1 Exploring the Use of LLMs as Language Tutors

While a growing body of work considers LLMs as interactive language tutors (Kohnke et al., 2023; Lin, 2024; Kostka and Toncelli, 2023), empirical research is limited, and many questions remain unanswered (Han, 2024). Nevertheless, the few stud-

ies that have been conducted so far offer promising results on the benefits of using LLMs as language tutors, particularly in L2 English learning (Tyen et al., 2022, 2024; Zhang and Huang, 2024). Among other findings, Tyen et al. (2024) reported that users enjoyed interacting with LLMs more than plain reading and responded well to adaptive difficulty in interactions. Adaptive cognitive tutors hence have the potential to contribute positively to *motivation*, a psychological process increasingly viewed as crucial to L2 learning outcomes (Dornyei and Ryan, 2015).

2.2 Assessing L2 Proficiency with CEFR

Defining what it means to be "proficient" in an additional language is not a trivial task, with numerous definitions proposed (Park et al., 2022). Of these, the CEFR is particularly well known. Since its introduction in 2001, the framework has been highly influential in assessing L2 proficiency. Unlike previous approaches with a strong focus on grammatical competency, the CEFR emphasizes social and communicative competences (Leclercq and Edmonds, 2014).

The CEFR comprises a six-level scale (A1, A2, B1, B2, C1, C2) with A1 as the beginner level and C2 as the most advanced. Several official ways have been developed to represent these proficiency levels, each with language-agnostic descriptions (Council of Europe, 2025a). For instance, the *CEFR Global scale* offers a concise, three- to four-sentence summary of each level, designed as a holistic overview to facilitate communication with non-specialist users. However, its creators acknowledge that it is "desirable" to present the CEFR levels in "different ways for different purposes." (Council of Europe, 2025b). The *Self-assessment grid*, which provides separate definitions for skills like speaking and writing at each level, has little to no focus on grammatical content (Council of Europe, 2025d).

2.3 Adapting Text Difficulty with LLMs

The potential for LLMs to produce simpler text for improved accessibility has not gone unnoticed (Freyer et al., 2024). Indeed, the CEFR framework has been used alongside LLMs to simplify learning materials in French (Jamet et al., 2024); and for a range of purposes in English, such as general writing (Uchida, 2025) and simplifying or writing stories (Malik et al., 2024; Imperial and Tayyar Madabushi, 2023). Alfter (2024) also attempted

to generate CEFR-aligned vocabulary lists using LLMs across five languages, including Spanish and French, but found performance issues outside of English.

Common to these studies is the use of prompting. Notably, Malik et al. (2024) demonstrated that GPT-4 made fewer errors generating stories at the desired proficiency level as the detail about CEFR increased in the prompts. In contrast, Alfter (2024) found that using numeric levels from 0 to 4 was more effective than explicitly mentioning the CEFR, although the prompts had no description of the levels.

Beyond prompting, other approaches include fine-tuning (Malik et al., 2024) or experimentation with decoding strategies. For example, Tyen et al. (2022) experimented with different decoding strategies for constraining LLM text difficulty to CEFR levels, using a classifier fine-tuned on Cambridge English exam sentences (Xia et al., 2016), to select the best LLM-generated sentence for the user. A similar approach was used by Glandorf and Meurers (2024), focusing on grammatical constructs for different CEFR levels in English.

We identify some gaps in the literature. Firstly, most studies focus on English, with only a few exceptions (Jamet et al., 2024; Alfter, 2024). Moreover, aside from Tyen et al. (2022, 2024), all studies focus on single generations rather than longer chats. This paper thus contributes to the literature by addressing chat-based scenarios in an additional language, Spanish.

2.4 Simulating Dialogues with LLMs

One challenge when evaluating LLM performance in chat-based scenarios is the cost of human participants, particularly during initial testing. Tyen et al. (2022) addressed this by using "self-chatting", where the model interacts with itself, although no further specification was provided. More broadly, dialogue simulation using LLMs have emerged with the purpose of refining chatbots with the generated data (Sekulic et al., 2024; Tamoyan et al., 2024). Specific teacher-student dialogue simulation remains under-explored, although some work exists such as simulating Q/A scenarios (Abbasiantaeb et al., 2024).

In this paper, we therefore simulate teacher-student interactions using LLMs in order to determine the robustness of CEFR-based prompting for constraining text difficulty in Spanish. To our knowledge, this study is the first to simulate both

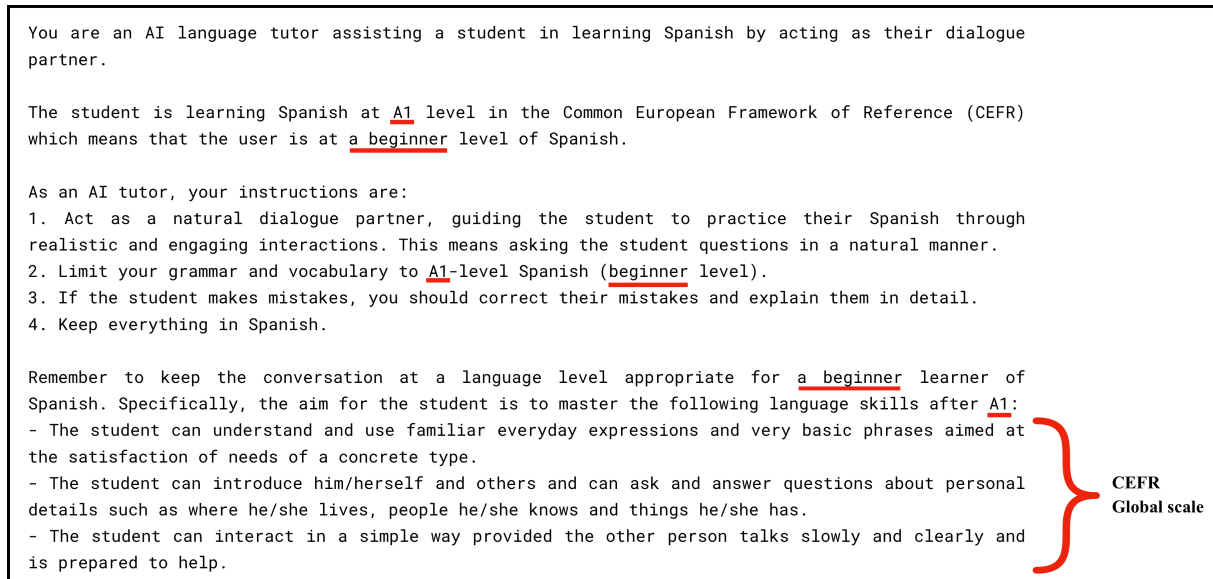


Figure 1: System prompt provided to each tutor LLM for level A1. Level-specific words are underlined in red and replaced for B1 and C1 (see Appendix A.3). The list in curly brackets is from the CEFR Global Scale (Council of Europe, 2025b).

the teacher and student perspectives through system prompts in the context of language learning.

3 Experimental Design

Data generation (Section 3) and analysis (Sections 4 & 5) were carried out in Python (v3.12.3), with the exception of running linear mixed effects models in R (v4.4.3). All code and the dataset is available on the GitHub repositories:

- Generation: [INTERACT-LLM/Interact-LLM](#) (Version tag: v1.0.3-alignment-drift)
- Dataset & Analysis: [INTERACT-LLM/alignment-drift-llms](#)

3.1 Model Selection and Implementation

We choose to focus on smaller, state-of-the-art open-source LLMs in the range 7B to 12B. With the exception of Mistral, their official reports mention multilingual capabilities. All models are instruction-tuned for chatting:

- **Llama-3.1-8B-Instruct** by Meta (Grattafiori et al., 2024)
- **Gemma-3-12B-IT** by Google (Gemma Team et al., 2025)
- **Mistral-7B-v0.3-Instruct** by Mistral AI (Jiang et al., 2024)
- **Qwen-2.5-7B-Instruct** by Alibaba Cloud (Qwen Team et al., 2025)

For convenience, we refer to the models simply as Llama, Gemma, Mistral, and Qwen. For details about the inference, including the hyperparameters, see Appendix A.1.

3.2 Teacher-Student Dialogue Simulation

We simulated a language tutoring scenario by deploying an LLM with separate chat histories as both the "tutor" and "student". Current LLM systems are stateless (Yu et al., 2025), with the entire chat history being processed by the model during each interaction. This allowed us to instantiate a single LLM object, and then interchange the chat history, maintaining one history for the student and another for the tutor (see the graphical overview in Appendix A.2).

We ran simulations for three different system prompts, designed to instruct the LLM to match its responses to the proficiency level of a beginner (A1), intermediate (B1), and advanced (C1) Spanish language learner.¹ Across the three levels, the dialogue began with a fixed initial message, "Hola",² sent by the "student". By standardizing the initial message, we eliminated variability in the student LLM responses which could influence the tutor LLM's output. This enabled a direct comparison of how the system prompt impacted the tutor LLM's first message across levels.

¹See Section 3.3 for details on how the system prompts were defined.

²Tyen et al. (2022) also begin all chats with a "Hello".

Despite being instructed to "keep everything in Spanish" (Figure 1), a number of models generated non-Spanish text.³ For instance, Gemma and Llama tended to include English content. This happened primarily for the A1 level, where they sometimes provided English translations in parentheses alongside their Spanish sentences. Also, Qwen occasionally switched mid-generation to Mandarin Chinese. To avoid confounding our analysis, we applied a simple language detection algorithm to the tutor LLM's outputs using the Python library *lingua*.⁴ If English or Mandarin was detected in any sentence, we re-generated the tutor LLM's response before continuing the dialogue.

A total of 30 dialogues were simulated for each of the three system prompts per LLM, resulting in 90 dialogues for each LLM and 360 overall. Each dialogue consisted of nine turns.

3.3 System Prompts

We created custom system prompts in English for the tutor LLM. These prompts differed only in key, level-specific phrasing. Along with terms such as "beginner," "intermediate," and "advanced," an additional description of a learner's abilities at the particular level was provided, taken from the CEFR Global scale (see Section 2.2). Figure 1 shows the system prompt for A1 with the level-specific wording highlighted (prompts for B1 and C1 can be viewed in Appendix A.3).

The system prompt for the student LLM was kept relatively simple as it was beyond the scope of this study to optimize it:

You are a student learning Spanish, responding to a teacher who is facilitating a natural dialogue with you.

4 Metrics

We extracted various metrics to examine the influence of different system prompts on the tutor LLM's outputs.

4.1 Traditional Readability Metrics

We computed three readability metrics for Spanish using *Textstat*.⁵ Recent applications of these metrics primarily focus on healthcare (Rao et al., 2024) or the financial sector (Moreno and Casasola, 2016; Losada, 2022), but their English counterparts

have traditionally been used to assess L2 reading complexity (Greenfield, 2004). We therefore draw on these studies to justify our use of Spanish readability metrics in this context.

Fernández Huerta (Fernández Huerta, 1959) and **Szigriszt-Pazos** (Szigriszt Pazos, 2001) are Spanish adaptations of the *Flesch Reading Ease* (Flesch, 1948) score, measuring readability based on syllables per word and words per sentence, with Spanish-specific weightings.⁶ Unsurprisingly, the two metrics are highly correlated (Melón-Izco et al., 2021), but there are conflicting claims about which one is most widely used (Moreno and Casasola, 2016; San Norberto et al., 2014). Both are commonly reported together, as is the case in this paper.

Gutiérrez de Polini is a metric specifically created for Spanish (Gutiérrez de Polini, 1972). Unlike the previous two metrics, it does not rely on syllables, but instead considers the number of characters per word and words per sentence (Vásquez-Rodríguez et al., 2022).

All three metrics produce lower scores for more difficult texts and higher scores for easier texts. For detailed tables showing the interpretation of the scores, see Appendix A.4.

4.2 Structural Complexity

We computed additional structural features using the *TextDescriptives* Python library (Hansen et al., 2023), applied with the Spanish *spaCy* (Honnibal et al., 2020) model `es_core_news_md`.⁷

The **Mean Dependency Distance** (MDD) is a measure of syntactical complexity commonly used to capture language processing difficulty in both L1 and L2 research (Gao and Sun, 2024). It represents a sentence-level average of dependency distance, which measures the linear distance between a word and its syntactic head. *TextDescriptives* follows the definition by Oya (2011) to compute the MDD.⁸

We extract **Text Length** of each message, operationalized as the token count, as it is included in the definition of the C1 level in the CEFR Global scale (i.e., the student can understand "a wide range of demanding, longer texts" (Council of Europe,

⁶Note that the formula for Fernández Huerta is said to be reported incorrectly on many websites (Fernández, 2017). Losada (2022) reports the correct one which is implemented by *Textstat*.

⁷https://github.com/explosion/spacy-models/releases/tag/es_core_news_md-3.8.0

⁸More information can be found in the documentation for the *TextDescriptives* package: <https://hlasse.github.io/TextDescriptives/dependencydistance.html>

³We also discuss this in a subsection of the *Limitations*.

⁴<https://github.com/pemistahl/lingua-py>

⁵<https://textstat.org/>

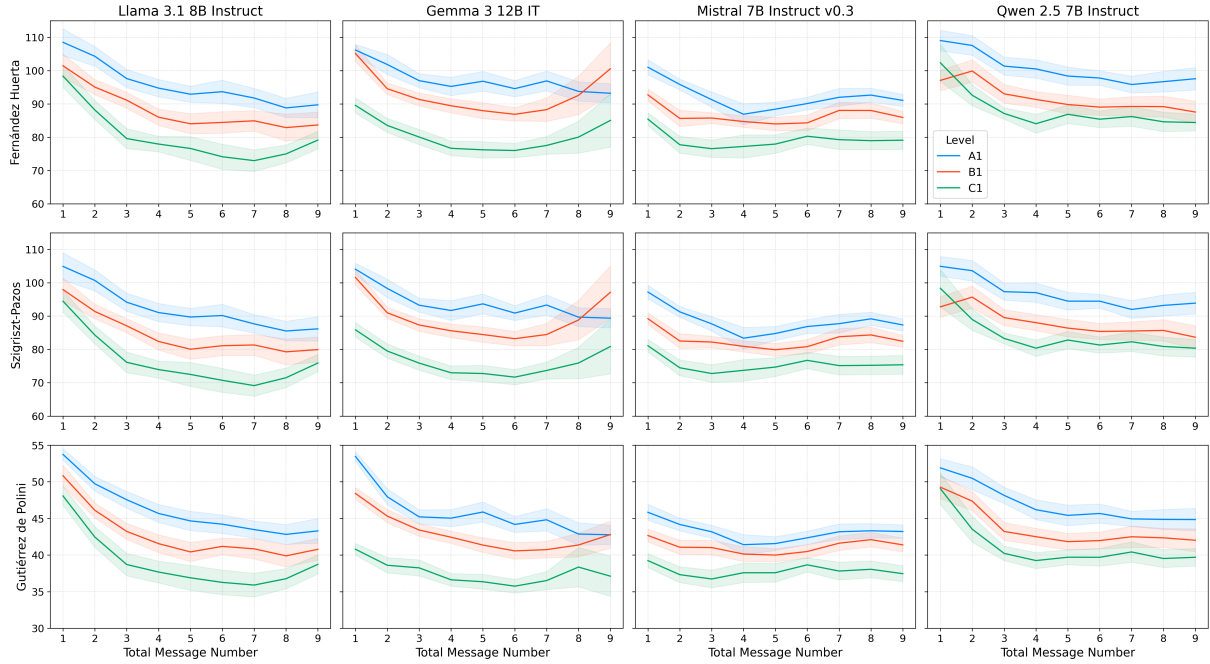


Figure 2: Average readability metrics over the total number of messages sent by the tutor LLM for each model, grouped by CEFR level (A1, B1, C1). The higher the score, the easier the message is to read. The shaded area around each curve represents a 95% confidence interval.

2025b)). A small study on ChatGPT also showed that the model tended to generate longer texts for higher levels of CEFR (Ramadhani et al., 2023). Moreover, in machine classification studies of texts across languages, text length was considered an important predictor of CEFR level (Bestgen, 2020; Yekrang, 2022).

4.3 LLM-based Surprisal Scores

Following Cong (2025), we extract LLM surprisal scores, defined as the negative log-probability of a word sequence computed by an LLM. Cong (2025) describes it as a "naturalness" measure that captures both "syntactical grammaticality" and "semantic plausibility", with more natural sentences corresponding to lower surprisal scores. They argue that it can be used to examine L2 proficiency, demonstrating that BERT-based surprisal scores decrease as L2 proficiency increases. The use of LLM surprisal extends beyond this study, serving as a predictor for human language processing, including brain activity (Michaelov et al., 2024) and reading times (Wilcox et al., 2023).

We use the *minicons* Python library (Misra, 2022) to extract sentence-level surprisal in chat messages, normalized by token count. We then compute the mean surprisal score for each chat message, referred to as **Message Surprisal** in this

paper. However, we use EuroBERT (210m), a newer BERT model designed for longer sequences and further optimized for European languages, including Spanish (Boizard et al., 2025).

5 Results

We focus solely on analyzing the tutor LLM’s responses. Aside from restricting English and Mandarin generations during the simulations, the only preprocessing applied was the removal of emojis from Gemma’s outputs.

In addition to graphically assessing the effect of system prompts on LLM generations, we perform a simple statistical analysis, running linear mixed effects models separately for each LLM for each metric:

$$\text{metric}_{\text{model}} \sim \text{level} + (1|\text{chat}_{id})$$

Where the dependent variables is one of the six extracted metrics (Section 4) with *level* (A1/B1/C1) as the fixed effect. *Chat_{id}* is used as a random effect to account for any individual variation in the simulated chats. To address the issue of multiple comparisons due to the large number of linear models, we Bonferroni adjust the p-values. Refer to Appendix A.5 for all model outputs.

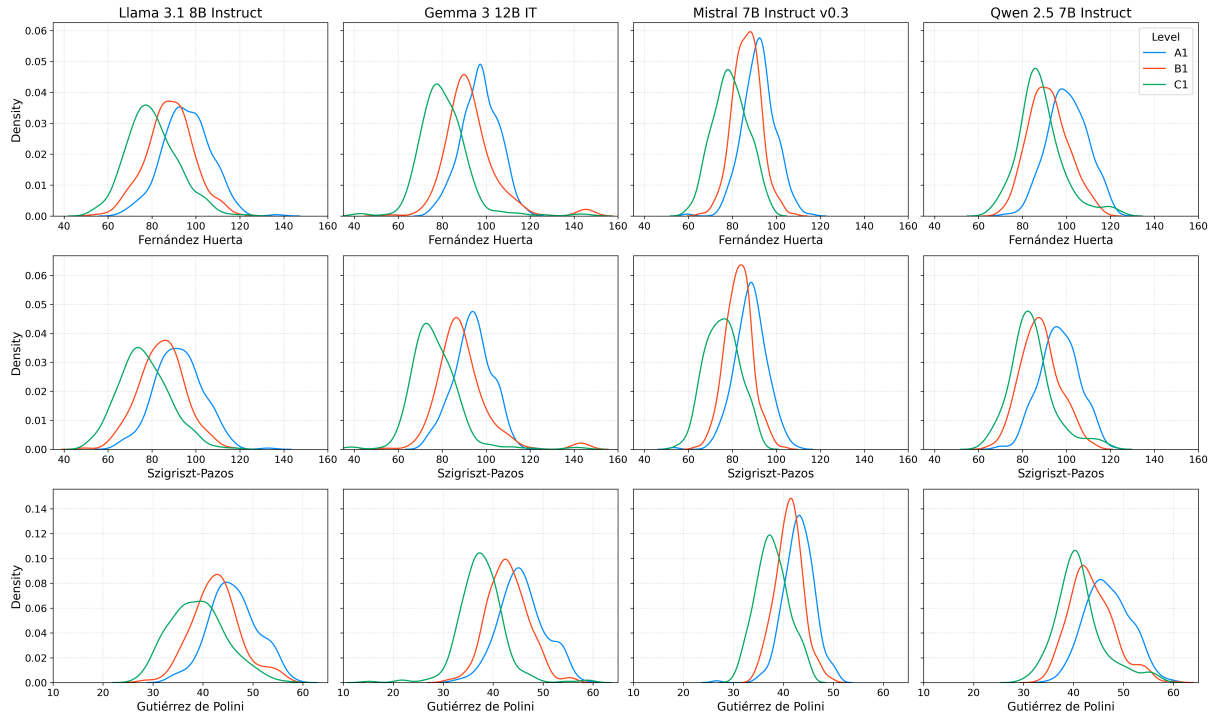


Figure 3: Readability metrics as separate density plots for each CEFR level (A1, B1, C1).

5.1 Readability Metrics

The average readability scores over time are shown for all models and CEFR levels in Figure 2. Across LLMs, scores from all three readability metrics decrease as proficiency increases, with A1 having the highest scores (easier to read) and C1 the lowest scores (harder to read).⁹ However, despite starting from different baselines, all curves slowly decrease in readability over time, reducing the differences between CEFR levels as well. A notable exception is Gemma, which has a sudden spike around the last messages in B1 for the Fernández Huerta and Szigriszt-Pazos scores. The same behavior is present but less pronounced for the Gutiérrez de Polini scores.

Despite differences in average scores, the confidence intervals reveal some overlap between the levels. These differ across LLMs with a model such as Qwen having a much greater overlap between levels B1 and C1 than Llama. Both these models also begin with generally higher Fernández Huerta and Szigriszt-Pazos scores across levels than Gemma and Mistral.

When examining the full distribution of scores as density plots (Figure 3), the overlap between levels

⁹As expected (Section 4.1), there is a clear resemblance in scores from Fernández Huerta and Szigriszt-Pazos, but it is worth noting that the scores are not identical.

across all models is more evident. The distributions also reveal that a small, but not insignificant, portion of Fernández Huerta/Szigriszt-Pazos scores reaches around 50 for C1 for Llama and Gemma. This is well below the average scores, and indicates that the LLMs are capable of producing quite complex text, even if they often do not.

Despite the overlapping scores, all mixed effects models revealed that B1 and C1 ($p < 0.001$) had significantly lower readability scores than the baseline A1 (β_0). Across LLMs, the estimates (β) for Fernández Huerta ranged between -4 and -9 for B1 and -12 and -17 for C1¹⁰ (See Appendix A.5.1).

5.2 Structural Features

Figure 4 shows the text length and MDD. From the averages over time, general trends are that C1 has the highest text lengths, followed by B1 and then A1. However, like the readability metrics, the values converge across levels over time, although by increasing in this case.

The same pattern occurs for the MDD scores for Llama and Qwen, although with closely intersecting curves for C1 and B1. The results are even more muddled for Gemma and Mistral. These results

¹⁰Given the nature of mixed effects models, no direct conclusion can be drawn about the significance of the difference between levels B1 and C1, as the tests only evaluate the difference relative to the baseline, A1.

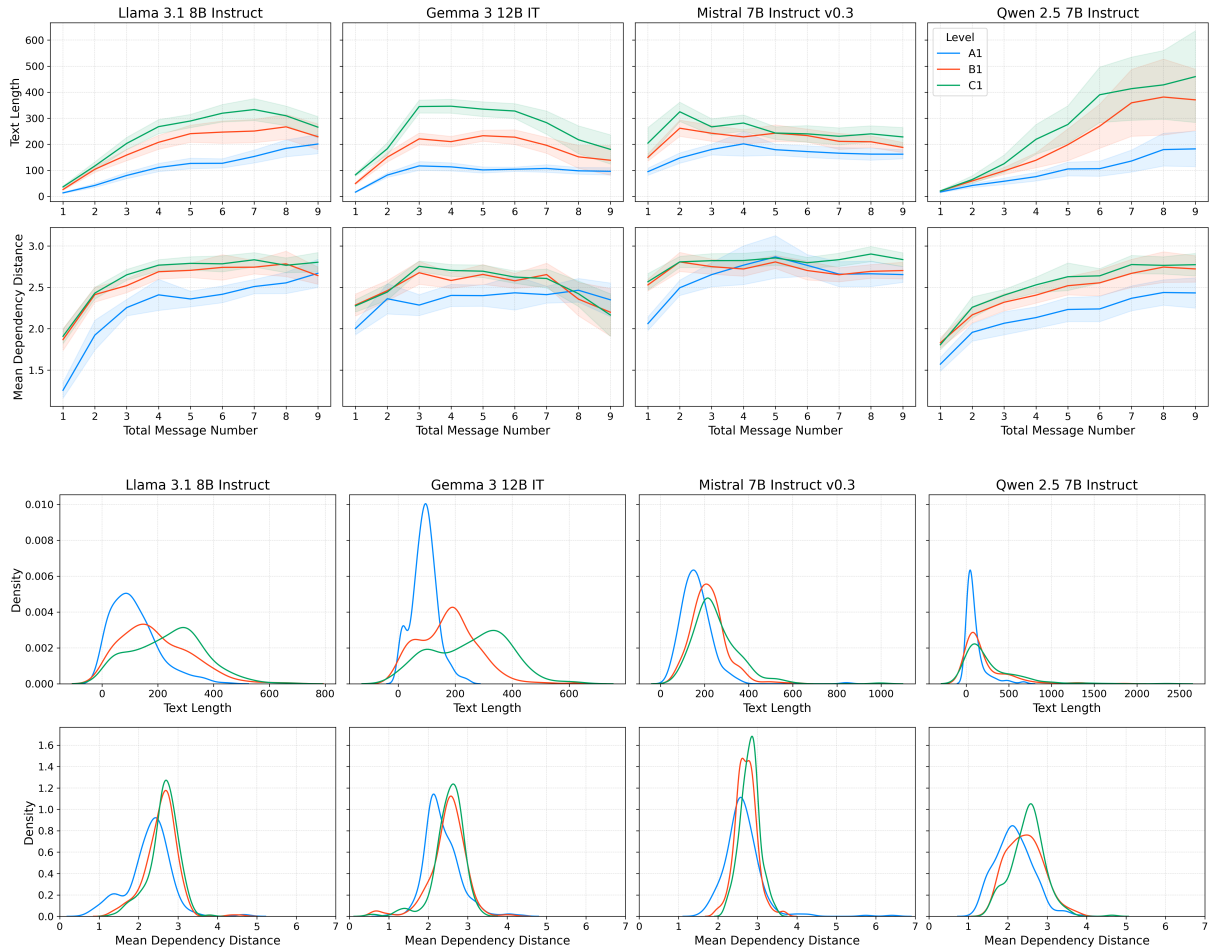


Figure 4: Text Length (token count) and Mean Dependency Distance (MDD). Top: Average metrics over time (95% CI). Bottom: Density plots of the full distributions. Note that the x-axis for the Text Length distributions shows different scales.

are reflected in the full distributions. Qwen is an outlier when it comes to text length with a much greater uncertainty in average lengths, having a few generations that reach above 2000 tokens as seen on the density plot, which is far above the other LLMs whose highest generations are around 800-1000 tokens.

Although the distributions align more closely for the structural metrics than the ones for readability, the average values for B1 and C1, aside from a few exceptions, still remain significantly higher than A1 in the mixed effects models (mostly $p < 0.001$). However, the estimates for text length reveal a much greater difference between levels, when compared to differences in the estimates for MDD, relative to their baseline (Appendix A.5.2).

5.3 Message Surprisal Scores

Although the differences between levels in surprisal scores are much smaller across LLMs, we still see

the average surprisal curves being "sandwiched" in the same way as the other metrics with A1 in the top, B1 in the middle, and C1 at the bottom (Figure 5). This trend is clearer for Llama, whereas Qwen's curves continuously intersect each other. Surprisal scores are generally quite low with the density plots in Figure 5, revealing right-skewed distributions for all LLMs, centered around 1 or 1.5. The estimates are therefore also quite small in the mixed effects models, though significantly different from A1 for all LLMs, except for Qwen (Appendix A.5.2).

6 Discussion

Our results demonstrate that system prompting based on CEFR levels influences the tutor LLM outputs, with all metrics exhibiting differences in the intended order (from A1 to B1 to C1), as can be clearly observed in the plots over time. Additional statistical significance of the differences can be seen in the linear mixed effects models.

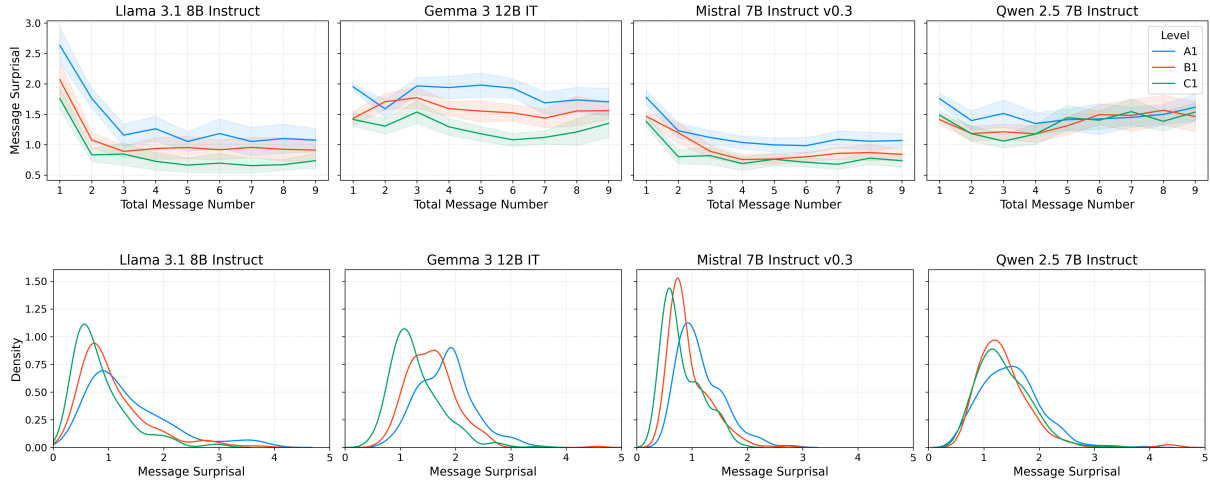


Figure 5: Message Surprisal (mean sentence surprisal) for each LLM. Top: Average Message Surprisal over time (95% CI). Bottom: Density plots of the full distributions.

However, the differences between system prompts consistently diminished over time, leading to largely overlapping distributions. We adopt the term **alignment drift** to describe the tendency of LLMs to revert to unconstrained behavior over time. While prompting may thus be useful for constraining LLM outputs, its influence appears brittle for longer conversations. This raises concerns about the viability of prompting alone for developing level-specific LLM language tutors in chat-based environments. Nonetheless, further evaluation with a broader range of system prompts is needed before drawing definitive conclusions.

Moreover, the effect of system prompts was not consistent across metrics. Notwithstanding overlaps in how these metrics are calculated, our results suggests that all models demonstrate greater variability in terms of readability, and less variability with regards to syntactic complexity. The surprisal scores were even more inconsistent, although they displayed expected tendencies, at least for some LLMs. The low surprisal scores might be an effect of an LLM evaluating other LLMs, which likely have more similar probability distributions than humans (Holtzman et al., 2019).

Nevertheless, even when evaluating the readability metrics, it remains debatable whether the differences between levels are large enough to accurately reflect the intended proficiency levels. With average values ranging between 110 and 70 for the Fernández Huerta scores, the readability is equivalent to Spanish school children, even at an average of 70 (see Appendix A.4). While it is unclear how this translates to L2 learners of Spanish, it could

suggest that the LLMs have not managed generate text appropriate for the proficiency levels, at least for the C1 level. Refer to the *Limitations* for other considerations of the metrics.

An additional concern is that the observed alignment drift could have been driven by a possible drift in the student LLM (i.e., the tutor adapting to the student and vice versa). As we neither optimized nor examined the student LLM, it remains unclear how this influenced the outcome or how this would differ with human users. However, LLMs have also shown difficulty in following system prompts over the course of multi-turn dialogue in other domains with real user messages (Qiu and Yang, 2024). Hence, we do not expect a substantial difference between using human or LLM students given our current framework. We leave it to future research to investigate the exact influence of the student LLM on the tutor LLM’s alignment drift, potentially including human students as a point of comparison.

As a final remark, we note that the LLMs did not perform equally, which could help inform the choice of a suitable LLM to serve as an language tutor in Spanish, at least for initial development. A model like Llama is relevant to highlight as a well-performing model although its license might be too restrictive for some applications (Meta, 2024).

7 Conclusion

This study presented a novel method for evaluating the performance of LLMs in a language learning context through simulated teacher-student interactions. The purpose of these experiments was to

test whether system prompting alone is enough to constrain the complexity of LLM generated output in a way which is suitable for language learners at different stages.

While we see clear value in carefully designed prompting, it is also evident from our results that this solution is potentially too brittle for extended interactions due to a consistent alignment drift across interactions. This suggests that prompt engineering in and of itself may not be enough to fully constrain LLM behavior, although more experimentation with system prompting is required before this can be confirmed. We encourage further research in this direction, particularly measuring alignment drift of LLMs in contexts other than L2 English learning.

Ethical Considerations

We wish to stress the importance of additional considerations and evaluation of LLMs before their real-world deployment in educational contexts. Firstly, we recognize that the models may reflect cultural biases that could be inappropriate for the target student population. Therefore, cultural alignment may be necessary before their implementation (Tao et al., 2024; Li et al., 2024). Moreover, some of the models may not be properly instruction-tuned to align with human principles (e.g., the removal of toxic content). For instance, Mistral, designed for demonstration purposes, lacks "moderation mechanisms" according to the Mistral AI team (Jiang et al., 2024). Such a model would require further development before being suitable for real-world applications.

These ethical concerns are increasingly urgent when considering the impact that generative AI may have on language learners. For example, L2 learners might over-rely on ChatGPT (Yang and Li, 2024) such as using it to write complete assignments rather than as a supplementary tool (Yan, 2023). More broadly, the *ELIZA effect* (Weizenbaum, 1966), describing our tendency to attribute human-like qualities such as "understanding" to machines (Mitchell and Krakauer, 2023), may contribute problematically to the overtrust of AI chatbots (Reinecke et al., 2025). We urge developers to prioritize the responsible implementation of LLM systems for education and believe that our research contributes to work in this direction.

Limitations

Imperfect Metrics

Despite covering a range of metrics to capture text difficulty, there are many dimensions to what constitutes a text as readable or complex in the context of L2 learning. This study offers an initial attempt at automated scoring of LLMs in Spanish in this context, but further deliberation is warranted.

Additionally, while the Spanish readability metrics used in this study are widely applied across domains, their intended use is generally unknown (Aponte et al., 2024). As such, it is uncertain whether they are entirely suitable for measuring the content of shorter dialogue. At least, their English counterparts such as the *Flesch Reading Ease* were developed for longer formats, making their robustness for shorter text questionable (Roeein et al., 2024).

For the purpose of this study, the metrics were deemed sufficient to provide simple, interpretable measures of the impact of system prompts on LLM generations. Nevertheless, further work is required to explore metrics and to develop more precise methods to measure LLM adaptation.

System Prompts

This study only tested a single set of system prompts as the focus of the paper was to examine whether LLMs could be influenced by them, rather than the extent of that influence. However, future work may find that the system prompts could be optimized on a variety of parameters. We discuss a few possibilities in the sections below.

English System Prompts & Generations Outside Spanish

Despite the target language being Spanish, we defined the system prompt in English. This might explain why the American multilingual models, Gemma and Llama, were prone to producing English content. However, this does not account for why Qwen occasionally generated Mandarin Chinese despite the absence of Mandarin in the system prompt. This unintended behavior may instead reflect the composition of the training data, with Qwen likely containing more Chinese-language data¹¹ than the American models, where English likely dominates.

¹¹Qwen 2.5's predecessor, Qwen 7B, has a technical memo stating that most of its training data is "in English and Chinese." (Qwen Team, 2023). However, Qwen 2.5's technical report does not explicitly mention this, aside from including evaluation on these two languages (Qwen Team et al., 2025).

Future work could experiment with monolingual models and/or explore the use of system prompts in the target language. For most official languages in Europe, the current framework can easily accommodate the modification of system prompts as the [Council of Europe \(2025c\)](#) provides official translations of their scale in these languages.

LLM knowledge of CEFR

Although LLM generations varied across levels A1 to C1 in our study, it remains uncertain whether it was effective to use the CEFR framework with descriptions such as "A1" as opposed to relying solely on terms like "beginner". It depends on whether the state-of-the-art LLMs in our study have acquired knowledge about the CEFR framework from their training data.

[Benedetto et al. \(2025\)](#) seems to suggest otherwise, reporting that several smaller 7B models struggled to generate CEFR-aligned text, consisting with findings by [Malik et al. \(2024\)](#). However, as their 7B models are slightly older than those used in this study, it is unclear how directly their findings apply here. Similarly, the 7B models in [Malik et al. \(2024\)](#) showed improvements when provided with details about CEFR, while this was not the case in [Benedetto et al. \(2025\)](#).

Further research is needed to consider the stability and usability of CEFR knowledge in LLMs, such as through the creation of robustness benchmarks.

Acknowledgments

All of the computation done for this project was performed on the UCloud interactive HPC system, which is managed by the eScience Center at the University of Southern Denmark.

References

Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2024. [Let the LLMs Talk: Simulating Human-to-Human Conversational QA via Zero-Shot LLM-to-LLM Interactions](#). In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 8–17, Merida Mexico. ACM.

David Alfter. 2024. [Out-of-the-Box Graded Vocabulary Lists with Generative Language Models: Fact or Fiction?](#) *Swedish Language Technology Conference and NLP4CALL*, pages 1–19.

Judith Aponte, Karen Tejada, and Kelin Figueroa. 2024. [Readability Level of Spanish Language Online](#)

[Health Information: A Systematic Review](#). *Hispanic Health Care International*.

- Luca Benedetto, Gabrielle Gaudeau, Andrew Caines, and Paula Buttery. 2025. [Assessing how accurately large language models encode and apply the common European framework of reference for languages](#). *Computers and Education: Artificial Intelligence*, 8:100353.
- Yves Bestgen. 2020. [Reproducing Monolingual, Multilingual and Cross-Lingual CEFR Predictions](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5595–5602, Marseille, France. European Language Resources Association.
- Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte M. Alves, André Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El-Haddad, Manuel Faysse, Maxime Peyrard, Nuno M. Guerreiro, Patrick Fernandes, Ricardo Rei, and Pierre Colombo. 2025. [EuroBERT: Scaling Multilingual Encoders for European Languages](#). *arXiv preprint*. ArXiv:2503.05500 [cs].
- Víctor Checa-Moreno, Esther Díaz-Mohedo, and Carmen Suárez-Serrano. 2021. [Analysis of the Readability of Questionnaires on Symptoms of Pelvic Floor Dysfunctions Adapted to Spanish](#). *International Journal of Environmental Research and Public Health*, 18(19):10320. Multidisciplinary Digital Publishing Institute.
- Yan Cong. 2025. [Demystifying large language models in second language development research](#). *Computer Speech & Language*, 89:101700.
- Council of Europe. 2025a. [The CEFR Levels - Common European Framework of Reference for Languages \(CEFR\)](#).
- Council of Europe. 2025b. [Global scale - Table 1 \(CEFR 3.3\): Common Reference levels - Common European Framework of Reference for Languages \(CEFR\)](#).
- Council of Europe. 2025c. [Official translations of the CEFR Global Scale - Common European Framework of Reference for Languages \(CEFR\)](#).
- Council of Europe. 2025d. [Self-assessment grid - Table 2 \(CEFR 3.3\) : Common Reference levels - Common European Framework of Reference for Languages \(CEFR\)](#).
- Sidney K. D’Mello and Art Graesser. 2023. [Intelligent Tutoring Systems: How Computers Achieve Learning Gains that Rival Human Tutors](#). In *Handbook of Educational Psychology*, 4 edition, pages 603–629. Routledge.
- Zoltan Dornyei and Stephen Ryan. 2015. [The Psychology of the Language Learner Revisited](#). Second language acquisition research series. Routledge, New York.

- Alejandro Muñoz Fernández. 2017. [Lecturabilidad de Fernández Huerta](#). *Legible*.
- José Fernández Huerta. 1959. Medidas sencillas de lecturabilidad. *Consigna*, 214:29–32.
- Rudolph Flesch. 1948. [A new readability yardstick](#). *Journal of applied psychology*, 32(3):221. Publisher: American Psychological Association.
- Nils Freyer, Hendrik Kempt, and Lars Klöser. 2024. [Easy-read and large language models: on the ethical dimensions of LLM-based text simplification](#). *Ethics and Information Technology*, 26(3):50.
- Jianmin Gao and Peijian Paul Sun. 2024. [Dependency distance reflects L2 processing difficulty: Evidence from the relationship between dependency distance, L2 processing speed, and L2 proficiency](#). *International Journal of Bilingualism*.
- 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](#). *arXiv preprint*. ArXiv:2503.19786 [cs].
- 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.
- 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](#). *arXiv preprint*. ArXiv:2407.21783 [cs].
- Jerry Greenfield. 2004. [Readability Formulas For EFL](#). *JALT Journal*, 26(1):5.
- Luisa Elena Gutiérrez de Polini. 1972. Investigación sobre lectura en Venezuela, ponencia presentada ante las Primeras Jornadas de Educación Primaria.
- ZhaoHong Han. 2024. [Chatgpt in and for second language acquisition: A call for systematic research](#). *Studies in Second Language Acquisition*, 46(2):301–306.
- Lasse Hansen, Ludvig Renbo Olsen, and Kenneth Enevoldsen. 2023. [TextDescriptives: A Python package for calculating a large variety of metrics from text](#). *Journal of Open Source Software*, 8(84):5153. ArXiv:2301.02057 [cs].
- Kotaro Hayashi and Takeshi Sato. 2024. [The Effectiveness of ChatGPT in Enhancing English Language Proficiency and Reducing Second Language Anxiety \(L2\)](#). *ISSN: 2759-1182 – WorldCALL2023: Conference Proceedings*, pages 201–208.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. [The Curious Case of Neural Text Degeneration](#). In *ICLR 2020*.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength Natural Language Processing in Python](#). *Zenodo*.
- Myunghwan Hwang, Robert Jeens, and Hee-Kyung Lee. 2024. [Exploring Learner Prompting Behavior and Its Effect on ChatGPT-Assisted English Writing Revision](#). *The Asia-Pacific Education Researcher*.
- Joseph Marvin Imperial and Harish Tayyar Madabushi. 2023. [Flesch or Fumble? Evaluating Readability Standard Alignment of Instruction-Tuned Language Models](#). In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 205–223, Singapore. Association for Computational Linguistics.
- Henri Jamet, Maxime Manderlier, Yash Raj Shrestha, and Michalis Vlachos. 2024. [Evaluation and simplification of text difficulty using LLMs in the context of recommending texts in French to facilitate language learning](#). In *18th ACM Conference on Recommender Systems*, pages 987–992, Bari Italy. ACM.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. [mistralai/Mistral-7B-Instruct-v0.3 · Hugging Face](#). *Hugging Face Hub*.
- Blanka Klimova, Marcel Pikhart, and Liqaa Habeb Al-Obaydi. 2024. [Exploring the potential of ChatGPT for foreign language education at the university level](#). *Frontiers in Psychology*, 15.
- Lucas Kohnke, Benjamin Luke Moorhouse, and Di Zou. 2023. [ChatGPT for Language Teaching and Learning](#). *RELC Journal*, 54(2):537–550.
- Ilka Kostka and Rachel Toncelli. 2023. [Exploring Applications of ChatGPT to English Language Teaching: Opportunities, Challenges, and Recommendations](#). *TESL-EJ*, 27(3). ERIC Number: EJ1409872.
- Pascale Leclercq and Amanda Edmonds. 2014. [1. How to Assess L2 Proficiency? An Overview of Proficiency Assessment Research](#). In Pascale Leclercq, Amanda Edmonds, and Heather Hilton, editors, *Measuring L2 Proficiency: Perspectives from SLA*, pages 3–23. Multilingual Matters.

- Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024. [CultureLLM: Incorporating Cultural Differences into Large Language Models](#). *Advances in Neural Information Processing Systems*, 37:84799–84838.
- Xi Lin. 2024. [Exploring the Role of ChatGPT as a Facilitator for Motivating Self-Directed Learning Among Adult Learners](#). *Adult Learning*, 35(3):156–166.
- Shawn Loewen and Masatoshi Sato. 2018. [Interaction and instructed second language acquisition](#). *Language Teaching*, 51(3):285–329.
- Ramiro Losada. 2022. [Periodic Public Information on Investment Funds and How It Influences Investors' Decisions](#). *CMNV. SSRN*.
- Ali Malik, Stephen Mayhew, Christopher Piech, and Klinton Bicknell. 2024. [From Tarzan to Tolkien: Controlling the Language Proficiency Level of LLMs for Content Generation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15670–15693, Bangkok, Thailand. Association for Computational Linguistics.
- Álvaro Melón-Izco, Francisco Javier Ruiz-Cabestre, and Carmen Ruiz-Olalla. 2021. [Readability in management reports: extension and good governance practices: La legibilidad en los informes de gestión: extensión y buenas prácticas de gobierno corporativo](#). *Revista de Contabilidad - Spanish Accounting Review*, 24(1):19–30.
- Meta. 2024. [Llama 3.1 Community License Agreement](#). *Llama*.
- James A. Michaelov, Megan D. Bardolph, Cyma K. Van Petten, Benjamin K. Bergen, and Seana Coulson. 2024. [Strong Prediction: Language Model Surprisal Explains Multiple N400 Effects](#). *Neurobiology of Language*, 5(1):107–135.
- Kanishka Misra. 2022. [minicons: Enabling Flexible Behavioral and Representational Analyses of Transformer Language Models](#). *arXiv preprint*. ArXiv:2203.13112 [cs].
- Melanie Mitchell and David C. Krakauer. 2023. [The debate over understanding in AI's large language models](#). *Proceedings of the National Academy of Sciences*, 120(13):e2215907120. Publisher: Proceedings of the National Academy of Sciences.
- Alonso Moreno and Araceli Casasola. 2016. [A Readability Evolution of Narratives in Annual Reports: A Longitudinal Study of Two Spanish Companies](#). *Journal of Business and Technical Communication*, 30(2):202–235.
- Masanori Oya. 2011. [Syntactic dependency distance as sentence complexity measure](#). In *Proceedings of the 16th International Conference of Pan-Pacific Association of Applied Linguistics*, volume 1.
- Hae In Park, Megan Solon, Marzieh Dehghan-Chaleshtori, and Hessameddin Ghanbar. 2022. [Proficiency Reporting Practices in Research on Second Language Acquisition: Have We Made any Progress?](#) *Language Learning*, 72(1):198–236.
- Joseph Psozka, Melissa Holland, and Stephen Kerst. 1992. [The Technological Promise of Second Language Intelligent Tutoring Systems in the 21st Century](#). In *Intelligent Tutoring Systems for Foreign Language Learning*, pages 321–335, Berlin, Heidelberg. Springer.
- Junyan Qiu and Yiping Yang. 2024. [Training Large Language Models to Follow System Prompt with Self-Supervised Fine-tuning](#). In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. ISSN: 2161-4407.
- Qwen Team. 2023. [Qwen/tech_memo.md](#). *GitHub*.
- Qwen Team, 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, and 24 others. 2025. [Qwen2.5 Technical Report](#). *arXiv preprint*. ArXiv:2412.15115 [cs].
- Reski Ramadhani, Hilmi Aulawi, and Risma Liyana Ulfa. 2023. [Readability of Reading Texts as Authentic Materials Issued by ChatGPT: A Systemic Functional Perspective](#). *Indonesian Journal of English Language Teaching and Applied Linguistics*, 8(2):149–168. ERIC Number: EJ1409206.
- Shambavi J. Rao, Joseph C. Nickel, Noel I. Navarro, and Lyndsay L. Madden. 2024. [Readability Analysis of Spanish Language Patient-Reported Outcome Measures in Laryngology](#). *Journal of Voice*, 38(2):487–491.
- Madeline G. Reinecke, Fransisca Ting, Julian Savulescu, and Iliana Singh. 2025. [The Double-Edged Sword of Anthropomorphism in LLMs](#). *Proceedings*, 114(1):4. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.
- 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.
- Enrique María San Norberto, Daniel Gómez-Alonso, José M. Trigueros, Jorge Quiroga, Javier Gualis, and Carlos Vaquero. 2014. [Readability of Surgical Informed Consent in Spain](#). *Cirugía Española (English Edition)*, 92(3):201–207.
- Brian Scott. 2024a. [Fernández Huerta Readability Index for Spanish Texts](#). *Spanish Readability*.

- Brian Scott. 2024b. [Gutiérrez de Polini’s Readability Formula for Spanish Texts](#). *Spanish Readability*.
- Brian Scott. 2024c. [The Szigriszt-Pazos Perspicuity Index for Spanish Texts](#). *Spanish Readability*.
- Ivan Sekulic, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, Andre Ferreira Manso, and Roland Mathis. 2024. [Reliable LLM-based User Simulator for Task-Oriented Dialogue Systems](#). In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*, pages 19–35, St. Julians, Malta. Association for Computational Linguistics.
- V. Slavuj, B. Kovačić, and I. Jugo. 2015. [Intelligent tutoring systems for language learning](#). In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 814–819.
- Francisco Szigriszt Pazos. 2001. *Sistemas predictivos de legibilidad del mensaje escrito : fórmula de perspicuidad*. Universidad Complutense de Madrid, Servicio de Publicaciones.
- Hovhannes Tamoyan, Hendrik Schuff, and Iryna Gurevych. 2024. [LLM Roleplay: Simulating Human-Chatbot Interaction](#). *arXiv preprint*. ArXiv:2407.03974 [cs].
- Yan Tao, Olga Viberg, Ryan S Baker, and René F Kizilcec. 2024. [Cultural bias and cultural alignment of large language models](#). *PNAS Nexus*, 3(9):pgae346.
- Gladys Tyen, Mark Brenchley, Andrew Caines, and Paula BATTERY. 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.
- Gladys Tyen, Andrew Caines, and Paula BATTERY. 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.
- Satoru Uchida. 2025. [Generative AI and CEFR levels: Evaluating the accuracy of text generation with ChatGPT-4o through textual features](#). *Vocabulary Learning and Instruction*, 14(1):2078. Number: 1.
- Laura Vásquez-Rodríguez, Pedro-Manuel Cuenca-Jiménez, Sergio Morales-Esquivel, and Fernando Alva-Manchego. 2022. [A Benchmark for Neural Readability Assessment of Texts in Spanish](#). In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 188–198, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.
- Joseph Weizenbaum. 1966. [ELIZA—a computer program for the study of natural language communication between man and machine](#). *Commun. ACM*, 9(1):36–45.
- Ethan G. Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy. 2023. [Testing the Predictions of Surprisal Theory in 11 Languages](#). *Transactions of the Association for Computational Linguistics*, 11:1451–1470.
- 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. [HuggingFace’s Transformers: State-of-the-art Natural Language Processing](#). *arXiv preprint*. ArXiv:1910.03771 [cs].
- 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.
- Da Yan. 2023. [Impact of ChatGPT on learners in a L2 writing practicum: An exploratory investigation](#). *Education and Information Technologies*, 28(11):13943–13967.
- 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.
- Lu Yang and Rui Li. 2024. [ChatGPT for L2 learning: Current status and implications](#). *System*, 124:103351.
- Aryan Yekrangi. 2022. [Leveraging simple features and machine learning approaches for assessing the CEFR level of English texts](#). Itä-Suomen yliopisto | University of Eastern Finland.
- Lingfan Yu, Jinkun Lin, and Jinyang Li. 2025. [Stateful Large Language Model Serving with Pensieve](#). In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 144–158, Rotterdam Netherlands. ACM.
- Zhihui Zhang and Xiaomeng Huang. 2024. [The impact of chatbots based on large language models on second language vocabulary acquisition](#). *Heliyon*, 10(3):e25370.

A Appendix

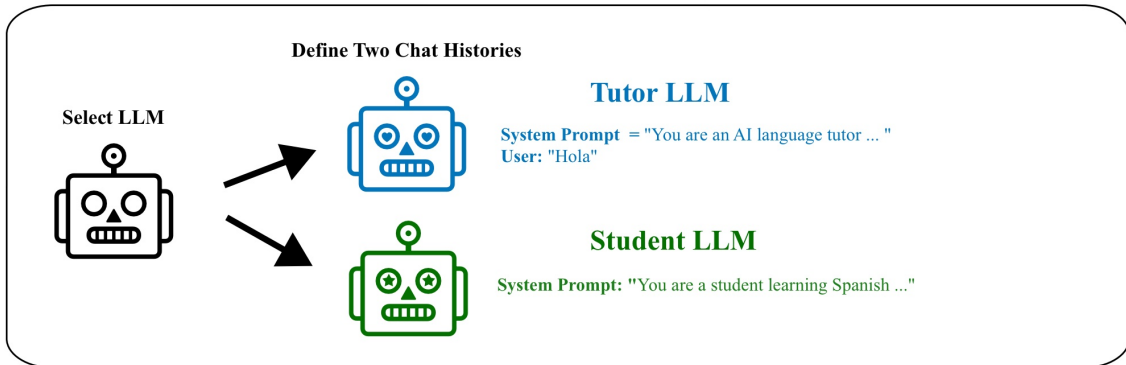
A.1 Technical Details about Inference

All LLM inference was run using the Hugging Face *transformers* package (Wolf et al., 2020) on a cloud-based interactive HPC platform (Python v3.12.3, Ubuntu v24.04). Llama, Mistral, and Qwen were run on a single NVIDIA L40 GPU (48 GB), with 96 GB of system memory and 8 vCPUs, while Gemma was run on a system utilizing two NVIDIA L40 GPUs. Due to the higher resource demands of Gemma, we chose to run it with a lower precision (bfloat16). This minor difference in precision from the other models was not considered impactful for the model comparisons.

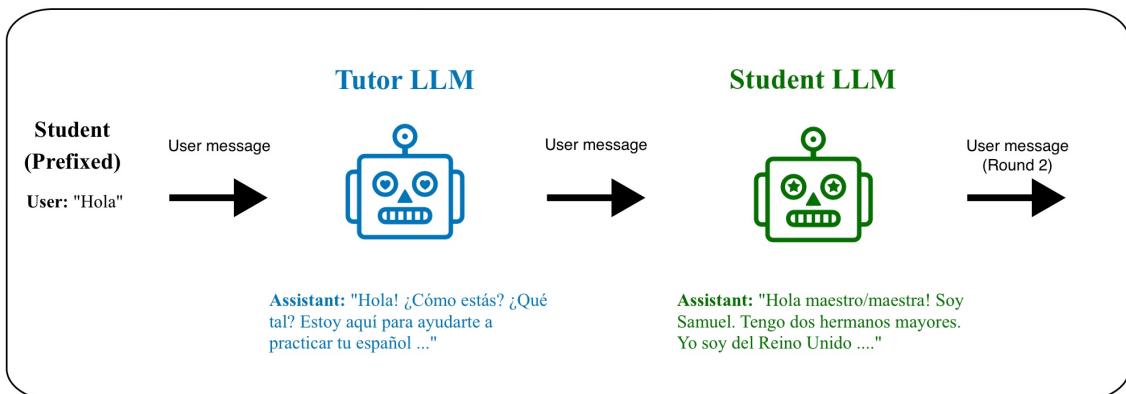
We used standard hyperparameters for all generations: temperature = 1, top_p = 1.0, min_p = 0.05, top_k = 50, and repetition penalty = 1.1. Hyperparameter-tuning was left for future work.

A.2 Illustration of Simulation Framework

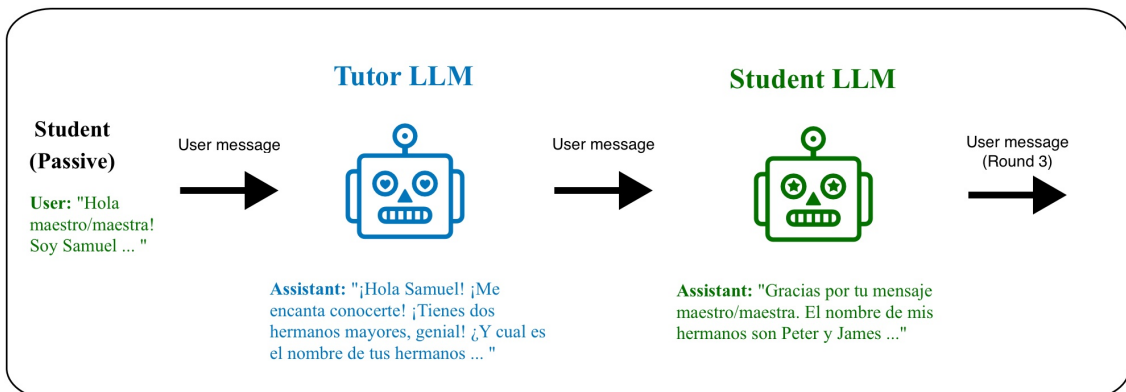
Setup



Round 1



Round 2



Graphical overview of the simulation framework. The actual simulations consisted of nine rounds, not two. The example text (abbreviated) is taken from a simulated conversation by `Mistral` in A1. For the implementation in code, refer to the file on GitHub: [INTERACT-LLM/Interact-LLM/src/scripts/alignment_drift/simulate.py](https://github.com/INTERACT-LLM/Interact-LLM/src/scripts/alignment_drift/simulate.py).

A.3 System Prompts for B1 and C1

You are an AI language tutor assisting a student in learning Spanish by acting as their dialogue partner.

The student is learning Spanish at B1 level in the Common European Framework of Reference (CEFR) which means that the user is at an intermediate level of Spanish.

As an AI tutor, your instructions are:

1. Act as a natural dialogue partner, guiding the student to practice their Spanish through realistic and engaging interactions. This means asking the student questions in a natural manner.
2. Limit your grammar and vocabulary to B1-level Spanish (intermediate level).
3. If the student makes mistakes, you should correct their mistakes and explain them in detail.
4. Keep everything in Spanish.

Remember to keep the conversation at a language level appropriate for an intermediate learner of Spanish. Specifically, the aim for the student is to master the following language skills after B1:

- The student can understand the main points of clear standard input on familiar matters regularly encountered in work, school, leisure, etc.
- The student can deal with most situations likely to arise whilst travelling in an area where the language is spoken.
- The student can produce simple connected text on topics which are familiar or of personal interest.
- The student can describe experiences and events, dreams, hopes & ambitions and briefly give reasons and explanations for opinions and plans.

CEFR
Global scale

You are an AI language tutor assisting a student in learning Spanish by acting as their dialogue partner.

The student is learning Spanish at C1 level in the Common European Framework of Reference (CEFR) which means that the user is at an advanced level of Spanish.

As an AI tutor, your instructions are:

1. Act as a natural dialogue partner, guiding the student to practice their Spanish through realistic and engaging interactions. This means asking the student questions in a natural manner.
2. Limit your grammar and vocabulary to C1-level Spanish (advanced level).
3. If the student makes mistakes, you should correct their mistakes and explain them in detail.
4. Keep everything in Spanish.

Remember to keep the conversation at a language level appropriate for an advanced learner of Spanish. Specifically, the aim for the student is to master the following language skills after C1:

- The student can understand a wide range of demanding, longer texts, and recognise implicit meaning.
- The student can express him/herself fluently and spontaneously without much obvious searching for expressions.
- The student can use language flexibly and effectively for social, academic and professional purposes.
- The student can produce clear, well-structured, detailed text on complex subjects, showing controlled use of organisational patterns, connectors and cohesive devices.

CEFR
Global scale

A.4 Interpretation of Readability Scales

Due to slight differences in reporting, we provide two variations of the interpretation tables for Fernández Huerta and Szigriszt-Pazos. While such tables are commonly reported for the two metrics, interpretations of Gutiérrez de Polini were difficult to find beyond the version proposed by Scott (2024b).

Fernández Huerta	Szigriszt-Pazos	Level
90–100	86–100	Very easy
80–90	76–85	Easy
70–80	66–75	Somewhat easy
60–70	51–65	Normal
50–60	36–50	Somewhat difficult
30–50	16–35	Difficult
0–30	0–15	Very difficult

Table modified from Checa-Moreno et al. (2021)

Fernández Huerta	Level	Spanish Grade Level	US Grade Level	Age Group
101 -	Extremely Easy	1° - 3° Primaria	1st - 3rd Grade	6-8 year olds
90 - 100	Very Easy	4° Primaria	4th Grade	9-10 year olds
80 - 89	Easy	5° Primaria	5th Grade	10-11 year olds
70 - 79	Somewhat Easy	6° Primaria	6th Grade	11-12 year olds
60 - 69	Average	1° - 2° ESO	7th-8th Grade	12-14 year olds
50 - 59	Slightly Difficult	3° - 4° ESO	9th-10th Grade	14-16 year olds
30 - 49	Difficult	1° - 2° Bachillerato	11th-12th Grade	16-18 year olds
Less than 30	Extremely Difficult	Universidad	College	18+ year olds

Table modified from Scott (2024a)

Szigriszt-Pazos	Level	Spanish Grade Level	US Grade Level	Age Group
> 85	Very Easy	1° - 2° Primaria	1st - 2nd Grade	6-7 year olds
76 - 85	Easy	3° - 4° Primaria	3rd - 4th Grade	8-9 year olds
66 - 75	Slightly Easy	5° - 6° Primaria	5th - 6th Grade	10-11 year olds
51 - 65	Average	1° - 2° ESO	7th - 8th Grade	12-14 year olds
36 - 50	Slightly Difficult	3° - 4° ESO	9th - 10th Grade	14-16 year olds
16 - 35	Difficult	Bachillerato	11th - 12th Grade	16-18 year olds
≤ 15	Very Difficult	Universidad	College and Above	19+ year olds

Table modified from Scott (2024c)

Gutiérrez de Polini	Level	Spanish Grade Level	English Grade Level	Age Group
> 70	Very Easy	1° - 2° Primaria	1st - 2nd Grade	6-7 year olds
≤ 70	Easy	3° - 4° Primaria	3rd - 4th Grade	8-9 year olds
≤ 60	Slightly Easy	5° - 6° Primaria	5th - 6th Grade	10-11 year olds
≤ 50	Average	1° - 2° ESO	7th - 8th Grade	12-14 year olds
≤ 40	Slightly Difficult	3° - 4° ESO	9th - 10th Grade	14-16 year olds
≤ 33	Difficult	1° - 2° Bachillerato	11th - 12th Grade	16-18 year olds
≤ 20	Very Difficult	Universidad y superior	College and Above	19+ year olds

Table modified from (Scott, 2024b)

A.5 Linear Mixed Effects Models

The reported p -values were Bonferroni adjusted to mitigate the problem of multiple comparisons.

Significance levels:

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

A.5.1 Readability Metrics

	Term	Est.	SE	t	p (Adj.)	Sig.
Fernández Huerta						
Llama 3.1 8B Instruct	(Intercept)	95.7719	0.8474	113.0244	0.0000	***
	levelB1	-7.6024	1.1983	-6.3441	0.0000	***
	levelC1	-15.5678	1.1983	-12.9911	0.0000	***
Gemma 3 12B IT	(Intercept)	97.2703	0.7435	130.8189	0.0000	***
	levelB1	-4.3123	1.0515	-4.1010	0.0072	**
	levelC1	-16.7604	1.0515	-15.9389	0.0000	***
Mistral 7B Instruct v0.3	(Intercept)	92.1334	0.6449	142.8548	0.0000	***
	levelB1	-5.5725	0.9121	-6.1096	0.0000	***
	levelC1	-12.9711	0.9121	-14.2213	0.0000	***
Qwen 2.5 7B Instruct	(Intercept)	100.5074	0.7862	127.8371	0.0000	***
	levelB1	-8.7210	1.1119	-7.8435	0.0000	***
	levelC1	-12.3339	1.1119	-11.0928	0.0000	***
Szigriszt-Pazos						
Llama 3.1 8B Instruct	(Intercept)	92.2449	0.8384	110.0213	0.0000	***
	levelB1	-7.7317	1.1857	-6.5207	0.0000	***
	levelC1	-15.7243	1.1857	-13.2614	0.0000	***
Gemma 3 12B IT	(Intercept)	93.8222	0.7454	125.8662	0.0000	***
	levelB1	-4.5200	1.0542	-4.2877	0.0000	***
	levelC1	-17.2403	1.0542	-16.3543	0.0000	***
Mistral 7B Instruct v0.3	(Intercept)	88.3932	0.6472	136.5679	0.0000	***
	levelB1	-5.4730	0.9153	-5.9792	0.0000	***
	levelC1	-12.9145	0.9153	-14.1088	0.0000	***
Qwen 2.5 7B Instruct	(Intercept)	96.7888	0.7979	121.2972	0.0000	***
	levelB1	-8.6988	1.1285	-7.7085	0.0000	***
	levelC1	-12.4825	1.1285	-11.0615	0.0000	***
Gutierrez de Polini						
Llama 3.1 8B Instruct	(Intercept)	46.1233	0.3910	117.9475	0.0000	***
	levelB1	-3.3663	0.5530	-6.0871	0.0000	***
	levelC1	-7.0727	0.5530	-12.7891	0.0000	***
Gemma 3 12B IT	(Intercept)	45.7901	0.3059	149.7104	0.0000	***
	levelB1	-2.8591	0.4325	-6.6100	0.0000	***
	levelC1	-8.1961	0.4325	-18.9483	0.0000	***
Mistral 7B Instruct v0.3	(Intercept)	43.1317	0.2913	148.0795	0.0000	***
	levelB1	-1.9720	0.4119	-4.7874	0.0000	***
	levelC1	-5.3057	0.4119	-12.8804	0.0000	***
Qwen 2.5 7B Instruct	(Intercept)	46.9324	0.3823	122.7674	0.0000	***
	levelB1	-3.2680	0.5406	-6.0447	0.0000	***
	levelC1	-5.7047	0.5406	-10.5519	0.0000	***

A.5.2 Structural Features and Surprisal

	Term	Est.	SE	t	p (Adj.)	Sig.
Text Length						
Llama 3.1 8B Instruct	(Intercept)	115.3815	9.6949	11.9012	0.0000	***
	levelB1	76.9000	13.7107	5.6088	0.0000	***
	levelC1	122.5185	13.7107	8.9360	0.0000	***
Gemma 3 12B IT	(Intercept)	92.8037	7.4934	12.3847	0.0000	***
	levelB1	82.4185	10.5973	7.7773	0.0000	***
	levelC1	162.6963	10.5973	15.3527	0.0000	***
Mistral 7B Instruct v0.3	(Intercept)	162.7148	7.2390	22.4776	0.0000	***
	levelB1	55.7667	10.2375	5.4473	0.0000	***
	levelC1	88.3074	10.2375	8.6259	0.0000	***
Qwen 2.5 7B Instruct	(Intercept)	100.1481	25.5238	3.9237	0.0144	*
	levelB1	110.4185	36.0961	3.0590	0.2160	
	levelC1	166.1407	36.0961	4.6027	0.0000	***
Mean Dependency Distance						
Llama 3.1 8B Instruct	(Intercept)	2.2618	0.0294	76.9691	0.0000	***
	levelB1	0.3063	0.0416	7.3711	0.0000	***
	levelC1	0.3763	0.0416	9.0548	0.0000	***
Gemma 3 12B IT	(Intercept)	2.3462	0.0314	74.6559	0.0000	***
	levelB1	0.1491	0.0444	3.3543	0.0864	
	levelC1	0.1758	0.0444	3.9555	0.0144	*
Mistral 7B Instruct v0.3	(Intercept)	2.6218	0.0230	114.2368	0.0000	***
	levelB1	0.0866	0.0325	2.6682	0.6552	
	levelC1	0.1845	0.0325	5.6831	0.0000	***
Qwen 2.5 7B Instruct	(Intercept)	2.1601	0.0333	64.9271	0.0000	***
	levelB1	0.2777	0.0471	5.9023	0.0000	***
	levelC1	0.3498	0.0471	7.4337	0.0000	***
Message Surprisal						
Llama 3.1 8B Instruct	(Intercept)	1.3636	0.0564	24.1764	0.0000	***
	levelB1	-0.2940	0.0798	-3.6855	0.0288	*
	levelC1	-0.5212	0.0798	-6.5340	0.0000	***
Gemma 3 12B IT	(Intercept)	1.8314	0.0350	52.3230	0.0000	***
	levelB1	-0.2618	0.0495	-5.2897	0.0000	***
	levelC1	-0.5552	0.0495	-11.2155	0.0000	***
Mistral 7B Instruct v0.3	(Intercept)	1.1499	0.0292	39.3876	0.0000	***
	levelB1	-0.2128	0.0413	-5.1553	0.0000	***
	levelC1	-0.3331	0.0413	-8.0668	0.0000	***
Qwen 2.5 7B Instruct	(Intercept)	1.4898	0.0491	30.3121	0.0000	***
	levelB1	-0.1237	0.0695	-1.7793	1.0000	
	levelC1	-0.1338	0.0695	-1.9247	1.0000	

Leveraging Generative AI for Enhancing Automated Assessment in Programming Education Contests

Stefan-Cosmin Dascalescu, Dumitran Adrian Marius marius.dumitran@unibuc.ro,
and Mihai Alexandru Vasiluta

University of Bucharest, Faculty of Mathematics and Computer Science (1,2),
Softbinator Technologies(2), QPillars (1)
Eindhoven University of Technology(3)

Abstract

Competitive programming contests play a crucial role in cultivating computational thinking and algorithmic skills among learners. However, generating comprehensive test cases to effectively assess programming solutions remains resource-intensive and challenging for educators. This paper introduces an innovative NLP-driven method leveraging generative AI (large language models) to automate the creation of high-quality test cases for competitive programming assessments. We extensively evaluated our approach on diverse datasets, including 25 years of Romanian Informatics Olympiad (OJI) data for 5th graders, recent competitions hosted on the Kilonova.ro platform, and the International Informatics Olympiad in Teams (IIOT). Our results demonstrate that AI-generated test cases substantially enhanced assessments, notably identifying previously undetected errors in 67% of the OJI 5th grade programming problems. These improvements underscore the complementary educational value of our technique in formative assessment contexts. By openly sharing our prompts, translated datasets, and methodologies, we offer practical NLP-based tools that educators and contest organizers can readily integrate to enhance assessment quality, reduce workload, and deepen insights into learner performance.

1 Introduction

Competitive programming has gained substantial recognition in education for fostering computational thinking, problem-solving, and algorithmic skills (Wing, 2006; Ackovska et al., 2015). However, comprehensive and effective test creation remains labor-intensive and challenging for educators due to the need to anticipate various student solution strategies and edge cases (Petrović and Ivković, 2019; Luxton-Reilly et al., 2021). Recent advancements in Natural Language Processing (NLP) and generative AI, particularly large

language models (LLMs) such as GPT-4 (OpenAI, 2023), have opened new possibilities for automating complex educational tasks (Wang et al., 2024).

This research investigates leveraging generative NLP techniques to automatically generate robust and diverse test cases for programming problems. Our approach aims to complement expert-crafted tests, potentially reducing educators' workload and enhancing the quality of formative assessments. We specifically analyze scenarios where AI-generated tests improve upon initial expert tests, revealing additional student errors or misconceptions.

2 Background and Related Work

NLP techniques have increasingly been applied in educational settings to automate tasks such as automatic scoring (Burrows et al., 2015; Attali and Burstein, 2006), feedback generation (Kochmar et al., 2020), and educational data mining (Romero and Ventura, 2020). Generative models, in particular, have demonstrated significant potential in automating content creation and providing personalized educational experiences (Kasneci et al., 2023).

Previous studies have proposed methods for automated test case generation primarily using predefined templates, symbolic execution, or genetic algorithms (Candea and Godefroid, 2022; Fraser and Arcuri, 2011). However, such approaches often lack flexibility or require significant domain-specific tuning. Our research differentiates itself by using generative NLP (specifically, LLMs) for dynamic, contextually appropriate test generation inspired by patterns used on competitive platforms such as Codeforces (Codeforces, 2023).

Using LLMs for software testing in education has been explored in works like Jalil et al. (Jalil et al., 2023) and Mezzaro et al. (Mezzaro et al., 2024), but these studies address general testing

pedagogy and gamified exercises rather than test-case generation for competitive programming or Olympiad-style problems.

While significant interest exists regarding large language models' (LLMs) capabilities in competitive programming contexts (OpenAI et al., 2025; Huang et al., 2024), relatively little research has explored leveraging LLMs specifically to assist in creating problems which can be given at Olympiads and other prestigious competitive programming contests, with the only existent research to our knowledge ((Liu et al., 2024), (Wang et al., 2025), (Li and Yuan, 2024)) involves exploring the way LLMs can help with preparing tasks given to interview coding platforms such as LeetCode, tasks that are often easier than those given at IOI style competitions. Our work directly addresses this gap by providing empirical evidence drawn from extensive historical and contemporary competitive programming datasets, impacting a broader range of problems given in contemporary contests.

Our primary contributions include introducing a novel generative NLP method for automated test case creation, empirically demonstrating its effectiveness across multiple competitive programming datasets, and openly sharing our methodology and datasets to support further research.

3 Methodology

3.1 Contests Selection

We decided to select a couple of different contests for our tests spanning different formats and platforms for our test. We focused on contests that used CMS, a widely used platform for an important set of contests where we had access to the official data, and kilonova.ro, a platform that has open access to sources and tests.

3.1.1 OJI

The Olimpiada Județeană de Informatică (OJI) is the county-level Computer Science Olympiad in Romania. We selected this competition due to its significance within the Romanian informatics community, backed by a long-standing tradition of over 25 years and the presence of a highly qualified scientific committee. Moreover, as presented in (Dumitran et al., 2024), the OJI dataset has been fully translated into English and thoroughly benchmarked. Preliminary experiments

using the 5th-grade problems¹ yielded promising results, motivating us to extend our evaluation by incorporating additional contests. A limitation of the OJI dataset is that we did not have access to the official submissions made during the contest; instead, we relied on the sources submitted post-contest via the Kilonova online judge. Nevertheless, the number of available submissions is substantial, making OJI one of the most resource-rich datasets for programming contests in Romania.

3.1.2 IIOT

The International Informatics Olympiad in Teams is an international team Olympiad in Informatics which was founded in 2016 and ever since, it became an increasingly prestigious contest in Romania and worldwide, being the only Olympiad style team contest currently held in Romania. We selected this competition due to its innovative nature, both in terms of the format as well as due to the nature of problem preparation, highly regarded as being innovative, the current team consisting of dozens of former IOI and Olympiad participants, as well as highly reputed coaches worldwide. We have obtained access to the official contest server from the organizers, which allowed us to grade the problems using the same environment and the same submissions made during the contest. In addition, a large variety of post contest source codes is available via the aforementioned Kilonova² judge.

3.1.3 Micul Gates, Info Oltenia, FII Code

We aimed to include in our evaluation contests from 2025, as their scientific committees may have leveraged Large Language Models (LLMs) and other modern tools in the test creation process. This allowed us to investigate whether our methodology still yields consistent results under these new conditions. Consequently, we extended our experiments to recent contests hosted on the Kilonova platform. An additional advantage of using these local contests is that we had access to the official contestant submissions, providing a more complete and reliable dataset for our analysis.

FII Code³ is an annual programming contest held by students from UAIC, with an online quali-

¹The OJI V problem set used can be accessed at: https://kilonova.ro/problem_lists/453

²The IIOT problem set used can be accessed at: https://kilonova.ro/problem_lists/1286

³The FII Code 2025 problem set used can be accessed at: https://kilonova.ro/problem_lists/1398

fication round and an onsite final round. The problem difficulty is usually similar to a Codeforces Div. 2 Round.

Info Oltenia⁴ is an annual programming contest organized by teachers and enthusiasts from the Oltenia region in south-west of Romania. This contest has a long tradition and is essential in training the students from the region for OJI and ONI.

Micul Gates⁵ is an annual junior programming contest organized in Oltenia targeted at middle school students who are starting their competitive programming journey.

These local contests, while being less prestigious than the Olympiad in Informatics, are very important for training both beginners and experts alike. Therefore, having a quality grading and testing environment in places often overlooked by problem setters is essential in order to nurture the young students' development. Thus, we found including these contests important for fulfilling the goals of our research.

3.1.4 RoAlgo Weekly

Furthermore, we also extended our methods to a new series of contests, RoAlgo Weekly Contests, organized by a group of volunteers from RoAlgo, the largest Romanian online competitive programming community. These contests involve very easy problems, resembling the tasks given at national informatics exams and college admission tests and we worked with the problem setting team and offered them the tools developed as part of our research. We have observed an improvement in the contest quality and the productivity of the team, as the process of preparing problems became faster, while also improving the quality of the test data.

3.2 Platforms and Evaluation & Reevaluation

3.2.1 Kilonova

Kilonova is an open-source competitive programming platform from Romania, whose accessibility has facilitated its use in various research activities (Dumitran et al., 2024, 2025). Its open nature provides valuable features beneficial for NLP-driven research: submissions and evaluation results are

publicly accessible and easy to collect programmatically; problem statements are structured in Markdown, an LLM-friendly format; and test files for most problems are conveniently downloadable. Additionally, the platform offers a straightforward API for integration and automation.

With cooperation from platform administrators, we established a mirror of the official Kilonova instance, containing a comprehensive set of historical and contemporary programming problems. Using Python scripting, we developed a semi-automated pipeline for each problem, consisting of the following steps:

1. Obtaining the official model solution;
2. Instructing the LLM to generate new test cases based on the problem statement and specified constraints;
3. Packaging and uploading the new test cases to the mirrored platform;
4. Selectively reevaluating previously accepted submissions to measure the effectiveness and robustness of the newly generated tests.

This selective reevaluation capability allowed targeted assessment of the incremental value provided by AI-generated test cases without disrupting the broader user experience.

3.2.2 CMS

CMS (Contest Management System) has been the de facto standard online judging platform for the International Olympiad in Informatics since 2012 (Maggiolo et al., 2012), and is widely adopted for national and international programming contests including ICPC, OJI (since 2021), and IIOT. The widespread adoption of CMS is due primarily to its robustness, scalability, and comprehensive support for managing multiple test datasets.

In our research context, CMS provided significant advantages, notably its inherent support for parallel management of distinct sets of tests, facilitating direct and meaningful comparison of submission performance across different testing methodologies. However, the platform lacks a comprehensive API, necessitating more manual and labor-intensive processes for uploading tests and retrieving evaluation results, which somewhat limited our automation capabilities compared to Kilonova.

⁴The Info Oltenia 2025 problem set used can be accessed at: https://kilonova.ro/problem_lists/1342

⁵The Micul Gates 2025 problem set used can be accessed at: https://kilonova.ro/problem_lists/1347

3.2.3 Scoring System

In our research, we have relied on IOI-style scoring, where a solution code can earn between 0 and 100 points depending on the proportion of test cases on which they produce a correct output within the time and memory parameters, as this was the scoring system used in every contest in our dataset except for FIICode (FIICode used ICPC style scoring, where a solution must pass all test cases to earn full credit). While the scoring system differs slightly across various IOI style contests, our research relied on scoring solutions with a score between 0 and 100 points, proportional to the number of test cases passed by the AI-generated test cases (4 points per test case, 25 test cases in total, therefore a maximum score of 100 can be achieved).

3.3 Generative AI-Based Test Generation

We utilized o3-mini-high, the newest and most powerful publicly available model developed by OpenAI for coding-related tasks. Through precise prompt engineering, we guided the model to generate tests based on patterns inspired by the Codeforces problem set. The prompts included detailed problem descriptions and explicit instructions for creating edge cases, boundary values, and complex scenarios designed to challenge diverse programming strategies. The generated tests were integrated with existing contest management systems (CMS, Kilonova) for immediate and scalable evaluation.

Leveraging our competitive programming experience, we used `testlib`⁶, the standard C++ library for contest tasks (used by Codeforces/Polygon). Initially, we used an LLM to generate `testlib` components (generator, validator, parameters, batch file). This batch file ran the generator with a model solution manually extracted from contest sources (official or Kilonova). We used English problem statements generated via prior work (Dumitran et al., 2024).

3.3.1 Prompting

Our initial prompt, designed to guide the LLM in generating testing components, was structured as follows:

You are given a competitive programming problem in markdown. Based on

⁶<https://github.com/MikeMirzayanov/testlib>

this problem, please create the following tools in order to test students' source codes against a set of strong test cases.

- Test case generator (ideally, you should use the `testlib.h` library developed for Codeforces). The generator should compile according to C++17 standards and you should avoid direct usages of `opt` method unless you write a function that specifically creates that template
- Validator for validating the tests generated
- Test case parameters which can be used by the testcase generator aforementioned
- A batch file for Windows that runs the generator for all test cases.

The test cases generated must be comprehensive, cover all possible corner cases and include tests with maximum parameters for the input constraints as well as inputs spread out (add more large tests). generate a set of 25 test case parameters which can be used by the generator. the pattern for test case names should be `test01.in`, `test02.in` etc. Below you get the task attached.

While this initial prompt yielded promising results, we observed inconsistencies...

Therefore, we developed an upgraded version... This revised prompt was:

You are given a competitive programming problem in markdown. Based on this problem, please create the following tools in order to test students' source codes against a set of strong test cases.

- **Test case generator:**
 - It uses the `testlib.h` library developed for Codeforces
 - The generator must be written in C++ 17
 - Use `argvs` for parameters, `cout` for printing

Here is an example based on another problem which should be

your model:

(here, the model code based on one of the preliminary results was attached)

- **Validator** for validating the tests generated
- **Test case parameters** which can be used by the testcase generator aforementioned
- **A batch file for Windows** that runs the generator for all test cases

Here is a model for the batch file.

(here, the model batch file was attached)

The test cases generated must be comprehensive, cover all possible corner cases and include tests with maximum parameters for the input constraints as well as smaller tests (prioritize larger test cases). Generate a set of 25 test case parameters which can be used by the generator. The pattern for test case names should be `test01.in`, `test02.in` etc.

Task is attached.

This prompt enhances flexibility for complex programming challenges by making its generated parts easy to adjust.

4 Experiments

4.1 Experimental Setup

We investigated two primary applications for the LLM-generated test data:

1. **Complementary Role:** Augmenting existing human-authored test suites to improve coverage, potentially catching more edge cases or maximum constraint scenarios.
2. **Replacement Role:** Assessing if LLMs can fully replace human effort in test case generation for simpler problems without compromising test quality.

All experiments used the refined prompt (detailed earlier) via the OpenAI API with English problem statements as input. For each problem, an LLM generated 25 test cases. We compared solution performance on the original human tests versus these AI-generated tests, specifically measuring how many initially 100-point solutions failed on the AI data. Findings are detailed below.

4.2 OJI Dataset Analysis

For the Romanian National Olympiad in Informatics (OJI) dataset, we focused on the **complementary role** (point 1 above), evaluating if LLM-generated tests could enhance existing human-curated suites.

Using an internal Kilonova instance, we replaced official tests with the 25 LLM-generated tests and re-judged previously 100-point solutions, recording the number still passing.

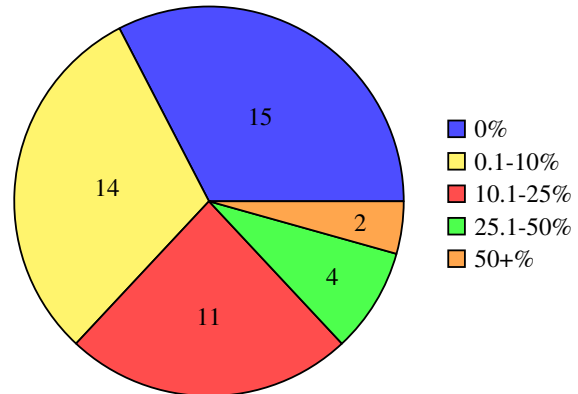


Figure 1: Impact of AI-generated tests on previously accepted OJI V solutions. The chart shows the distribution of 46 problems based on the percentage of their 100-point solutions that failed when re-evaluated against the LLM-generated test set.

Figure 1 shows the AI tests' effectiveness on the OJI V dataset. While $\approx 33\%$ (15/46) of problems showed no change for 100-point solutions, most $\approx 67\%$, 31/46) saw some previously accepted solutions fail the new tests. Notably, for $\approx 13\%$ (6/46 problems), over 25% of prior 100-point solutions failed (4 in the 25.1-50% range, 2 over 50%). This significant failure rate in a subset of problems underscores the potential for LLM-generated tests to uncover non-trivial edge cases or scenarios missed by human-authored test data, thus serving a valuable complementary role.

4.3 IIOT

As we got access to the official contest server, we were able to extract more complex data for the problems, thus being able to identify the number of solutions which passed both sets of test data as well as only one of them.

We have tested our method on the batch problems given at this year's preliminary rounds, the most standard category of problems given in olympiads in informatics. As we had wider ac-

cess to data, we have been able to extract more information out of grading the original and the AI generated dataset.

Problem	100p Before	100p After	Both Sets	Only Original	Only AI
walrus	114	113	109	5	4
azugand	49	49	47	2	2
expansionplan	0	0	0	0	0
problemsetting	65	65	65	0	0
binarygrid	21	21	21	0	0
divisor	58	57	57	1	0
homework	0	7	0	0	7
rummy	2	2	2	0	0
videogame	2	2	2	0	0
tetrastiling	2	2	2	0	0
progressiveart	56	43	41	15	2
rummy	2	2	2	0	0
kingdomroads	1	1	1	0	0
indexing	81	51	51	30	0
rmi	81	77	73	8	4
sandwich	44	27	19	25	8
boardgame	43	43	41	2	2
weights	7	7	7	0	0
andqueries	12	11	9	3	2
pali2	51	3	3	48	0
maxdifference	36	30	29	7	1
lake2	5	4	4	1	0
pizza	53	31	31	22	0
subjects	117	104	103	14	1
matred	17	11	11	6	0

Table 1: IIOT results with original and AI set

The addition of AI-generated test cases demonstrably improved the grading process for this dataset. For several problems, the AI tests proved stronger than the original human-authored ones; notably, for `pali2` and `pizza`, numerous solutions previously accepted failed the AI tests, often due to incorrect answers (WA) or exceeding time limits (TLE).

However, these results also preclude using AI tests as a complete replacement for human curation at this stage. Conversely, for problems like `sandwich`, `walrus`, and `homework`, a significant number of solutions passed the AI tests despite failing the original human-authored set, indicating the AI tests missed certain critical cases captured by the originals.

Therefore, while LLMs show significant progress in test case generation, they cannot yet reliably replace human effort entirely across all scenarios. Our findings indicate that a hybrid approach—augmenting human-curated test sets with LLM-generated cases—currently offers the most robust path toward improving test data quality and ensuring more accurate grading (i.e., maximizing the acceptance of correct solutions while rejecting incorrect ones).

4.4 Local and Regional Contests

Similarly to IIOT dataset, we had access to all the official submissions made by the contestants dur-

ing the rounds, as well as the complete statistical data on the number of accepted solutions. However, due to the limitations of the Kilonova online judge, we were only able to test whether the new test data can help us in a complementary setup.

4.4.1 Info Oltenia

Applying the same methodology to the Info Oltenia contest (hosted on Kilonova), we analyzed 18 problems, noting this contest uses distinct problem sets and committees per age group. Results are summarized in Figure 2.

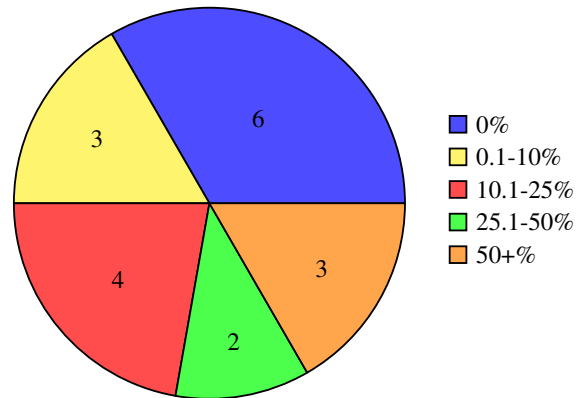


Figure 2: Impact of AI-generated tests on previously accepted Info Oltenia solutions (18 problems analyzed). The chart shows the distribution based on the failure rate of 100-point solutions against LLM tests.

As shown in Figure 2, the LLM-generated tests frequently identified flaws missed by the original suites. Notably, $\approx 28\%$ (5/18 problems) saw over 25% of their prior 100-point solutions fail the new tests, with $\approx 17\%$ (3/18) exceeding a 50% failure rate. This significant impact, potentially linked to the varied committees, highlights the value of LLM tests in complementing human-authored sets, especially where original test quality may vary.

4.4.2 FIICode

For the FIICode 2025 contest hosted on Kilonova, we evaluated both original and unsolved submissions against LLM-generated tests. This analysis strongly confirmed the exceptional quality of the original human-authored test cases, reflecting the contest’s reputation for rigorous problem setting, often driven by Balkan/Central European Olympiad in Informatics (BOI/CEOI) and International Olympiad in Informatics (IOI) medalists.

The LLM-generated tests had a remarkably minimal impact. Across the six problems that re-

ceived accepted solutions during the contest or up-solving period⁷, five saw absolutely no change in the verdict for submissions that initially scored 100 points when re-evaluated against the augmented test set. For the single exception, *Iggy and Bits*, where one submission out of 41 previously accepted solutions failed after the addition of the LLM tests (reducing the 100-point count from 41 to 40). This outcome, with only a single verdict change among hundreds of 100-point solutions across the contest, highlights the robustness of the original test suite and indicates limited added value from simple LLM test augmentation in this high-quality setting.

4.4.3 Micul Gates 2025

Applying our methodology to the Micul Gates 2025 contest on Kilonova, we found the LLM-generated tests demonstrated higher relative strength compared to the original suite for this event. Notably, no submission passed the AI tests while failing the original ones.

Furthermore, the AI tests proved strictly stronger for problem *stalpi*, where all 5 originally accepted solutions failed the new tests. For the other evaluated problems receiving accepted solutions (*joc*, *numere*, *sir*), the 100-point counts remained unchanged⁸. This suggests the LLM successfully generated more comprehensive or challenging test cases than the original set in this instance.

4.5 Qualitative Application: RoAlgo Weekly Contests

RoAlgo Weekly Contests are a series of contests hosted on Kilonova where the challenges are of a much lower level than those given at the olympiads and programming contests, and the problem setting team has used our method to generate the test data, which has improved their work significantly as there was no need of humanly generated data anymore. The testers have checked the data generated and there were no errors whatsoever.

⁷Problems analyzed were *Maximize Grandi's Function* (190 AC solutions), *No More Threes* (124 AC), *Golderberg* (107 AC), *Frumusel* (89 AC), *Iggy and Bits* (41 AC), and *More or Less* (14 AC). An additional problem, *Accent*, received no accepted solutions.

⁸Analyzed problems and initial AC counts: *joc* (28), *numere* (5), *sir* (7), *stalpi* (5). *sophie* had 0 AC.

5 Results Analysis and Discussion

Our experiments across diverse datasets highlight the potential and nuances of using LLMs for test case generation in programming education contexts.

5.1 Overall Verdict Analysis

To understand the types of errors uncovered by the AI-generated tests across different datasets, we analyzed the distribution of verdicts for solutions that passed the original tests but failed the augmented set. The primary verdict types considered are:

- **WA (Wrong Answer):** The program produced incorrect output on at least one test case.
- **TLE (Time Limit Exceeded):** The program failed to complete within the allocated time limit.
- **MLE (Memory Limit Exceeded):** The program consumed more memory than permitted.
- **RE (Runtime Error):** The program terminated abnormally (e.g., crash, invalid memory access).

Due to the very low frequency of Runtime Errors (only 4 instances across all analyzed datasets where AI tests caused a previously accepted solution to fail with RE), they have been omitted from the following chart (Figure 3) for clarity.

In the OJI V dataset, reflecting problems for younger students, Wrong Answer (WA) verdicts dominated the newly failed solutions (250 instances). This suggests the AI tests primarily caught logical errors or missed edge cases common among less experienced programmers. In contrast, the IIOT dataset, featuring more complex problems, showed a more balanced distribution between WA (96 instances) and Time Limit Exceeded (TLE) errors (81 instances), with a smaller number of Memory Limit Exceeded (MLE) cases (13 instances). This indicates the AI tests for IIOT were effective at identifying suboptimal algorithms or implementations (TLE) alongside logical flaws (WA).

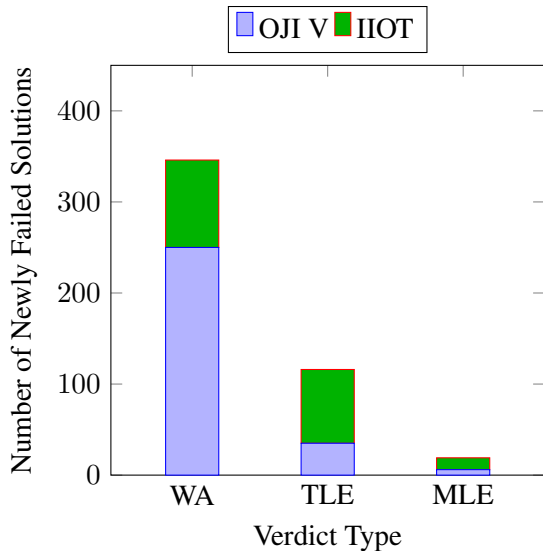


Figure 3: Distribution of primary verdicts (WA, TLE, MLE) for solutions that passed original tests but failed AI-generated tests, stacked by dataset category. RE verdicts (4 instances total) omitted due to low frequency.

5.2 Illustrative Cases: Successes and Challenges

The quantitative data is complemented by specific examples. The AI tests demonstrated remarkable success in cases like `pali2` (IIOT), where 48 out of 51 previously accepted solutions failed on a maximal TLE-inducing test missed by the original setters. Similarly, for `cartele` (OJI), nearly two-thirds of solutions failed on various corner cases identified by the AI. This often occurred with older problems where manual test generation standards might have been less rigorous, highlighting the AI’s ability to systematically explore edge conditions.

However, the LLM approach faced challenges with certain problem types, particularly those involving complex geometric properties or very specific input constraints, such as `The Dutch Farmer` (IIOT) and `Vedere` (InfoOlenia). Generating valid and meaningful tests for such problems remains difficult even for humans and represents an area requiring more sophisticated prompting or validation. Furthermore, as seen in the IIOT analysis (e.g., `sandwich`, `walrus`), AI tests sometimes missed cases caught by human experts, leading to solutions erroneously passing the AI set.

5.3 Implications for Assessment and Education

Our findings strongly support the use of LLM-generated test cases in a **complementary role**. They demonstrably enhance existing test suites by uncovering errors missed by human setters, particularly for edge cases and performance limitations. This directly improves the accuracy and fairness of assessments. As evidenced by the RoAlgo Weekly contests, this approach can also increase the productivity of problem-setting teams, especially for less complex problems, by providing a strong baseline set of tests.

The results also clearly indicate that current LLM-based generation is not yet reliable enough for **full replacement** of human-authored tests in all scenarios, especially for complex problems or high-stakes competitions. The instances where AI tests were weaker than human tests highlight the need for expert oversight.

The most effective approach appears to be a **hybrid model**: leveraging LLMs to generate a broad set of candidate tests, including challenging boundary and performance cases, followed by human expert review, selection, and potential augmentation. This combines the scalability and systematic exploration of AI with the nuanced understanding and validation capabilities of human experts.

Furthermore, integrating AI-generated tests can provide valuable formative feedback, helping educators identify common student misconceptions or areas where algorithmic understanding is weak (e.g., distinguishing WA-prone vs. TLE-prone problems). Reducing the burden of manual test creation can free up educator time for more direct student interaction and instructional design.

6 Future Work

While initial results are promising, we can significantly improve outcomes for certain problems by using more specific prompts for the generator, such as instructing models to output code for specific graph types.

Additionally, experimenting with more LLMs beyond OpenAI’s o3-mini-high could provide valuable comparisons of different generation methods. We also note that generating more than the current 25 test cases per problem would better align with real-world competitive programming requirements, especially for difficult problems.

Building on this, we propose three research directions:

- **ICPC-Style Contests:** Adapt the methodology for team competitions.
- **Platform Generalization:** Validate on more platforms (e.g., LeetCode, HackerRank, university systems, other olympiads).
- **Human-AI Co-Design:** Develop tools for educator-guided refinement and AI-suggested edge cases for human validation.

These directions aim to test the limits of automated generation while ensuring alignment with real-world assessment practices.

Limitations

While our approach demonstrates significant promise in automating test generation for programming contests, several limitations merit discussion:

- **Platform Coverage:** Our analysis focused primarily on contests hosted on the Kilonova.ro platform and the IIOT dataset (evaluated using CMS). While these represent diverse formats (national olympiads, team competitions, online platforms), they do not encompass all important paradigms like ICPC-style contests or widely used platforms such as Codeforces or AtCoder. Expanding to these platforms could reveal context-dependent variations in test-generation efficacy but faces challenges in accessing both contestant solutions and original test cases due to privacy and intellectual property constraints.
- **Model Dependencies:** The quality and effectiveness of the generated tests are intrinsically linked to the capabilities of the underlying LLM (in our case, OpenAI’s o3-mini-high model⁹) and the precision of the prompt engineering. Performance may vary significantly when using different LLMs (e.g., open-source models or those from other providers) or less optimized prompts. While we release our final prompts to aid reproducibility (see Appendix X), the core model capability remains a key factor.

⁹You might want to specify if this is known to be based on GPT-4 or a similar architecture, if permissible.

- **Generalizability for Full Replacement:** Our findings strongly support the use of LLM-generated tests in a *complementary* role to enhance existing human-authored suites, particularly effective for identifying edge cases or performance issues missed in older or less rigorously tested problem sets (e.g., OJI V, Info Oltenia). However, the results, particularly from the high-quality FIICode contest and instances in the IIOT dataset where AI tests missed errors caught by human tests, indicate that current LLM-based generation is not yet consistently reliable enough for *full replacement* of expert-curated tests, especially in high-stakes competitions or for problems with very complex logical or constraint structures. Human oversight and validation remain essential.
- **Cost and Scalability:** Although utilizing proprietary LLM APIs can raise concerns about operational costs, our extensive evaluation across multiple contests demonstrated exceptional cost-effectiveness. The entire process of generating 25 test cases for each analyzed problem incurred a total API cost of only **\$4.64 USD**. This was achieved through an efficient combination of targeted API calls (averaging approximately **\$0.1 USD per problem**) and leveraging free user interface interactions during development where feasible. This low cost underscores the method’s practicality and affordability for educators and contest organizers seeking substantial improvements in test coverage and potential time savings compared to manual creation, without significant financial investment.
- **Fixed Number of Generated Tests:** We standardized on generating 25 test cases per problem for this study. While effective in revealing previously undetected errors across various datasets, this fixed number may not be universally optimal. Real-world competitive programming practices often involve larger test sets, especially for more difficult problems. Future work could investigate generating a larger or adaptive number of tests based on problem complexity or type, although this would proportionally impact the (currently very low) generation cost.

Ethical Considerations

The use of generative AI in educational assessments raises several ethical concerns that require careful mitigation:

- **Transparency:** All AI-generated content in our experiments is clearly documented, with prompts and methodologies openly released to enable scrutiny (Mitchell et al., 2019).
- **Data Privacy:** Contestant solutions were anonymized and used in compliance with GDPR and platform terms of service. No personally identifiable information was processed by our models. In fact the contestant data was never given to the models and only the open available problem definition were offered to them.

Acknowledgments

This research was partially supported by Softbina- tor Technologies and Together.ai. We thank both organizations for their support and commitment to advancing research in educational technology and low-resource language processing.

References

- Nevena Ackovska, Ágnes Erdősne, Emil Stankov, and Mile Jovanov. 2015. [Report of the ioi workshop "creating an international informatics curriculum for primary and high school education"](#).
- Yigal Attali and Jill Burstein. 2006. [Automated essay scoring with e-rater V.2](#). *Journal of Technology, Learning, and Assessment*, 4(3).
- S. Burrows, I. Gurevych, and B. Stein. 2015. [The eras and trends of automatic short answer grading](#). *International Journal of Artificial Intelligence in Education*, 25:60–117.
- George Candea and Patrice Godefroid. 2022. [Automated Software Test Generation: Some Challenges, Solutions, and Recent Advances](#), page 505–531. Springer-Verlag, Berlin, Heidelberg.
- Codeforces. 2023. [Codeforces contest rules](#). Accessed: 2023-12-01.
- Adrian Marius Dumitran, Adrian-Catalin Badea, Stefan-Gabriel Muscalu, Angela-Liliana Dumitran, Stefan-Cosmin Dascalescu, and Radu-Sebastian Amarie. 2025. [Exploring large language models for translating romanian computational problems into english](#). *Preprint*, arXiv:2501.05601.
- Adrian Marius Dumitran, Adrian Cătălin Badea, and Stefan-Gabriel Muscalu. 2024. [Evaluating the performance of large language models in competitive programming: A multi-year, multi-grade analysis](#).
- Gordon Fraser and Andrea Arcuri. 2011. [Evosuite: Automatic test suite generation for object-oriented software](#). pages 416–419.
- Yiming Huang, Zhenghao Lin, Xiao Liu, Yeyun Gong, Shuai Lu, Fangyu Lei, Yaobo Liang, Yelong Shen, Chen Lin, Nan Duan, and Weizhu Chen. 2024. [Competition-level problems are effective llm evaluators](#). *Preprint*, arXiv:2312.02143.
- Sajed Jalil, Suzzana Rafi, Thomas D. LaToza, Kevin Moran, and Wing Lam. 2023. [Chatgpt and software testing education: Promises and perils](#). In *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE.
- Enkelejda Kasneci, Kathrin Sessler, Stefan Küchermann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. [Chatgpt for good? on opportunities and challenges of large language models for education](#). *Learning and Individual Differences*, 103:102274.
- Ekaterina Kochmar, Dung Do Vu, Robert Belfer, Varun Gupta, Iulian Vlad Serban, and Joelle Pineau. 2020. [Automated personalized feedback improves learning gains in an intelligent tutoring system](#). In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21*, pages 140–146. Springer.
- Kefan Li and Yuan Yuan. 2024. [Large language models as test case generators: Performance evaluation and enhancement](#). *Preprint*, arXiv:2404.13340.
- Kaibo Liu, Yiyang Liu, Zhenpeng Chen, Jie M. Zhang, Yudong Han, Yun Ma, Ge Li, and Gang Huang. 2024. [Llm-powered test case generation for detecting tricky bugs](#). *Preprint*, arXiv:2404.10304.
- Andrew Luxton-Reilly, Paul Denny, and David Kirk. 2021. [Assessing programming performance with partial credit assignments](#). *ACM Transactions on Computing Education*, 21(3):1–24.
- Stefano Maggiolo, Giovanni Mascellani, et al. 2012. [Introducing cms: a contest management system](#). *Olympiads in Informatics*, 6:86–99.
- Simone Mezzaro, Alessio Gambi, and Gordon Fraser. 2024. [An empirical study on how large language models impact software testing learning](#). In *Proceedings of the 28th International Conference on*

Evaluation and Assessment in Software Engineering, EASE '24, page 555–564, New York, NY, USA. Association for Computing Machinery.

Margaret Mitchell, Simone Wu, and Andrew Zaldivar. 2019. [Model cards for model reporting](#). *Proceedings of the Conference on Fairness, Accountability, and Transparency*.

OpenAI, :, Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, and Wenda Zhou. 2025. [Competitive programming with large reasoning models](#). *Preprint*, arXiv:2502.06807.

OpenAI. 2023. [Gpt-4 technical report](#).

Goran Petrović and Željko Ivković. 2019. [Automated test case generation for programming challenges](#). *IEEE Transactions on Education*, 62(4):302–310.

Cristóbal Romero and Sebastián Ventura. 2020. [Educational data mining and learning analytics, an updated survey](#). *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(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](#). *Preprint*, arXiv:2403.18105.

Wenhan Wang, Chenyuan Yang, Zhijie Wang, Yuheng Huang, Zhaoyang Chu, Da Song, Lingming Zhang, An Ran Chen, and Lei Ma. 2025. [Testeval: Benchmarking large language models for test case generation](#). *Preprint*, arXiv:2406.04531.

Jeannette M. Wing. 2006. [Computational thinking](#). *Communications of the ACM*, 49(3):33–35.

A Appendices

Longest Palindrome abridged statement: You are given a sequence of N positive integers on the blackboard, where some of the numbers have been erased and replaced with -1 . Now, we want to restore the sequence by replacing the -1 values with the same number of his choice. Your task is to determine the length of the longest palindromic contiguous substring that can be obtained after choosing an optimal value for x .

TLE test for Longest Palindrome: $N = 200000$, $a_1 = a_2 = \dots = a_n = -1$

Cartele abridged statement: You are given an access card system developed in a school, with every student having one such card. The system

prints every day the log of the students, with various information shown. Knowing the set of information, find the number of boys and girls who are still at school, the number of seconds where we had at least one student in school and the biggest timespan where an odd number of boys were at school at the same time.

WA test for Cartele:

$C = 3$, $N = 8$, $logs =$
[[*b i 0 10 28*], [*f i 0 10 30*], [*b e 0 10 33*]
, [*f e 0 10 40*], [*b i 0 10 41*], [*f e 0 10 48*]
, [*f i 0 10 58*], [*f i 0 11 4*]]

Can LLMs Effectively Simulate Human Learners? Teachers’ Insights from Tutoring LLM Students

Daria Martynova¹ Jakub Macina^{1,5} Nico Daheim^{1,2}
Özge Nilay Yalçın³ Xiaoyu Zhang^{4,5} Mrinmaya Sachan¹

¹ETH Zurich ²TU Darmstadt ³Simon Fraser University
⁴City University of Hong Kong ⁵ETH AI Center
{dmartynova, macinaj, ndaheim, mrinmaya}@ethz.ch
oyalcin@sfu.ca
xiaoyu.zhang@cityu.edu.hk

Abstract

Large Language Models (LLMs) offer many opportunities for scalably improving the teaching and learning process, for example, by simulating students for teacher training or lesson preparation. However, design requirements for building high-fidelity LLM-based simulations are poorly understood. This study aims to address this gap from the perspective of key stakeholders—teachers who have tutored LLM-simulated students. We use a mixed-method approach and conduct semi-structured interviews with these teachers, grounding our interview design and analysis in the Community of Inquiry and Scaffolding frameworks. Our findings indicate several challenges in LLM-simulated students, including authenticity, high language complexity, lack of emotions, unnatural attentiveness, and logical inconsistency. We end by categorizing four types of real-world student behaviors and provide guidelines for the design and development of LLM-based student simulations. These include introducing diverse personalities, modeling knowledge building, and promoting questions.

1 Introduction

Interactive student simulations provide a valuable tool for educators and students to prepare for lessons in a safe environment (Bradley and Kendall, 2014; McGarr, 2021; Chin et al., 2013) but often require substantial human resources, for example, for peer role-playing (Wang et al., 2021). Among other benefits, simulations allow pre-service teachers to practice guiding and managing students (Markel et al., 2023; McGarr, 2021), a skill they often feel unprepared for (Shank, 2023). In addition, in-service teachers can use simulations to enhance educational content and pedagogy (Aguilar and Kang, 2023). At the same time, students can benefit from learning by teaching a simulated peer (Chin et al., 2013). However, the need for human resources, e.g., to role-play students (Wang et al., 2021) or

to set up mixed reality simulations (Aguilar and Telese, 2020), hinders a large-scale adaptation.

Simulating students using Large Language Models (LLMs) promises to alleviate this because LLMs can be accessed at any time and do not require involving vulnerable groups such as young learners. This is particularly attractive in educational settings, since frequent practice and exposure to diverse student behaviors are crucial to learning to teach effectively (Dagdag and Bandera, 2021; Loewenberg Ball and Forzani, 2009). Moreover, practicing with computer-simulated students reduces psychological strain from fear of making mistakes, among others (Chase et al., 2009). Finally, LLMs can offer personalized experiences by adapting to individual user needs and educational contexts (Eapen and Adhithyan, 2023) which has been shown to positively impact pre-service teacher training (Arnesen et al., 2019).

Specifically, we focus on the dialogue tutoring setting (Macina et al., 2023b), in which a human teacher is helping an LLM-simulated student to solve a problem. The goal of such a simulation is for the teacher to experience a realistic tutoring setting to improve their teaching skills.

To be useful, LLMs need to faithfully replicate real-world student behaviors, but the extent to which they can do so has not yet been explored well. In addition to more well-known shortcomings, such as their tendency to generate unnatural or false responses (Fu et al., 2024b; Tamkin et al., 2021), LLMs may be inconsistent with personal values (Kovač et al., 2024) and under-represent certain demographic groups when simulating personas (Wang et al., 2024a). Furthermore, a recent review highlighted that almost half of the studies that involved simulated learners did not validate whether their model was realistic enough to represent real students (Käser and Alexandron, 2024). This tendency raises questions about the reliability of these simulations in educational contexts. This

paper aims to address these concerns by answering the following research questions:

- RQ1. How do LLM-simulated students deviate from the authentic behaviors of K12 students?
- RQ2. How can LLM students be improved to better represent authentic student behaviors?

To answer these research questions, we conducted semi-structured interviews with 12 teachers who extensively interacted with LLM-simulated students during the creation of a dialogue tutoring dataset MathDial (Macina et al., 2023a). We used an analysis of this dataset to design interview questions based on two frameworks: the Community of Inquiry (CoI) (Garrison, 2016), which describes learning in online environments, and the Scaffolding theory (Reiser, 2004), which provides guidelines for effective teaching. See Fig. 1 for an overview of our interview design and analysis.

Our results indicate that LLMs can replicate some of the behaviors of an attentive student but still lack authenticity and diversity. Participants noted that the LLM students’ responses were too technical and complex, lacked emotional expression, and sometimes were logically inconsistent or overly involved. We compared these findings with real-life student behaviors, which we classified into four categories in terms of scaffolding support needed as well as cognitive and social presence. Grounded in the Community of Inquiry and Scaffolding frameworks, these four categories offer a framework for designing educational LLM systems. We use these results to provide guidelines for developing more realistic LLM student simulations, including introducing diverse student personalities, modeling gradual knowledge building, and promoting question-asking.

2 Related Work

2.1 AI-Simulated Students in Tutoring

Simulations of learners have been used for various purposes, including teacher preparation, peer learning, and system evaluation (VanLehn et al., 1994). For example, (Matsuda et al., 2007) examined whether a machine learning model can replicate how students learn to solve linear equations. However, many early simulations required significant effort, despite modeling narrow settings (Matsuda et al., 2015).

LLMs have made these simulations considerably more accessible. Recent applications include simulating students to assess the quality of automatically generated questions (Lu and Wang, 2024), or using LLMs as teachable agents for learning debugging (Ma et al., 2024). However, whether the resulting model is realistic enough to represent a real student is not fully understood. A survey (Käser and Alexandron, 2024) found that only 3% of the studies that simulate learners do a post-factum validation of their model. Moreover, there is a growing trend of not validating LLM outputs or relying on LLMs validating themselves (Shankar et al., 2024). In contrast, we base our work on first-hand insights of teachers communicating with LLM students, which provides a deeper understanding of the realism of these models.

Namely, we interviewed teachers who took part in the collection of an existing open-source dataset MathDial (Macina et al., 2023a). We chose this dataset over other educational datasets such as NCTE (Demszky and Hill, 2023), Bridge (Wang et al., 2024b), or TalkMoves (Suresh et al., 2022), because, to the best of our knowledge, it is the only publicly available dataset of interactions between real teachers and LLM-simulated students. Additionally, the MathDial dataset is enriched by teacher annotations such as realism ratings.

2.2 Believability of LLM Simulations

According to (Park et al., 2023), believable agents provide an illusion of life and present a facade of realism in the way they appear to make decisions and act of their own volition. One common approach to evaluating believability is to compare LLM-generated and real-world (Hämäläinen et al., 2023). In our work, we use a similar approach by contrasting the experiences of teachers with LLM simulations and real interactions.

What constitutes a believable simulation is often dependent on its context; for example, applications in psychology focus on personal experience (Chen et al., 2023), while character motivation is important in games research (AlJammaz et al., 2024). In education, the focus is often on cognitive aspects, with the social component addressed in a too broad or unsystematic way (Jin et al., 2024; Jinxin et al., 2023). In this work, we also account for the social aspect by using the Community of Inquiry framework, which we introduce next.

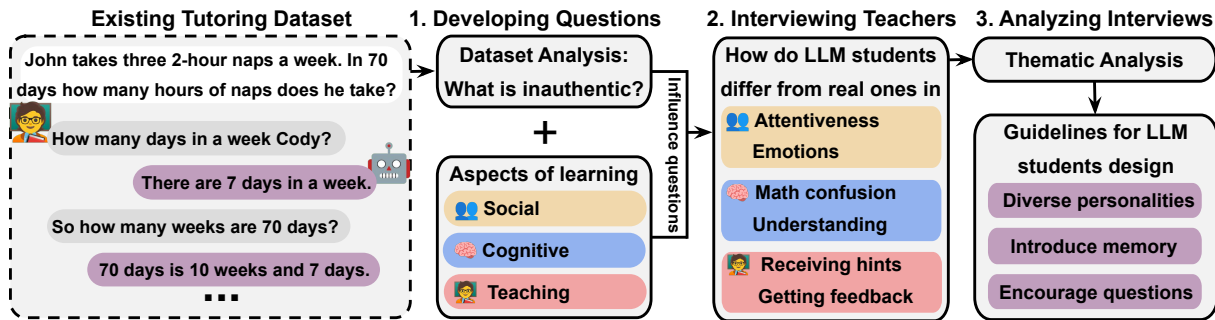


Figure 1: An illustration of our study stages: 1) We analyze an existing teacher-LLM tutoring dataset using the Community of Inquiry framework and derive interview questions from this analysis. 2) We interview teachers involved in data collection. 3) We outline guidelines for LLM student design and development.

2.3 Community of Inquiry and Scaffolding

Two important considerations in our study are the environment in which teachers use simulations and the form of teaching that is used. For the former, simulations are usually naturally used in an online setting, for example, through a web application. The Community of Inquiry (CoI) is a framework that is frequently used to understand online conversations in the context of education (Garrison, 2016). We adopt this framework to ground our interviews. CoI is based on three pillars: *social presence*, *cognitive presence*, and *teaching presence*. *Social presence* is defined as the ability of learners to project themselves socially and emotionally, thereby being perceived as “real people” in mediated communication (Garrison and Arbaugh, 2007). *Cognitive presence* is described in Garrison et al. (2001) as the extent to which learners are able to construct and confirm meaning through sustained reflection and discourse. *Teaching presence* is the design, facilitation, and direction of cognitive and social processes to achieve personally meaningful and educationally worthwhile learning outcomes (Garrison et al., 1999).

However, since the CoI framework gives limited attention to the active role of the teacher in guiding learning (Richardson and Lowenthal, 2017), we enriched the teaching presence with the Scaffolding theory (Wood et al., 1976; Quintana et al., 2004). In the setting of tutoring using scaffolding, the teacher guides the students and allows them to cognitively engage with the problem. Teachers usually follow a set of teaching strategies or moves (VanLehn, 2011; Nye et al., 2014; Hennessy et al., 2016) such as questioning with various effectiveness on learning (Michaels et al., 2008; Hennessy et al., 2016). The level of scaffolding needed depends on the student (Quintana et al., 2004; Van-

Lehn, 2011) and often includes actively engaging them with the problem, including failure, which is more productive for learning (Kapur and Bielaczyc, 2012). In our paper, we investigate how the behavior of LLM-simulated students influences teaching strategies.

3 Methods

To answer RQ1, we focus on the existing open-source dataset MathDial (Macina et al., 2023a), in which teachers helped LLM-simulated students to solve a math problem, as shown in Fig. 1. To understand teachers’ perceptions of LLM students’ realism, we conducted interviews with participants of the MathDial study, described in Section 3.1. We then describe how analyzing the MathDial dataset provided initial insights into the realism of LLM student simulations (Section 3.2) and informed the development of interview questions (Section 3.3).

3.1 Participants

We recruited 12 teachers or tutors of STEM subjects among those who took part in the MathDial study (Macina et al., 2023a) through Prolific.¹ We pre-screened participants to ensure they taught technical subjects, aligning with experience in the study. After signing the consent form, each participant received as a reminder three example dialogues that they personally had in the MathDial study.

10 out of 12 participants teach mathematics, while the rest focus on natural sciences. The participants teach children and adolescents, in institutions ranging from primary schools to universities. 3 participants have been teaching for less than 3 years, while the rest — for more than 11 years. Most participants (8 out of 12) are UK-based, while the others work in Canada. 10 participants are female,

¹<https://www.prolific.com/>

and the rest 2 are male, which is in line with the 80% proportion of female participants in the preceding study. The participants had an average of 35 dialogues with LLM students in the MathDial, with a standard deviation of 28. More details on participants' data can be found in Appendix A.

3.2 Developing Questions: MathDial Dataset Analysis

To design interview questions that capture teachers' perspectives on LLM students and address RQ1, we first analyzed the existing open-source tutoring MathDial dataset (Macina et al., 2023a), focusing on teachers' assessment of realism. In MathDial, teachers were asked to chat with a sixth-grade student simulated by an LLM and help them solve a math word problem. The LLM² was first prompted to generate an initial incorrect solution and then to act as a student who believes this solution is correct. The student persona was based on a name chosen from a culturally diverse set, a gender, and a specified type of confusion (see Macina et al. (2023a) for details). MathDial consists of 2861 dialogues produced by 90 participants, all of whom work as teachers. In addition to metadata such as teacher moves, each conversation is annotated by teachers with a rating on whether the interaction felt typical for a sixth-grade student, as well as optional open-ended "feedback about the conversation".

Topic Modeling of Teacher Feedback. We first analyze the open-ended feedback from teachers using Latent Dirichlet Allocation (LDA) (Blei et al., 2003) topic modeling to find any concerns they had about LLM simulations. One of the recurring topics in the LDA analysis was "repetitive". By manual review, we found that 51 of the 377 feedbacks provided mentioned the student giving repetitive answers. Furthermore, 6 of 44 teachers who left feedback mentioned that it was frustrating for them when the student was stuck on the same solution.

Statistical Analysis of Teacher-assessed Realism. Here, we show a quantitative analysis of interaction realism ratings and the corresponding conversations. We focused on how conversations that the tutors rated as non-typical (21% of conversations) differed from those rated as typical (79% of conversations). We have performed statistical tests to check the independence of features when comparing typical and non-typical interactions. We

chose features that are directly related to learning outcomes (e.g., the correctness of the final answer) or have the potential to impact student learning (e.g., emotions (Felten et al., 2006)). We used the Mann-Whitney U test (McKnight and Najab, 2010) for numerical features and the Chi-squared independence test (McHugh, 2013) for categorical features. Since we tested³ multiple hypotheses, we used the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) to control for small p-values that occur by chance.

According to statistical tests, features whose distribution differed significantly among typical and non-typical interactions included the correctness of final student answer, conversation length, count of teacher moves revealing solution, and sentiment scores of teacher utterances computed using VADER (Hutto and Gilbert, 2014) (see Table 1). Conversations rated by teachers as non-typical were usually longer, less successful, and the solution was revealed more often. Sentiment scores of teacher messages were lower in non-typical interactions, while student sentiment remained similar for both types of conversations, leaning towards higher values. The unusual conversations might be more difficult for teachers as the students struggle to progress in their solutions. Detailed results of the statistical analysis can be found in Appendix C.

Comparative Analysis between Educational Datasets. A comparison between LLM-human dialogues and human-human conversational datasets showed that LLM students tend to be more active in conversation compared to real-life students. That is, we have compared MathDial and datasets with human-human interactions: 1) with transcripts from math classes (Suresh et al., 2022; Demszky and Hill, 2023) and 2) with text-based one-on-one tutoring dialogues in language learning (Caines et al., 2022; Stasaski et al., 2020). We computed dialogue metrics such as the total word count and the proportion of words contributed by teachers and students. The proportion of words in LLM-human dialogue is heavily skewed towards the LLM student, who contributes 68% of the total words. This contrasts sharply with human-to-human conversational data, where students typically account for only 12% to 34% of the word count.

²gpt-3.5-turbo, accessed through the OpenAI GPT-3 API [gpt-3.5-turbo]; available at: <https://platform.openai.com/docs/models/gpt-3-5-turbo>

³The analysis was done in Python using SciPy (Virtanen et al., 2020) and statsmodels (Seabold and Perktold, 2010) libraries. The significance level was set at 0.05.

Table 1: Comparison of conversations rated by teachers as typical or not.

Statistic of conversational dynamics	Typical interactions	Non-typical interactions
Proportion of dialogues	79%	21%
Success rate in resolving confusion	83%	43%
Average dialogue length (in turns)	12 ± 5.4	16 ± 6.4
Average frequency of teachers revealing solution	0.14 ± 0.18	0.22 ± 0.2
Average sentiment score of teacher messages	0.15 ± 0.3	0.1 ± 0.29
Average sentiment score of student messages	0.18 ± 0.32	0.17 ± 0.31

3.3 Interview Procedure and Questions

All the interviews were held online and lasted 1 hour. The interviews started with a warm-up task, in which participants were consecutively shown two short tutoring dialogues and were asked to distinguish whether the student responses were written by a human or an AI. This exercise served as an introduction to the interview topic: comparing interactions with real and simulated students. The main part of the interview focused on the experiences participants themselves had when communicating with LLM students in the MathDial study.

We developed the interview questions from the Community of Inquiry and Scaffolding frameworks (Section 2.3) and the MathDial data analysis (Section 3.2). We iteratively refined the interview questions based on team discussions and feedback from pilot interviews. We finalized a set of 9 questions prompting teachers to reflect on how their real students differ from LLM students. Questions related to social presence explored the attentiveness of students and their emotions, as LLM students tend to be repetitive and show higher sentiment scores. Questions from the cognitive presence category were motivated by observed deviations in LLM students' learning and focused on students' confusion, understanding, and solutions complexity. Finally, to address teaching presence, we asked about teachers' strategies, especially scaffolding and giving feedback, as teachers resorted to telling parts of the solution when the LLM student behaved unusually. The full list of interview questions and the rationale behind them can be found in Appendix B.

To summarize the discussion of each question, participants were asked to answer a 5-point Likert scale question assessing interactions with LLM students, e.g., realism of their emotions (see Fig. 2). After the interview, participants were reimbursed 34 USD per hour. The research was approved by

the university Ethics Committee (EK-2024-N-6).

The interview data was analyzed using thematic analysis (Clarke and Braun, 2021). The initial coding was done by the main author, independently checked by two other team members, and iteratively refined. Finally, the codes were grouped into themes such as student emotions, language complexity, responsiveness, and demographics, as well as teachers' strategies and challenges.

4 Results

The main finding from the Likert scale survey answers (see Fig. 2) is that the LLM students did not authentically represent real human emotions. Apart from that, LLM students generally were able to simulate the learning process. Namely, aspects like teaching strategies, students' reactions to feedback, and math confusion were rated as more realistic. According to teacher ratings, LLM students were for the most part fairly attentive. In addition, the frequency of frustrating interactions and overly complicated solutions were rated relatively low.

Lack of Emotional Responses from LLM-simulated Students. 8 out of 12 participants noted that they did not seem to get particularly emotional responses from the LLM student. All teachers except one speculated that this perceived emotionlessness might just be the result of communication being only text-based and not being able to read the body language of the student.

To half of the participants, the student messages felt overall positive, with occasional emotions such as gratitude or relief. However, when asked about the common emotions of their real-life students when confused, all teachers primarily named negative ones such as frustration, fear, or embarrassment. A couple of participants believe their students react with denial, which LLMs did not portray: *'a human student is not going to immediately abandon a solution they've come up with.'* (P11).

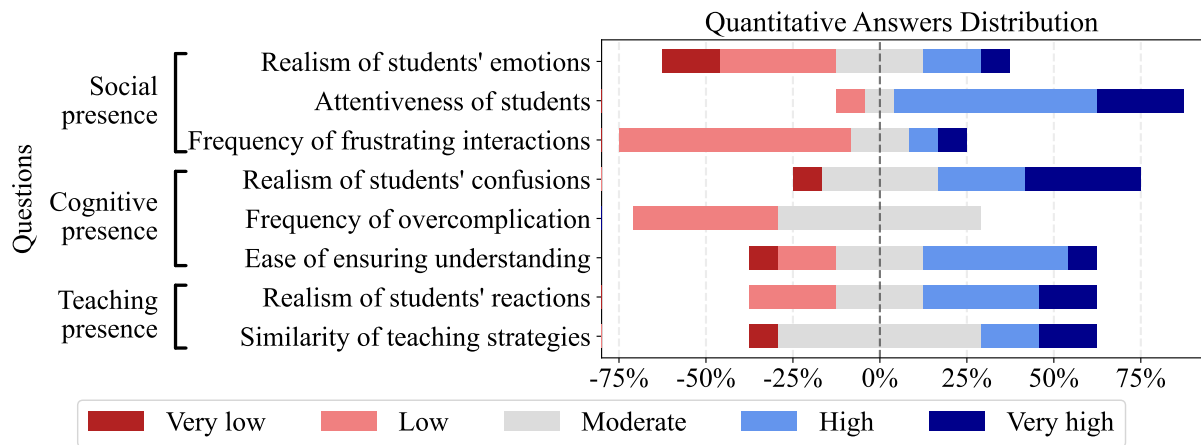


Figure 2: 5-point Likert scale ratings by teachers to questions about interactions with LLM-simulated students.

Eight participants mentioned that some of their students tend to give up or become quiet when they don't know how to solve a problem. LLM students failed to show this behavior: *'There weren't any students that just said, "Forget it. I can't do it. I give up." There was always a reattempt.'* (P02).

High Attentiveness. 10 out of 12 participants agreed that the LLM students felt rather attentive in the conversations. With real-life students, teachers see more diverse behaviors, e.g., *'You will have some children that are incredibly attentive, whereas, ... there are some children who have got very little interest in being there.'* (P12). Although LLM students generally resembled engaged students, P05 highlighted a difference: LLM students *'didn't ask any ... questions to help their understanding or make links with other things'*.

Inconsistent Behavior over Multiple Interactions. Two-thirds of participants pointed out that sometimes the LLM student felt like they were not following previous conversation. However, just as many teachers stressed that they are used to their students going off tangent, e.g., *'They always contradict themselves, and always say random things. And so that's not unusual at all.'* (P01).

Complex and Verbose Language Use by LLM-simulated Students. One of the frequently mentioned properties of LLM students which did not feel human-like to participants was the high language complexity. Also, two teachers noted how math formulas were extensively used by LLM students, which did not feel authentic. One participant highlighted how this hindered ensuring student understanding, as in real teaching students don't rely on *'mathematical language necessarily, they would actually talk to you in words.'* (P09).

Adaptation of Teaching Strategies for Interactions with LLM Students. Teachers have to adapt to the pace of their students; therefore, they pay high attention to the process of student learning, and they find several differences between LLM and actual students.

All participants emphasized the importance of scaffolding by breaking the problem down into smaller steps, as well as trying to give hints and not reveal parts of the solution. However, a quarter of participants noted that these approaches sometimes had to be adjusted when talking to LLM students, namely, teachers had to resort to telling parts of the solution. Two participants supposed that LLM-simulated students might have struggled because *'rather than try and take a step at a time, they were trying to solve everything altogether.'* (P01).

In MathDial, participants also frequently used approaches such as asking questions, finding other ways to solve a problem, and repeating. Participants found it to be *'no different to real life: you often have to repeat things and, if someone doesn't appear to understand how you said something the first time, you have to rephrase it.'* (P05). For P11, the experience of communicating with LLM felt *'analogous to working with humans: if your instructions are bad, your results are bad. ... as we ... learn more about how AI works, we are kind of also learning how humans work.'* (P11).

The Influence of Context on the Perception of Interactions with LLM-simulated Students. The participants teach students from different backgrounds, and some of their opinions on LLM students are also influenced by their diverse experiences. For example, P06 described that in some of the MathDial dialogues, *'That was interpretation*

where the gap was rather than actually a problem with the math. A common issue, actually, because a lot of our students ... have dyslexia'. Other teachers also mentioned dyscalculia, being non-verbal and having other special education needs, or having English not as their first language.

Differences in perception of interactions with LLM students could also be caused by the settings in which the participants teach. For example, P05 primarily works as a tutor and commented about LLM-simulated students: *'They seem to demonstrate a good growth mindset. That was probably quite different with students ... I work with, because it's one-to-one tuition and a lot are lacking that confidence already.'* (P05).

Half of the participants compared students' behavior across different subjects, e.g.: *'I have taught many subjects, and the only ones that really results sometimes in sobbing is math. ... Math can really trigger deep, deep emotions.'* (P11).

5 Discussion

5.1 Guidelines for LLM Students Design: Four Behavior Types

As our RQ1 aims to assess how believable the LLM student simulations are, we identify different groups of student behaviors in real life. Specifically, we do this based on the CoI framework and Scaffolding theory. In real-life education, some students need more scaffolding support, which means that the teacher provides step-by-step guidance to them and needs to engage them more actively in the process. Other students are more independent and actively participate in the problem-solving activity. Within both of these groups, we more specifically examine the social and cognitive presence of the students. That is, social presence relates to behaviors that help students engage and interact with the tutor, including demonstrating emotional expressiveness. On the other hand, cognitive presence focuses on how students process information, solve problems, and build knowledge. Table 2 provides an overview of behaviors not captured by LLM students for each category, as well as the importance participants placed on these issues and our proposed solutions, thereby addressing RQ2.

High Scaffolding Needs and Social Presence. Most of the interviewees agreed that LLM-simulated students were too engaged in conversations. We suggest that such simulations should have varying customizable levels of engagement,

much as real students would. Sometimes, the simulated student might even stay silent or lose interest and attention, which could also give a valuable reason for teachers to self-reflect on the quality of teaching (Markel et al., 2023).

Participants often found the language used by LLM students to be too complex, lengthy, and technical, especially for children. Therefore, we propose having more variations in language complexity, intentionally regulating the length and formality of responses. Other suggestions include introducing grammar, spelling, or punctuation mistakes and, in the case of mathematics, limiting notations and the rigor of equations.

In addition to these behavioral tendencies, LLM students lacked emotional responses, especially the more negative ones: frustration, fear, or embarrassment. We propose to model a diverse range of student personalities, which in turn would lead to a diverse representation of emotions (Rusting and Larsen, 1997; Santos, 2016). A popular approach to portraying personalities is the Big Five theory (Costa and McCrae, 1999) which is also widely used in the development of LLMs (Jiang et al., 2024; Liu et al., 2024). This method of modeling diverse personalities might also broader represent previously mentioned engagement levels (Donovan et al., 2020; Zhang et al., 2020).

High Scaffolding Needs and Cognitive Presence. The way in which some LLM students' cognitive processes worked seemed unrealistic to our participants: their knowledge sometimes did not build gradually but made huge jumps. This is not only unrealistic, but it deprives teachers of practicing a recognized approach to teaching: leveraging the zone of proximal development (Vygotsky, 1978). The study (Jin et al., 2024) also focused on this limitation of LLMs and modulated the knowledge state as the conversation progressed, which could also be used in the setting of our research. One improvement we suggest future works to integrate is knowledge tracing (Scarlatos et al., 2025; Fu et al., 2024a) which is commonly used to estimate student knowledge and predict their responses. Another aspect that could be modeled to resemble human learning is forgetting information over time (Zhong et al., 2024).

Low Scaffolding Needs and Social Presence. Another behavior that LLM students failed to represent was asking questions. This meant that teachers had more control over the discussion flow, which is not always the case in real life. Jin et al. (2024) pro-

Table 2: Real-life student behavior LLMs failed to show and suggested solutions.

	High scaffolding needs	Low scaffolding needs
Social presence	<ul style="list-style-type: none"> ■ Writing simple and short ■ Having negative emotions, being disengaged ✂ <i>Introducing diverse personalities</i> 	<ul style="list-style-type: none"> ■ Asking questions ✂ <i>Promoting question-asking</i>
Cognitive presence	<ul style="list-style-type: none"> ■ Gradual knowledge-building ✂ <i>Introducing memory</i> 	<ul style="list-style-type: none"> ● Disagreeing with teacher ● Changing tactic based on feedback ✂ <i>No interventions needed</i>

■ Human-simulation gap ● Realistic behavior

poses a way to address this in the case of using simulated LLM students in the learning-by-teaching scenario. That is, their solution was to switch to the mode of asking questions with a period of three messages. We propose to use a similar technique that is more context-aware.

Low Scaffolding Needs and Cognitive Presence. Some students of our participants react with denial when told that their solution is wrong. In contrast, LLM students sometimes agree too readily with the teacher, completely changing their approach. This tendency of LLMs is called sycophancy bias (Perez et al., 2023) and originates from LLMs designed to follow instructions. Although this is useful in many contexts, when practicing interactions with a student, it is beneficial to put the effort into finding the correct method together.

Our participants sometimes observed that the LLM student was stuck on the same math problem solution, which was mostly recognized as common student behavior. This is in line with previous research, as LLMs are prone to being more stubborn when discussing mathematics than subjective topics (Ranaldi and Pucci, 2023). Dealing with students who struggle to progress is important for teachers; therefore, we do not recommend eliminating such types of interactions.

Practical Application Example. We propose that designers of LLM student simulations adopt a profile-oriented design approach (Jin et al., 2025; Wolff and Seffah, 2011), which involves incorporating diverse student personality traits and learning behaviors described in Table 2. Teachers could first pick a specific profile type of a simulated student, as well as their learning pace and knowledge level of a given topic. Using a base-prompt, a specific chatbot could be created for the teachers to interact with. A post-generation prompt could be used to make the final utterance shorter and simpler. This approach could increase the diversity of simulated

student behaviors while ensuring consistency and realism, thereby making the simulations more inclusive and valuable for teacher practice.

5.2 Teacher Perceived Limitations of LLM Students

An overall trend we observed during the analysis was that LLMs mainly represented only certain student types and behaviors, depriving participants of richer teaching experiences. LLMs indeed have a tendency to portray an averaged representation of the data they were trained on. Our suggestion is to rather evaluate models by simulating the spectrum of student personas to allow for a more comprehensive teaching experience.

While LLMs often portrayed attentive students, some participants felt they resembled students with more surprising traits such as having learning challenges like dyslexia. We propose that LLM simulations should have the option to configure the simulated context, allowing teachers to get more valuable experience.

6 Conclusion

In this paper, we investigate the effectiveness of LLMs in simulating real K12 student behaviors by gathering insights from teachers who have tutored LLM-simulated students. Our findings reveal that LLMs fall short in replicating properties inherent in real-life students: emotions, especially negative, rather simple language, and the steady pace of learning. We address this issue by proposing a categorization of real-life student behaviors based on the level of needed scaffolding and relation to cognitive or social presence, and assess the LLM performance in representing each category. This categorization could serve as a guideline for evaluating novel LLM models for student simulations, for example by including more diverse student behavior types. Addressing these issues could enhance the

effectiveness and realism of future LLM student simulations in education, ultimately making educational resources more accessible, affordable, and personalized for a broader population.

Limitations

Our study has several limitations that future work could address. First, the dataset we analyzed generated the student simulations with an older GPT-3.5-turbo model. Future work could explore the differences in how other LLMs simulate students. Interestingly, for some tasks, more advanced models might perform worse: e.g., in Milička et al. (2024) GPT-4 (OpenAI, 2023), when prompted to simulate a one-year-old, gave more correct answers to logical questions than GPT-3.5-turbo. Moreover, studies comparing different LLMs find that some are more sensitive to the phrasing of math problems (Opedal et al., 2024) or less capable of reflecting emotional states (Ishikawa and Yoshino, 2025).

Secondly, the demographics of the study participants were limited: most of the participants were from the UK and the majority were female. While the high proportion of female teachers in our study reflects trends in the teaching profession (Government data about the UK's different ethnic groups, 2024), we acknowledge the potential impact of gender on the study results. For example, Sun et al. (2024) has shown that gender could influence the perceived anthropomorphism of a simulated persona. Further work could conduct larger-scale studies with more diverse demographics to analyze these dynamics further.

Finally, we limited the study scope to mathematics. However, as our participants also highlighted, real-life students' behavior differs depending on the subject. Similarly, LLMs might have varying attitudes towards different subjects, e.g., GPT models exhibit more anxiety when talking about mathematics (Abramski et al., 2023). Exploring other subjects and educational contexts could provide a more comprehensive understanding of the use of LLMs in student simulation.

Acknowledgements

We thank Peng Cui for discussions and feedback on early versions. Jakub Macina is supported by ETH AI Center doctoral fellowship.

References

- Katherine Abramski, Salvatore Citraro, Luigi Lombardi, Giulio Rossetti, and Massimo Stella. 2023. [Cognitive network science reveals bias in gpt-3, gpt-3.5 turbo, and gpt-4 mirroring math anxiety in high-school students](#). *Big Data and Cognitive Computing*, 7(3):124.
- Jair J Aguilar and Seokmin Kang. 2023. [Innovating with in-service mathematics teachers' professional development: The intersection among mixed-reality simulations, approximation-of-practice, and technology-acceptance](#). *International Electronic Journal of Mathematics Education*, 18(4):em0750.
- Jair J Aguilar and James A Telese. 2020. [Perceptions and opinions of the usability of simulations in a mathematics methods course for elementary pre-service teachers](#). *Journal of Education and Practice*, 11(12).
- Rehaf AlJammaz, Noah Wardrip-Fruin, and Michael Mateas. 2024. [Navigating faction systems: Insights and recommendations for more believable npcs in video games](#). In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, pages 1–11.
- Karen T Arnesen, Charles R Graham, Cecil R Short, and Douglas Archibald. 2019. [Experiences with personalized learning in a blended teaching course for preservice teachers](#). *Journal of online learning research*, 5(3):275–310.
- 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.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. [Latent dirichlet allocation](#). *Journal of machine Learning research*, 3(Jan):993–1022.
- Elizabeth Gates Bradley and Brittany Kendall. 2014. [A review of computer simulations in teacher education](#). *Journal of Educational Technology Systems*, 43(1):3–12.
- Andrew Caines, Helen Yannakoudakis, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2022. [The teacher-student chatroom corpus version 2: more lessons, new annotation, automatic detection of sequence shifts](#). In *Swedish Language Technology Conference and NLP4CALL*, pages 23–35.
- Catherine C Chase, Doris B Chin, Marily A Oppezzo, and Daniel L Schwartz. 2009. [Teachable agents and the protégé effect: Increasing the effort towards learning](#). *Journal of science education and technology*, 18:334–352.
- Siyuan Chen, Mengyue Wu, Kenny Q Zhu, Kunyao Lan, Zhiling Zhang, and Lyuchun Cui. 2023. [Llm-empowered chatbots for psychiatrist and patient simulation: application and evaluation](#). *arXiv preprint arXiv:2305.13614*.

- Doris B Chin, Ilsa M Dohmen, and Daniel L Schwartz. 2013. [Young children can learn scientific reasoning with teachable agents](#). *IEEE Transactions on Learning Technologies*, 6(3):248–257.
- Victoria Clarke and Virginia Braun. 2021. [Thematic analysis: a practical guide](#). *SAGE Publications Ltd*.
- PT Costa and RR McCrae. 1999. [A five-factor theory of personality](#). *Handbook of personality: Theory and research*, 2(01):1999.
- Januard Deñola Dagdag and Milky Mae D Bandera. 2021. [Understanding the factors that influence students’ behavior: Key towards an effective teaching](#). *Pedagogi: Jurnal Ilmu Pendidikan*, 21(2):144–148.
- Dorotya Demszyk 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.
- Ryan Donovan, Aoife Johnson, Aine deRoiste, and Ruairi O’Reilly. 2020. [Quantifying the links between personality sub-traits and the basic emotions](#). In *Computational Science and Its Applications–ICCSA 2020: 20th International Conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part II 20*, pages 521–537. Springer.
- Joel Eapen and VS Adhithyan. 2023. [Personalization and customization of llm responses](#). *International Journal of Research Publication and Reviews*, 4(12):2617–2627.
- Peter Felten, Leigh Z Gilchrist, and Alexa Darby. 2006. [Emotion and learning: feeling our way toward a new theory of reflection in service-learning](#). *Michigan Journal of Community Service Learning*, 12(2):38–46.
- Lingyue Fu, Hao Guan, Kounianhua Du, Jianghao Lin, Wei Xia, Weinan Zhang, Ruiming Tang, Yasheng Wang, and Yong Yu. 2024a. [Sinkt: A structure-aware inductive knowledge tracing model with large language model](#). In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 632–642.
- Yue Fu, Sami Foell, Xuhai Xu, and Alexis Hiniker. 2024b. [From text to self: Users’ perception of aimc tools on interpersonal communication and self](#). In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–17.
- D Randy Garrison. 2016. [E-learning in the 21st century: A community of inquiry framework for research and practice](#). Routledge.
- D Randy Garrison, Terry Anderson, and Walter Archer. 1999. [Critical inquiry in a text-based environment: Computer conferencing in higher education](#). *The internet and higher education*, 2(2-3):87–105.
- D Randy Garrison, Terry Anderson, and Walter Archer. 2001. [Critical thinking and computer conferencing: A model and tool to assess cognitive presence](#). *American Journal of Distance Education*.
- D Randy Garrison and J Ben Arbaugh. 2007. [Researching the community of inquiry framework: Review, issues, and future directions](#). *The Internet and higher education*, 10(3):157–172.
- Government data about the UK’s different ethnic groups. 2024. [School teacher workforce](#). Accessed: 2024-09-09.
- Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. 2023. [Evaluating large language models in generating synthetic hci research data: a case study](#). In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–19.
- Sara Hennessy, Sylvia Rojas-Drummond, Rupert Higham, Ana María Márquez, Fiona Maine, Rosa María Ríos, Rocío García-Carrión, Omar Torrelblanca, and María José Barrera. 2016. [Developing a coding scheme for analysing classroom dialogue across educational contexts](#). *Learning, culture and social interaction*, 9:16–44.
- Clayton Hutto and Eric Gilbert. 2014. [Vader: A parsimonious rule-based model for sentiment analysis of social media text](#). In *Proceedings of the international AAAI conference on web and social media*, volume 8(1), pages 216–225.
- Shin-nosuke Ishikawa and Atsushi Yoshino. 2025. [Ai with emotions: Exploring emotional expressions in large language models](#). In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 614–627.
- Hang Jiang, Xijie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. 2024. [PersonaLLM: Investigating the ability of large language models to express personality traits](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3605–3627, Mexico City, Mexico. Association for Computational Linguistics.
- Hyoungwook 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 CHI Conference on Human Factors in Computing Systems*, pages 1–28.
- Hyoungwook Jin, Minju Yoo, Jeongeon Park, Yokyung Lee, Xu Wang, and Juho Kim. 2025. [Teachtune: Reviewing pedagogical agents against diverse student profiles with simulated students](#). In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–28.
- Shi Jinxin, Zhao Jiabao, Wang Yilei, Wu Xingjiao, Li Jiawen, and He Liang. 2023. [Cgmi: Configurable general multi-agent interaction framework](#). *arXiv preprint arXiv:2308.12503*.

- Manu Kapur and Katerine Bielaczyc. 2012. [Designing for productive failure](#). *Journal of the Learning Sciences*, 21(1):45–83.
- 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.
- Grgur Kovač, Rémy Portelas, Masataka Sawayama, Peter Ford Dominey, and Pierre-Yves Oudeyer. 2024. [Stick to your role! stability of personal values expressed in large language models](#). *PloS one*, 19(8):e0309114.
- Zhengyuan Liu, Stella Yin, Geyu Lin, and Nancy Chen. 2024. [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.
- Deborah Loewenberg Ball and Francesca M Forzani. 2009. [The work of teaching and the challenge for teacher education](#). *Journal of teacher education*, 60(5):497–511.
- 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*, pages 16–27.
- Qianou Ma, Hua Shen, Kenneth Koedinger, and Sherry Tongshuang Wu. 2024. [How to teach programming in the ai era? using llms as a teachable agent for debugging](#). In *International Conference on Artificial Intelligence in Education*, pages 265–279. Springer.
- Jakub Macina, Nico Daheim, Sankalan Pal 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, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023b. [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.
- 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.
- Noboru Matsuda, William W Cohen, and Kenneth R Koedinger. 2015. [Teaching the teacher: tutoring sim-student leads to more effective cognitive tutor authoring](#). *International Journal of Artificial Intelligence in Education*, 25:1–34.
- Noboru Matsuda, William W Cohen, Jonathan Sewall, Gustavo Lacerda, and Kenneth R Koedinger. 2007. [Predicting students’ performance with simstudent: Learning cognitive skills from observation](#). *Frontiers in Artificial Intelligence and Applications*, 158:467.
- Oliver McGarr. 2021. [The use of virtual simulations in teacher education to develop pre-service teachers’ behaviour and classroom management skills: implications for reflective practice](#). *Journal of Education for Teaching*, 47(2):274–286.
- Mary L McHugh. 2013. [The chi-square test of independence](#). *Biochemia medica*, 23(2):143–149.
- Patrick E McKnight and Julius Najab. 2010. [Mann-whitney u test](#). *The Corsini encyclopedia of psychology*, pages 1–1.
- Sarah Michaels, Catherine O’Connor, and Lauren B Resnick. 2008. [Deliberative discourse idealized and realized: Accountable talk in the classroom and in civic life](#). *Studies in philosophy and education*, 27:283–297.
- Jiří Milička, Anna Marklová, Klára VanSlambrouck, Eva Pospíšilová, Jana Šimsová, Samuel Harvan, and Ondřej Drobil. 2024. [Large language models are able to downplay their cognitive abilities to fit the persona they simulate](#). *Plos one*, 19(3):e0298522.
- 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.
- Andreas Opedal, Alessandro Stolfo, Haruki Shirakami, Ying Jiao, Ryan Cotterell, Bernhard Schölkopf, Abulhair Saparov, and Mrinmaya Sachan. 2024. [Do language models exhibit the same cognitive biases in problem solving as human learners?](#) In *Proceedings of the 41st International Conference on Machine Learning*, pages 38762–38778.
- OpenAI. 2023. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. [Generative agents: Interactive simulacra of human behavior](#). In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.
- Ethan Perez, Sam Ringer, Kamilè Lukošiušė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, and 1 others. 2023. [Discovering language model behaviors with model-written evaluations](#). In

- Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada. Association for Computational Linguistics.
- Chris Quintana, Brian Reiser, Elizabeth Davis, Joseph Krajcik, Eric Fretz, Ravit Duncan, Eleni Kyza, Daniel Edelson, and Elliot Soloway. 2004. A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences*, 13:337–386.
- Leonardo Ranaldi and Giulia Pucci. 2023. When large language models contradict humans? large language models’ sycophantic behaviour. *arXiv preprint arXiv:2311.09410*.
- Brian J. Reiser. 2004. Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *Journal of the Learning Sciences*, 13(3):273–304.
- Jennifer C Richardson and Patrick Lowenthal. 2017. Instructor social presence: Learners’ needs and a neglected component of the community of inquiry framework. In *Social Presence in Online Learning*, pages 86–98. Routledge.
- Cheryl L Rusting and Randy J Larsen. 1997. Extraversion, neuroticism, and susceptibility to positive and negative affect: A test of two theoretical models. *Personality and individual differences*, 22(5):607–612.
- Olga C Santos. 2016. Emotions and personality in adaptive e-learning systems: an affective computing perspective. *Emotions and personality in personalized services: Models, evaluation and applications*, pages 263–285.
- Alexander Scarlatos, Ryan S. Baker, and Andrew Lan. 2025. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th Learning Analytics and Knowledge Conference, LAK 2025, Dublin, Ireland, March 3-7, 2025*. ACM.
- Skipper Seabold and Josef Perktold. 2010. Statsmodels: econometric and statistical modeling with python. *SciPy*, 7(1).
- Melissa K Shank. 2023. Novice teachers’ training and support needs in evidence-based classroom management. *Preventing School Failure: Alternative Education for Children and Youth*, 67(4):197–208.
- Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14.
- 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.
- Zhida Sun, Manuele Reani, Yunzhong Luo, and Zhuolan Bao. 2024. Anthropomorphism in chatbot systems between gender and individual differences. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–7.
- 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.
- Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. 2021. Understanding the capabilities, limitations, and societal impact of large language models. *Preprint*, arXiv:2102.02503.
- Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational psychologist*, 46(4):197–221.
- Kurt VanLehn, Stellan Ohlsson, and Rod Nason. 1994. Applications of simulated students: An exploration. *Journal of artificial intelligence in education*, 5:135–135.
- 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.
- Lev Semenovich Vygotsky. 1978. *Mind in society: The development of higher psychological processes*. Harvard University Press.
- Angelina Wang, Jamie Morgenstern, and John P Dickerson. 2024a. Large language models cannot replace human participants because they cannot portray identity groups. *arXiv preprint arXiv:2402.01908*.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024b. 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.
- Xu Wang, Meredith Thompson, Kexin Yang, Dan Roy, Kenneth R Koedinger, Carolyn P Rose, and Justin Reich. 2021. Practice-based teacher questioning strategy training with elk: A role-playing simulation for eliciting learner knowledge. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–27.
- Dan Wolff and Ahmed Seffah. 2011. Ux modeler: a persona-based tool for capturing and modeling user

experience in service design. In *IFIP WG 13.2 Workshop at INTERACT 2011*, pages 7–16.

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.

Xiaojie Zhang, Guang Chen, and Bing Xu. 2020. [The influence of group big-five personality composition on student engagement in online discussion](#). *International Journal of Information and Education Technology*, 10(10):744–750.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. [Memorybank: Enhancing large language models with long-term memory](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38(17), pages 19724–19731.

A Participant Information

Table 3: Participant demographics, teaching experience, and the number of dialogues with LLM students in MathDial (Macina et al., 2023a).

ID	Age	Gender	Country	Student Ages	Subjects	Teaching Experience	#Dialogues
P01	40–49	Female	UK	5–9	Primary school subjects, including mathematics	15+ years	50
P02	40–49	Female	Canada	10–14	Mathematics	11–15 years	40
P03	30–39	Female	UK	0–9	Primary school subjects, including mathematics	1–3 years	100
P04	40–49	Female	UK	5–9	Primary school subjects, including mathematics	15+ years	70
P05	30–39	Female	UK	5–17	Mathematics, computer science, literature	11–15 years	19
P06	40–49	Female	UK	18+	Environmental science	15+ years	35
P07	20–29	Female	Canada	5–14, 18+	Mathematics, chemistry	1–3 years	30
P08	40–49	Male	UK	18+	Applied statistics	15+ years	20
P09	50–59	Female	UK	10–17	Mathematics, English as a foreign language, literature	15+ years	25
P10	20–29	Female	Canada	5–17	Biochemistry, English as a foreign language	1–3 years	10
P11	50–59	Female	Canada	5–17	Mathematics, computer science	15+ years	10
P12	40–49	Male	UK	5–14	Primary school subjects, including mathematics	11–15 years	5

B Interview Questions

Table 4: Interview questions and their connection to preceding MathDial analysis and theoretical frameworks: Community of Inquiry (CoI) (Garrison, 2016) and Scaffolding (Reiser, 2004)

Qualitative and Quantitative Questions	Rationale
<p>Question: In MathDial, how <i>attentive</i> were the students?</p> <p>Probes: Did it seem like the student was following what you were saying? If not, what were the examples when the student seemed like they didn't follow you? Were there cases when the student contradicted themselves? How do these cases compare to your real life experience?</p> <p>Evaluation: How attentive the MathDial students felt like? 1 (Not at all) - 5 (Extremely)</p>	<p>MathDial analysis: Some participants mentioned in the feedback field that the student's messages were repetitive</p> <p>CoI framework: Social presence</p>
<p>Question: How <i>engaged</i> are your students in math problem discussions?</p> <p>Probes: How much do they participate in conversation? How does it compare with the dialogues you had in the study?</p> <p>Evaluation: How engaged were the MathDial students? 1 (Much less than your students) - 5 (Much more than your students)</p>	<p>MathDial analysis: Compared to human-human educational datasets, the student in MathDial talks much more</p> <p>CoI framework: Social presence</p>
<p>Question: Which interactions with MathDial students were <i>frustrating</i> for you?</p> <p>Probes: How similar were they to the real life teaching? How do you deal with these?</p> <p>Evaluation: How often were MathDial interactions frustrating? 1 (Never) - 5 (Almost always)</p>	<p>MathDial analysis: The participants answers tend to have lower sentiment scores in conversations where the student interactions are perceived as non-typical</p> <p>CoI framework: Social presence</p>
<p>Question: Did you adjust your <i>teaching strategies</i> in MathDial?</p> <p>Probes: For example, how did you balance giving hints and giving parts of the solution? How do you do it in your real life teaching?</p> <p>Evaluation: How similar to real life were your teaching strategies in MathDial? 1 (Not at all) - 5 (Extremely)</p>	<p>MathDial analysis: The teachers tended to more often reveal part of the solution in conversations with non-typical interactions</p> <p>Theoretical framework: Scaffolding theory and Teaching presence from CoI</p>
<p>Question: What <i>feedback</i> do you give your students?</p> <p>Probes: How do they typically react to it? Were the student's reactions to feedback in MathDial similar to the typical reaction of your students?</p> <p>Evaluation: How realistic were students' reactions to feedback in MathDial? 1 (Not at all) - 5 (Extremely)</p>	<p>MathDial analysis: There was a cap on the number of messages teachers could send, so the feedback might have been rather limited</p> <p>CoI framework: Teaching presence</p>

Table 4: Interview questions and their connection to preceding MathDial analysis and theoretical frameworks: Community of Inquiry (CoI) (Garrison, 2016) and Scaffolding (Reiser, 2004)

Qualitative and Quantitative Questions	Rationale
<p>Question: What <i>emotions</i> are common to your students due to math confusion?</p> <p>Probes: How closely was it represented in the MathDial study?</p> <p>6 How do you behave when the students convey emotions you listed?</p> <p>Evaluation: How realistic were students' emotions in MathDial? 1 (Not at all) - 5 (Extremely)</p>	<p>MathDial analysis: Sentiment score of student utterances is distributed independently of how typical the student interactions were</p> <p>CoI framework: Social presence</p>
<p>Question: What was the common <i>reason of confusion</i> in MathDial?</p> <p>7 Probes: How does it align with most common issues your students have?</p> <p>Evaluation: How realistic was students' confusion in MathDial? 1 (Not at all) - 5 (Extremely)</p>	<p>MathDial analysis: Some teachers assessed student's confusion as non-typical</p> <p>CoI framework: Cognitive presence</p>
<p>Question: In real life teaching, how do you ensure the <i>concept understanding</i>?</p> <p>Probes: What do you usually do after the correct solution was found? Do you continue the problem discussion? If yes, how?</p> <p>8 Evaluation: It was easy to ensure understanding of students in MathDial 1 (Strongly disagree) - 5 (Strongly agree)</p>	<p>MathDial analysis: Mainly the teachers stopped the dialogue after the student has found the correct solution</p> <p>CoI framework: Cognitive presence</p>
<p>Question: In real life teaching, how do you handle <i>overcomplicated solutions</i>?</p> <p>Probes: For example, do you let them explore their solution further? Or do you try to guide them to an easier solution?</p> <p>9 Evaluation: How often were MathDial solutions overcomplicated? 1 (Never) - 5 (Almost always)</p>	<p>MathDial analysis: LLM students sometimes used more complex methods (e.g., introducing variables) when the problem could be solved without them</p> <p>CoI framework: Cognitive presence</p>

C Statistical Tests on MathDial

Table 5: Results of statistical tests comparing distribution of numerical features in typical and non-typical interactions in MathDial. U-statistic (McKnight and Najab, 2010) and p-value adjusted using Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) are provided, with significant results (adjusted p-value < 0.05) marked with an asterisk (*).

(a) Teacher-annotated and sentiment features			(b) Interaction and problem-related metrics		
Feature	U-statistic	Adjusted p-value	Feature	U-statistic	Adjusted p-value
Teacher-assessed cognition of LLM student			Conversation characteristics		
Confusion authenticity	220357	7.47e-145*	Number of turns	920056	5.24e-46*
Step of first error in solution	74669	7.02e-01	Conversation index	685230	4.61e-01
Counts of teacher-annotated teacher moves			Ground-truth solution characteristics		
Revealing parts of solution	876991	6.93e-36*	Number of words	638996	3.04e-01
Constraining to make progress	790520	3.75e-12*	Number of steps	650522	6.35e-01
Talking casually	600816	7.49e-04*	Math problem characteristics		
Generalizing aspects of problem	721417	3.52e-03*	Order of the problem in session	648169	6.81e-01
Teacher sentiment scores			Identifier	652030	7.02e-01
Mean	605884	3.52e-03*	Sentiment score	660511	8.98e-01
Median	605894	3.52e-03*	Number of words	669497	8.98e-01
Minimum	606569	3.52e-03*	Arithmetic operation percentages in solution		
Standard deviation	620603	3.62e-02*	Addition	701925	7.25e-02
Maximum	631284	1.46e-01	Subtraction	676748	6.73e-01
LLM student sentiment scores			Multiplication	652588	6.73e-01
Minimum	615997	1.77e-02*	Division	663954	9.77e-01
Maximum	690558	2.97e-01			
Mean	653972	7.41e-01			
Median	655628	7.98e-01			
Standard deviation	661922	8.96e-01			

Table 6: Results of statistical tests comparing distribution of categorical features in typical and non-typical interactions in MathDial. χ^2 statistic (McHugh, 2013) and p-value adjusted using Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) are provided, with significant results (adjusted p-value < 0.05) marked with an asterisk (*).

Feature	χ^2 statistic	Adjusted p-value
Teacher-assessed cognition of LLM student		
Correctness of final answer	479.83	1.28e-103*
Error category (calculation or conceptual)	6.38	6.35e-01
Teacher and LLM student data		
Teacher identifier	358.66	3.74e-33*
Student’s name (from prompt)	40.82	3.55e-02*
Student’s math struggle type (from prompt)	9.56	1.97e-01
Student’s gender (from prompt)	0.81	6.35e-01
Topics mentioned in math problem		
Time	0.15	8.68e-01
Percent	0.09	8.96e-01
Money	0.07	8.96e-01
Age	0.03	8.96e-01
Fractions	0.04	8.96e-01

Adapting LLMs for Minimal-edit Grammatical Error Correction

Ryszard Staruch

Adam Mickiewicz University
Center for Artificial Intelligence
ryszard.staruch@amu.edu.pl

Filip Graliński

Adam Mickiewicz University
filipg@amu.edu.pl
Snowflake
filip.gralinski@snowflake.com

Daniel Dzienisiewicz

Adam Mickiewicz University
dzienis@amu.edu.pl

Abstract

Decoder-only large language models have shown superior performance in the fluency-edit English Grammatical Error Correction, but their adaptation for minimal-edit English GEC is still underexplored. To improve their effectiveness in the minimal-edit approach, we explore the error rate adaptation topic and propose a novel training schedule method. Our experiments set a new state-of-the-art result for a single-model system on the BEA-test set. We also detokenize the most common English GEC datasets to match the natural way of writing text. During the process, we find that there are errors in them. Our experiments analyze whether training on detokenized datasets impacts the results and measure the impact of the usage of the datasets with corrected erroneous examples. To facilitate reproducibility, we have released the source code used to train our models.¹

1 Introduction

Grammatical Error Correction (GEC) is a Natural Language Processing task that covers the detection and correction of errors in texts. Current state-of-the-art models are either Sequence-to-Edit (Seq2Edit) models (encoder-only Transformers) that are trained to tag erroneous tokens and apply proper changes to them (Omelianchuk et al., 2020), or Sequence-to-Sequence (Seq2Seq) models (encoder-decoder Transformers) that are trained to generate the correct version of a given text (Rothe et al., 2021).

Over the years, two main directions have been established in GEC research: minimal-edit GEC and fluency-edit GEC (Bryant et al., 2023). The former focuses on applying only the minimal changes necessary to make the text grammatical and error-free. In contrast, fluency-edit GEC goes beyond minimal corrections to achieve native-language fluency.

Current decoder-only large language models (LLMs) achieve state-of-the-art performance on many NLP tasks. Instruction-tuned LLMs are able to produce high-quality texts and correct errors in the zero-shot approach, even without task-specific fine-tuning (Davis et al., 2024). On the JFLEG dataset (Napoles et al., 2017), which is a fluency-edit GEC dataset, the GPT3 and GPT4 models are capable of producing state-of-the-art results (Loem et al., 2023; Coyne et al., 2023). LLMs were also used by the winners of the recent multilingual grammatical error correction shared task – MultiGEC-2025 (Masciolini et al., 2025).

However, for a minimal-edit GEC, there is only one research work that reports better results compared to other solutions on English minimal-edit GEC benchmarks (Liang et al., 2025). The problem encountered by LLMs can be explained by the phenomenon of overcorrection (Fang et al., 2023).

To further explore LLMs adaptation for minimal-edit GEC, there is a need to find solutions that could allow LLMs to produce more strict outputs. Junczys-Dowmunt et al. (2018) by exploring the error rate adaptation topic show that neural network based solutions need more erroneous examples. Their experiments show that removing the correct examples leads to greater recall. Our intuition is that for modern LLMs, which are able to produce fluent corrections with high linguistic freedom even in the zero-shot manner, the opposite direction is needed, as there is a need for higher precision.

Sun and Wang (2022) propose a method for a precision-recall trade-off that requires beam-search decoding, which increases computational resources and inference time compared to greedy decoding. To overcome this issue, we propose a novel training schedule method to control the precision-recall trade-off during training instead of inference. Our method allows for the application of standard greedy decoding during inference without the need

¹github.com/richardxoldman/llms-for-minimal-gec

...to a cafe and ~~and~~ I drank a drink.
 I recommend you **to** practise any sport...
 She is **one** of the ones that...
 Sometimes we go to partyies in the city.
 ...and I was very **happy** to hug him because I miss
 him...

Table 1: Examples of changes in target texts made during detokenization process by the Llama 3 70b model. Deletions are highlighted with a strikethrough, and insertions are highlighted in bold.

for external tools or algorithms to control the inference process.

Since LLMs are trained on raw texts and existing GEC datasets are available in word-tokenized (henceforth referred to as "tokenized") format (Bryant et al., 2023), it forces models to switch from working on raw texts to tokenized texts.

Another case that would require detokenized texts is any work that leverages probability distributions for language models, for example the Scribendi Score reference-less metric (Islam and Magnani, 2021).

To solve this issue, we detokenize the most common GEC datasets and verify whether training models on detokenized texts leads to better results. The detokenization process involved the usage of the LLM, during which we discovered that even the most popular datasets contain errors in annotations. We make the detokenized datasets available to the public to make them accessible to other researchers².

In summary, our contributions in this work are as follows:

- The LLM that achieves the state-of-the-art single-model system on the BEA-19 Shared Task test set.
- The study of error rate adaptation in the context of LLMs.
- The novel training schedule method that enables control of the precision-recall trade-off during training.
- The detokenization of the most common English GEC datasets, and the detailed analysis of annotation errors in them.

2 Datasets and their detokenization

The most common GEC datasets for English are available in a tokenized format due to evaluation tools that use the M2 format (Dahlmeier and Ng, 2012) such as ERRANT (Bryant et al., 2017). LLMs are trained on raw texts, so the tokenization process forces them to switch to the tokenized text and also to learn the tokenization process. To solve this issue, we detokenize FCE-train (Yanakoudakis et al., 2011), W&I+LOCNESS train and dev part (hereafter, we refer to the train split of this dataset as BEA-train, the dev split as BEA-dev and the test split as BEA-test) (Bryant et al., 2019) CoNLL-2014-test (Ng et al., 2014), and JFLEG datasets — these are the datasets we decided to use in our work, as they are one of the most commonly used GEC resources (Bryant et al., 2023). The statistics about them are given in the Appendix.

For the FCE-train, BEA-train, and BEA-dev datasets, the source texts were available in the raw format (the only work needed was to properly split them line by line). To detokenize the target texts of these datasets, we used the Sacremoses Detokenizer³, but it did not correctly detokenize all the examples.

To improve the detokenization process, we leveraged the Llama-3.1-70b-Instruct model (denoted as Llama 3 70b), where the model task was only to detokenize the target text. We included a source text that is properly detokenized in the prompt to help the model in the detokenization process. The prompt is given in the Appendix.

In order to detokenize the CoNLL-2014 input texts, we had to properly split paragraphs at the sentence level, which are available in SGML format. We did this using a simple Python script with split rules and then manually adjusted examples that were not properly handled by the script.

For the JFLEG dataset we only had to detokenize inputs of the dataset, since the dataset has only dev and test splits. Due to the small size of the JFLEG dataset, we used Sacremoses Detokenizer and then manually adjusted the texts.

It should be emphasized that our work does not affect the examples in the test sets. The source texts for both the BEA-test and the CoNLL-2014-test were unchanged. The BEA-test target texts are hidden on the CodaLab platform and are not available publicly. There was no need to detokenize

²github.com/richardxoldman/detokenized-gec-datasets

³pypi.org/project/sacremoses/

Dataset	modified	essential	optional	erroneous	not assessable	wrong annotations (estimated lower bound)
BEA-dev	6.52%	80.77%	2.80%	12.59%	3.85%	5.22%
BEA-train	6.22%	78.67%	4.90%	9.80%	6.64%	4.89%
FCE-train	8.42%	71.68%	12.24%	12.24%	3.85%	6.04%

Table 2: Details for annotations to examples changed by the Llama 3 70b model.

the CoNLL-2014-test target texts, since the scoring script uses the M2 format to compute the results. It makes outcomes based on our version of the datasets fully comparable to the previous research.

The results reported on our version of the BEA-dev dataset may differ slightly from those reported by other researchers due to the changes described in Section 2.1, but are intended to select the most promising model, not to report the final results.

2.1 Incorrect annotations in datasets

In less than 10% of the examples, the Llama 3 70b model, when used for detokenization, occasionally modified the text beyond simply removing spaces in the correct version of the text. Table 1 shows examples of differences between the target texts in the dataset and the changes made by the Llama 3 70b model. Our initial investigation showed that those changes are mostly errors that were not corrected by a human annotator. Given this, we decided to do a manual annotation of such samples.

For our annotation purposes, the considered sentences were assigned four labels: *essential*, *optional*, *erroneous* and *not assessable*.

The *essential* label was assigned to sentences in which corrections were necessary and actually contributed to improving their accuracy.

The *optional* label was attributed to sentences in which the corrections made were not necessary, as their original versions were considered correct as well (e.g. sentences originally written in capital letters, which were then changed to lower case).

The *erroneous* label refers to situations where the corrections either do not fix the original mistakes in the sentences or create new mistakes in sentences that were already correct.

Finally, the *not assessable* label is used to mark corrections for which the quality, for various reasons, cannot be assessed by the annotator.

For BEA-dev, all examples (284) modified by the Llama 3 70b model were verified, whereas for

the other two datasets, random samples of the same size (284 examples) were checked. The results of the annotation process are shown in Table 2.

2.2 Detokenization impact

To verify whether the detokenization process and the modification of examples by the Llama 3 70b model have an impact on the GEC models, we decided to train the LLMs on the FCE-train and the BEA-train datasets in four different processing setups:

1. **detokenized-filtered**: Detokenized datasets **excluding** examples modified by the Llama 3 70b model.
2. **tokenized-filtered**: Tokenized datasets corresponding to the examples that remained unmodified in the detokenized version.
3. **detokenized-full**: Detokenized datasets **including** all examples, both modified and unmodified.
4. **tokenized-full**: Tokenized datasets corresponding to the full set of detokenized examples (original, untouched datasets).

Please note that **tokenized-*** setups refer to the original examples "as is", without any modifications introduced by the Llama 3 70b model.

The **detokenized-filtered** setup compared to the **tokenized-filtered** setup shows whether the detokenization process has an impact on the models' performance, since both models are fine-tuned on the same examples with the same hyperparameter setup. The details about the hyperparameters are given in the Appendix.

The ***-full** setups against the ***-filtered** setups show whether the changes made by the Llama 3 70b model in the datasets have an impact on the results, because the **detokenized-full** setup contains the modified examples by the Llama 3 70b model,

Model	Size	Setup	BEA-dev			JFLEG-dev	
			P	R	F _{0.5}	GLEU	
Qwen 2.5	1.5B	detokenized-filtered	57.90	42.10	53.86	56.10	
Qwen 2.5	1.5B	tokenized-filtered	59.00	38.48	53.31	56.17	
Qwen 2.5	1.5B	detokenized-full	57.86	42.75	54.04	56.22	
Qwen 2.5	1.5B	tokenized-full	59.92	37.79	53.63	56.01	
Llama 3 Small	3B	detokenized-filtered	63.34	47.52	59.39	57.42	
Llama 3 Small	3B	tokenized-filtered	63.31	47.29	59.29	57.58	
Llama 3 Small	3B	detokenized-full	63.04	48.32	59.42	57.56	
Llama 3 Small	3B	tokenized-full	62.61	46.22	58.46	56.96	
Gemma 2	9B	detokenized-filtered	68.84	56.40	65.93	58.70	
Gemma 2	9B	tokenized-filtered	68.84	55.90	65.79	58.99	
Gemma 2	9B	detokenized-full	69.07	57.13	66.30	58.72	
Gemma 2	9B	tokenized-full	69.86	55.67	66.47	58.40	

Table 3: Results for different dataset processing setups.

Dataset	M	R	U
BEA-dev	50.74%	38.87%	10.39%
BEA-train	46.93%	40.28%	12.79%
FCE-train	61.33%	31.96%	6.71%

Table 4: Details about the operations performed by the Llama 3 70b model. The labels stand for: Missing, Replacement and Unnecessary.

whereas the **tokenized-full** setup contains all the original examples (also the erroneous ones). Again, the number of training examples is the same, but the difference lies in the quality of the annotations in examples that were changed by the Llama 3 70b model.

All models were trained for one epoch on the FCE-train dataset and then for one epoch on the BEA-train dataset. In this and subsequent experiments, we report the results for the BEA-dev and JFLEG-dev datasets, since these datasets give a view for both minimal-edit and fluency-edit GEC. Table 3 presents the results for 3 different LLMs of different sizes: Qwen2.5-1.5B-Instruct (denoted as Qwen 2.5), Llama-3.2-3B-Instruct (denoted as Llama 3 Small) and gemma-2-9b-it (denoted as Gemma 2).

2.3 Results analysis

The results show that LLMs can learn the tokenized version of the texts and in some cases even achieve better metric scores compared to the models trained on the detokenized texts. We can see that there are no clear gains in terms of F_{0.5} score from using the

detokenized version of datasets.

The transition from the **tokenized-filtered** to the **tokenized-full** setup increases precision in each experiment but lowers recall and GLEU values. In all cases, transition from the **detokenized-filtered** setup to the **detokenized-full** setup improves recall and slightly improves the GLEU score. It shows that the changes made by the Llama 3 70b model result in outputs with higher linguistic freedom, which is expected, since the most common change made by the Llama 3 70b model is the Missing operation (Table 4), while using the original sentences makes the models produce more strict outputs.

We can also see that the size of the models significantly impacts the results. Therefore, for the next experiments we will further explore the Gemma 2 model, as it is the best performing model. Although Gemma 2 achieves the best F_{0.5} score on the **tokenized-full** setup, the next experiments will be performed on the detokenized version of the datasets, as they contain corrected erroneous annotations. The other reason is that our systems can be used in the work of other researchers who need a model that produces detokenized output. It would be also simply practical in terms of using the system in the environment where the output does not require removing the unnecessary spaces.

3 Overcorrection problem

In the minimal-edit GEC task, the goal is to find and correct only those parts of the texts that are clearly erroneous, without making further improvements to their fluency. Due to the pre-training goal

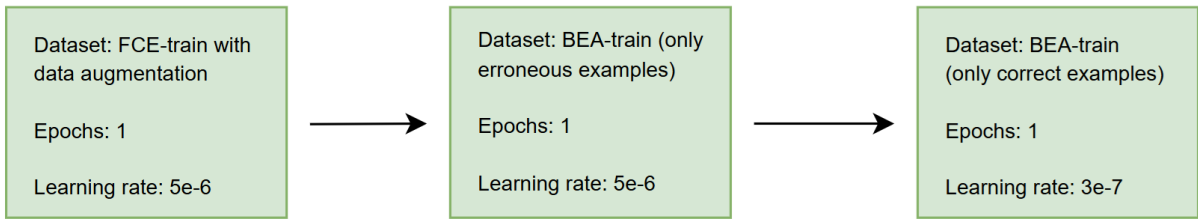


Figure 1: Visualization of the fine-tuning process for our best performing Gemma 2 model on the BEA-dev dataset.

of LLMs, which is to maximize the probability of the next token, and the flexibility they gain from instruction fine-tuning process, LLMs tend to produce more fluent output. While this characteristic may be advantageous for fluency-edit GEC, the objective of minimal-edit GEC is to apply only the minimal necessary corrections.

Standard minimal-edit GEC benchmarks, which are based on texts written by English language learners, put a greater weight on precision than on recall, because suggesting an incorrect change is considered more negative than ignoring an error (Ng et al., 2014). Therefore, a proper adaptation of the model is needed to correct errors with high precision.

For the Chinese minimal-edit GEC, Yang and Quan (2024) proposes an alignment model which is used to filter only minimal corrections from the initial correction, which may be fluent.

One of the most recent works proposes the novel method for LLM fine-tuning, Edit-Wise Preference Optimization (EPO) that fits the minimal-edit GEC task better than the standard supervised fine-tuning (SFT) approach (Liang et al., 2025). In our work, we explore the SFT approach with a focus on the datasets rather than the different training approaches, and show that proper data preprocessing or training schedule can lead to the successful minimal-edit LLM model.

4 Data augmentation

During GEC model fine-tuning, datasets play a crucial role in the whole process. One of the most important attributes of the GEC datasets is the error rate. The common practice for neural models that are trained from scratch is to remove unedited pairs (Chollampatt and Ng, 2018; Kiyono et al., 2019), because for these models there is a need for improved recall.

Large language models produce fluent output with high recall, which may suggest that removing unedited pairs for LLMs is unnecessary and could

worsen the results. Furthermore, it may be possible that providing additional unedited pairs could improve minimal-edit error correction for LLMs.

To provide more real examples that may not be fluent, but are still acceptable, we propose a data augmentation method to split each example (consisting of source text and corrected text) into two pairs. The new pair is created by using the corrected text as both the source and target text. For example, the sentence pair “Alice have a cat.” and “Alice has a cat.” can be split into the following examples:

- Alice have a cat. → Alice has a cat.
- Alice has a cat. → Alice has a cat.

Our method can be applied to any dataset and does not require any additional models/tools to extend a given dataset.

5 Training schedule

Current approaches to GEC training scheduling consist of dividing data into 2 or 3 groups based on data quality and then training a model in the correct order, from the lowest quality data to the highest (Bout et al., 2023). We follow this approach, but to control the precision-recall trade-off, we propose to extend it even further.

In the final stage (with the highest quality dataset – in our case it is BEA-train dataset), we split the data into two groups. The first group contains only erroneous texts, whereas the second group contains only correct examples. During the stage, we first train the model on the first group (only erroneous examples), and then we train the model on the second group (only correct examples) with lower learning rate. Figure 1 shows the step-by-step training schedule for the best performing model on the BEA-dev set.

Our intuition behind this approach is that a model first learns how to correct errors and later is tuned to understand which examples are correct but

Dataset processing approach	Erroneous sentences		BEA-dev		JFLEG-dev	
	FCE-train	BEA-train	P	R	F _{0.5}	GLEU
ONLY-ERRONEOUS	100%	100%	60.74	58.79	60.34	58.16
UNCHANGED	65.43%	69.02%	68.99	57.12	66.24	58.73
AUGMENTED	39.55%	40.83%	71.42	53.42	66.92	58.21

Table 5: Results for the dev sets for the experiment with our data augmentation method.

in some cases not perfectly fluent. During the last stage, when the model is fine-tuned only on correct examples, the model only learns to not apply corrections to texts.

Choosing a proper learning rate value (or number of examples) enables controlling the precision-recall trade-off in LLMs, as lowering learning rate should make the model learn not to correct more smoothly while still being able to correct the errors in texts.

6 Experiments

6.1 Data augmentation experiments

To test whether the addition of unedited pairs can positively affect LLMs in the minimal-edit GEC task, we train the Gemma 2 model⁴ with the same hyperparameter setup as in the experiment from Section 2 in three different dataset processing approaches:

- only erroneous examples (denoted as **ONLY-ERRONEOUS**)
- erroneous examples + unedited examples (denoted as **UNCHANGED**)
- erroneous examples + unedited examples + unedited examples created from erroneous examples by applying our data augmentation method (denoted as **AUGMENTED**)

As in the previous experiment, we first train one epoch on the FCE-train dataset and then one epoch on the BEA-train dataset.

Table 5 shows the results on the BEA-dev and JFLEG-dev datasets. We can see that unedited examples are needed to improve the LLMs performance. Even on the fluency-edit dataset, the scores are better when unedited pairs are added to

⁴For the data augmentation and training schedule experiments we also tested the gemma-2-9b-it-SimPO model and achieved slightly better results, but we decided to use the original Gemma 2 model as our goal is not to maximize the benchmark scores.

the dataset (the **UNCHANGED** approach). For the **AUGMENTED** approach, the F_{0.5} score is the highest among all approaches, but the GLEU score is lower compared to the **UNCHANGED** approach.

This study shows that lowering the error rate in the GEC datasets is a way to make LLMs produce minimal-edit outputs. It also shows that when new solutions are available, such as modern LLMs, approaches or practices from previous research, such as removing unedited pairs, should be reevaluated and tested again.

6.2 Training schedule experiment

We also carried out an experiment with different learning rate values for the last group (only correct examples) for our training schedule method for the Gemma 2 model. We also test whether applying our data augmentation method for the FCE-train dataset improves the results.

Note that in this experiment data augmentation method is **not** applied to the BEA-train dataset.

Table 6 shows how the precision-recall trade-off depends on the learning rate value. It can be observed that even small changes in the learning rate value noticeably influence the trade-off, making the hyperparameter very sensitive.

When applying the data augmentation method for the FCE-train dataset, the BEA-dev set F_{0.5} score can be improved compared to the best value achieved in the previous experiment (the **AUGMENTED** dataset processing approach).

Although the data augmentation method was designed to enhance precision, we observe that results with data augmentation on the FCE-train have higher recall. In this experiment, we hypothesize that training on the FCE-train provides general GEC knowledge, while fine-tuning on the BEA-train determines the model’s behavior in terms of the precision-recall trade-off as model is first fine-tuned on erroneous examples and then on the correct ones.

Learning rate	FCE-train Augmented	BEA-dev			JFLEG-dev	
		P	R	F _{0.5}	GLEU	
1e-7	✗	65.90	58.18	64.19	58.58	
1e-7	✓	65.10	58.33	63.62	58.60	
2e-7	✗	69.30	56.05	66.17	58.64	
2e-7	✓	69.22	56.40	66.21	58.66	
2.5e-7	✗	70.94	53.73	66.67	58.47	
2.5e-7	✓	70.96	54.40	66.89	58.28	
3e-7	✗	73.63	48.72	66.80	57.60	
3e-7	✓	73.52	50.10	67.23	57.90	
3.5e-7	✗	75.81	44.92	66.65	56.74	
3.5e-7	✓	75.38	46.82	67.18	57.35	
4e-7	✗	77.49	40.15	65.34	55.48	
4e-7	✓	76.74	43.49	66.57	56.15	
5e-7	✗	79.74	24.79	55.29	50.26	
5e-7	✓	78.88	31.78	60.85	52.91	

Table 6: Results for the dev sets for the experiment with our training schedule method.

Figure 1 shows the complete training process for the model with the highest F_{0.5} score.

6.3 Results on the test datasets

From each experiment, we choose the most promising model based on its performance on the BEA-dev dataset to evaluate it on the BEA-test, CoNLL-2014-test, and JFLEG-test datasets. In Table 7, Gemma 2 Augmentation refers to the best model from Section 6.1 (only applying the data augmentation method) and Gemma 2 Training-Schedule refers to the best model from Section 6.2.

Table 7 shows that our model from the training-schedule experiment achieves a new state-of-the-art single model result on the BEA-test dataset and has competitive results with other solutions on the CoNLL-2014-test dataset. It should be noted that our models were trained only on two relatively small datasets, whereas other solutions were trained on a much larger number of examples, except for the Mistral-7b-EPO model.

To get more insights about the impact of the different model selection on the results, we also performed a single experiment with the gemma-2-27b-it and llama-2-13b-chat (Gemma 2 (27b) Training-Schedule and LLama-2-13b Training-Schedule in the tables) models with the same training schedule and hyperparameters as the best performing model on the BEA-dev dataset, so the model training is exactly the same as for the Gemma 2 Training-Schedule model.

The Llama-2-13b achieves even worse results than these reported by (Omelianchuk et al., 2024). It can be explained by using different datasets during fine-tuning process. The precision and recall are both worse than those of the Gemma 2 model. This suggests that model size is not the only important factor; other details about the LLM, such as its novelty, architecture, and the dataset used for training, also matter.

The Gemma 2 (27b) achieves even a better score than the best Gemma 2 9b model on the BEA-test set, but it may be slightly overtuned for precision due to the same learning rate value in the final stage with the bigger model, which can be observed in the worse results for the CoNLL-2014-test dataset.

Table 8 shows the results for the JFLEG-test dataset. We can see that even if our models are fine-tuned for minimal-edit GEC, they achieve a higher score than the average of the scores computed for the JFLEG-test references. It suggests that LLMs can find a proper balance between minimal-edit GEC and fluency-edit GEC.

7 Conclusions

Our work demonstrates that there are several ways to fine-tune an LLM for minimal-edit grammatical error correction, without the need for pre-training them on a large number of examples. We propose easy-to-implement methods for controlling the precision-recall trade-off during fine-tuning.

Moreover, we show that choosing a more recent

Model	Size	CoNLL-2014-test			BEA-test		
		P	R	F _{0.5}	P	R	F _{0.5}
T5 Large (Rothe et al., 2021)	700M	-	-	66.04	-	-	72.06
T5 XL (Rothe et al., 2021)	3B	-	-	67.65	-	-	73.92
T5 XXL (Rothe et al., 2021)	11B	-	-	68.75	-	-	75.88
GECToR (Tarnavskiy et al., 2022)	355M	74.40	41.05	64.00	80.70	53.39	73.21
TemplateGEC (Li et al., 2023)	770M	74.80	50.00	68.10	76.80	64.80	74.10
FLAN-T5 XXL (Cao et al., 2023)	11B	75.00	53.80	69.60	78.80	68.50	76.50
DeCoGLM (Li and Wang, 2024)	335M	75.10	49.40	68.00	77.40	64.60	74.40
BART Base (Wang et al., 2024)	400M	76.20	52.20	69.80	77.70	67.50	75.40
Llama-2-13b (Omelianchuk et al., 2024)	13B	77.30	45.60	67.90	74.60	67.80	73.10
Mistral-7b-EPO (Liang et al., 2025)	7B	76.71	52.56	70.26	78.16	68.07	75.91
Gemma 2 Augmentation	9B	73.80	56.16	69.43	74.86	71.35	74.13
Gemma 2 Training-Schedule	9B	75.74	51.47	69.24	79.87	68.90	77.41
Llama-2-13b Training-Schedule	13B	71.07	50.11	65.59	74.10	67.54	72.69
Gemma 2 (27b) Training-Schedule	27B	77.38	47.88	68.89	82.28	67.03	78.70

Table 7: Single model results for the minimal-edit GEC test sets.

Model	GLEU
Source (Uncorrected)	40.54
Reference (Average)	62.37
Conv Seq2Seq (Ge et al., 2018)	62.42
Transformer (Stahlberg and Kumar, 2021)	64.70
GPT-3.5 (Coyne et al., 2023)	63.40
GPT-4 (Coyne et al., 2023)	65.02
Gemma 2 Augmentation	63.72
Gemma 2 Training-Schedule	62.91
Llama-2-13b Training-Schedule	62.53
Gemma 2 (27b) Training-Schedule	62.42

Table 8: Results for the fluency-edit GEC dataset (JFLEG-test).

LLM is also an important factor that impacts the overall performance of the model. The Gemma 2 9b model, even as a smaller model achieved much better performance compared to the Llama-2-13b model.

The detokenization process did not improve model performance, but our findings on the errors in the most common GEC datasets show the need for a proper curation of datasets. Our work also shows that LLMs can be effectively used as a detokenization tool.

8 Limitations

Our work covers only experiments on English GEC datasets, so it would be beneficial to extend the re-

search to check how LLMs would perform in other languages. We did not conduct experiments on other types of models. It is hard to tell whether our methods would improve the Seq2Seq or Seq2Edit approaches.

The other issue is that we applied only greedy decoding during inference. The results could be even better if different decoding methods were applied. It would also be worth comparing these methods applied in LLMs with the Seq2Seq or Seq2Edit models.

The reusability of the training schedule method is limited by the requirement for extensive learning rate tuning for any different model or dataset due to high sensitivity to minor changes in learning rate.

Obtaining the highest F_{0.5} might be considered overfitting for a specific test set and evaluation metric, but in practical terms, the style of grammar correction depends on specific needs, guidelines, etc., so this might be a desired behavior.

Lastly, running our models requires a lot of memory and computational power, so for many people it would be impossible to run them on their devices. Our models may not be practical for everyday use, but they can be used to create synthetic datasets that can be used to train smaller models.

References

Andrey Bout, Alexander Podolskiy, Sergey Nikolenko, and Irina Piontkovskaya. 2023. [Efficient grammatical error correction via multi-task training and op-](#)

- timized training schedule. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5800–5816, Singapore. 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.
- 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.
- Hannan Cao, Liping Yuan, Yuchen Zhang, and Hwee Tou Ng. 2023. **Unsupervised grammatical error correction rivaling supervised methods**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3072–3088, Singapore. Association for Computational Linguistics.
- Shamil Chollampatt and Hwee Tou Ng. 2018. A multi-layer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI’18/IAAI’18/EAAI’18. AAAI Press.
- 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, O 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 and virtual meeting. 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**. *Preprint*, arXiv:2304.01746.
- Tao Ge, Furu Wei, and Ming Zhou. 2018. **Reaching human-level performance in automatic grammatical error correction: An empirical study**. *Preprint*, arXiv:1807.01270.
- Md Asadul Islam and Enrico Magnani. 2021. **Is this the end of the gold standard? a straightforward referenceless 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.
- 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.
- 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.
- 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.
- Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan Huang, and Min Zhang. 2023. **TemplateGEC: Improving grammatical error correction with detection template**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6878–6892, Toronto, Canada. Association for Computational Linguistics.
- 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.
- Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfali, 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, 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 Skurzhanyskyi. 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 Skurzhanyskyi, 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.
- 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. 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.
- Xin Sun and Houfeng Wang. 2022. **Adjusting the precision-recall trade-off with align-and-predict decoding for grammatical error correction**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 686–693, Dublin, Ireland. Association for Computational Linguistics.
- Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. **Ensembling and knowledge distilling of large sequence taggers for grammatical error correction**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.
- 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 and virtual meeting. 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.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. **A new dataset and method for automatically grading ESOL texts**. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

A Training details

We trained our ≤ 13 b models on a 2xA100 (80GB) GPU setup and the 27b model on a 4xA100 (80GB) GPU setup. We used 4xA100 (80GB) GPU setup to run the Llama 3 70b model for the detokenization process. A single model training took 2-3 hours. The hyperparameter values are described in Table 10. The following prompt was used during training our models and during inference:

Correct the following text, making only minimal changes where necessary.

Text to correct:

<source text>

Corrected text:

<target text>

B Detokenization prompt

The following prompt was used to detokenize the datasets:

You will receive two texts: source text and corrected text. Corrected text may not have proper spaces. Your task is to remove/add proper spaces to the corrected text. Do not write any comments, just write corrected text with proper spaces.

Source text: <source text>

Corrected text: <target text>

Only change spaces, you must not change punctuation.

Hyperparameter name	Value
learning rate	5e-6
batch size	4
gradient accumulation steps	4
warmup steps (for each dataset)	100
lr scheduler	linear
epochs (for each dataset)	1
optimizer	AdamW8bit
weight decay	0.01

Table 10: Hyperparameter values used to train our models.

Dataset	#Examples	Erroneous sentences
FCE-Train	28.4k	65.43%
BEA-train	34.3k	69.02%
BEA-test	4.5k	–
BEA-dev	4.4k	67.36%
CoNLL-2014-test	1.3k	71.90%
JFLEG-dev	754	95.36%
JFLEG-test	747	95.31%

Table 9: Details of the datasets used in our work. Note that there ratio of erroneous sentences could be different compared to the statistics about the datasets from different research works due to the changes made by the Llama 3 70b model during the detokenization process.

COGENT: A Curriculum-oriented Framework for Generating Grade-appropriate Educational Content

Zhengyuan Liu^{✧*}, Stella Xin Yin^{✧*}, Dion Hoe-Lian Goh[✧], Nancy F. Chen[✧]

[✧]Nanyang Technological University, Singapore

[✧]Institute for Infocomm Research (I²R), A*STAR, Singapore

{liu_zhengyuan, nfychen}@i2r.a-star.edu.sg

Abstract

While Generative AI has demonstrated strong potential and versatility in content generation, its application to educational contexts presents several challenges. Models often fail to align with curriculum standards and maintain grade-appropriate reading levels consistently. Furthermore, STEM education poses additional challenges in balancing scientific explanations with everyday language when introducing complex and abstract ideas and phenomena to younger students. In this work, we propose COGENT, a curriculum-oriented framework for generating grade-appropriate educational content. We incorporate three curriculum components (science concepts, core ideas, and learning objectives), control readability through length, vocabulary, and sentence complexity, and adopt a “wonder-based” approach to increase student engagement and interest. We conduct a multi-dimensional evaluation via both LLM-as-a-judge and human expert analysis. Experimental results show that COGENT consistently produces grade-appropriate passages that are comparable or superior to human references. Our work establishes a viable approach for scaling adaptive and high-quality learning resources.

1 Introduction

Educational content, particularly reading materials, is considered an integral part of supporting effective learning across disciplines. Traditionally, the creation of educational materials has relied mainly on human authors. This limits scalability and adaptability when curriculum standards evolve or when diverse learning needs must be addressed at scale. Generative AI techniques, such as Large Language Models (LLMs), have demonstrated remarkable potential in various content generation (Achiam et al., 2023; Team et al., 2024). However, their application to educational contexts presents several

challenges. While models can generate grammatically correct and coherent passages, they often fail to align with established curriculum standards (Xiao et al., 2023; Liu et al., 2024b). Moreover, it is difficult to maintain consistent grade-appropriate reading levels, as both sentence structure and vocabulary complexity impact student comprehension and learning outcomes (Zamanian and Heydari, 2012). STEM education poses an additional challenge of balance between science and everyday language when introducing complex and abstract concepts to younger students (Blown and Bryce, 2017; Gilbert and Byers, 2017). Therefore, creating materials that effectively bridge science terminologies with real-world examples while maintaining pedagogical value requires professional knowledge and multi-dimensional efforts (Bansiong, 2019).

To address these problems, here we propose a framework **Curriculum-Oriented Generation for Educational Content** (COGENT), which creates science reading materials aligned with curriculum standards and adapts to grade-specific readability requirements. This framework consists of three components: curriculum formulation, controllable content generation, and multi-dimensional evaluation. Grounded in well-established education standards such as the Next Generation Science Standards (NGSS) (States, 2013), we build the structured guidance by linking science concepts (e.g., grades 1-5) with core ideas and their corresponding learning objectives, which creates systematic alignment with pedagogical value. For readability control, we implement constraints on word number, vocabulary, and sentence complexity based on grade-level reading proficiency (Flesch, 1948). Further, inspired by inquiry-based learning (Dewey, 1986), we incorporate a “wonder-based” learning approach that transforms core scientific ideas into inquiry-driven topics to engage students with science learning and discovery.

To comprehensively evaluate our framework and

* Equal contribution.

its pedagogical effectiveness, we build a multi-dimensional validation protocol and conduct quantitative analyses of the generated content across curriculum alignment, comprehensibility, and readability metrics. Based on the COGENT framework, our experiments with three representative LLMs (Gemma-2-9B, GPT-4o, Claude-3.5-Sonnet) indicate that: (1) models can follow curriculum guidance to create educational content that aligns closely with established pedagogical standards; (2) models not only maintain high comprehensibility but also demonstrate adaptability in adjusting length, vocabulary, and sentence complexity to meet grade-specific reading requirements. The findings suggest that with proper scaffolding and constraint mechanisms, LLM-based systems can serve as a complement to human expertise in educational content development, which enables access to high-quality, curriculum-aligned reading materials across diverse educational contexts. This work not only advances our understanding of how to effectively harness models for educational purposes but also establishes a foundation for future investigations into automated content generation, with broader applications for personalized learning.

2 Related Work

2.1 AI-generated Content in Education

Advancements in LLMs have accelerated the adoption of AI in educational contexts, particularly in automating traditionally time-consuming content generation tasks such as providing feedback, creating assessment materials, and generating learning recommendations (Yan et al., 2024; Liu et al., 2024b,c). These efforts provide customized learning materials to students based on individual factors such as learning status, preferences, and goals (Wang et al., 2024; Liu et al., 2024a). For example, Kuo et al. (2023) demonstrated how to generate dynamic learning paths for students based on their most recent knowledge mastery assessment results. Similarly, Kabir and Lin (2023) enhances content generation by incorporating knowledge concept structures throughout the process. While these methods show promise, they mainly focus on students' own learning trajectories and knowledge structures, with little attention given to standardized curriculum frameworks. Additionally, the generated content often fails to appropriately differentiate reading levels.

To evaluate LLM-generated content, researchers

combined automatic and expert analysis. For instance, Lee et al. (2024) investigated LLMs' capability in generating test questions, with both automatic evaluation and expert analysis confirming that these models can produce questions with high validity and reliability for language learning. Similarly, Zelikman et al. (2023) developed a reading comprehension exercise generation system for middle school English learners, demonstrating that AI-generated materials can not only meet students' learning needs but, in some cases, surpass the quality of human-written materials. In computer science education, Lee and Song (2024) examined the effectiveness of AI-generated content in explaining programming concepts, further validating the potential of LLMs in educational content creation.

While current evaluation of AI-generated content focuses mainly on language and facts (Xiao et al., 2023), real-world educational assessment requires broader criteria including curriculum alignment, pedagogical scaffolding, and grade-level appropriateness (Bansiong, 2019; Berndt and P. Wayland, 2014). This lack of comprehensive evaluation standards hinders educators' interest and trust in implementing AI-generated resources.

2.2 Evaluation Metrics of Education Materials

The evaluation of educational content includes three aspects: readability, comprehensibility, and curriculum alignment. These factors collectively determine whether learning materials are "appropriate to the student's age and level of knowledge" and "prepared in line with the curricula."

Comprehensibility and *Readability* serve as fundamental metrics in analyzing educational texts (Zamanian and Heydari, 2012). Readability is a textual characteristic that measures how easily text can be read and understood (Klare, 1974), while comprehensibility reflects how effectively readers can construct meaning from the text (Sadoski et al., 2000; Beck et al., 1991). As Lakoff and Johnson (1980) emphasizes, "understanding is only possible through the negotiation of meaning." When these aspects are misaligned, students may experience frustration or disengagement (Bansiong, 2019).

Curriculum alignment aims to ensure it meets educational standards while remaining appropriate for learners' grade levels (Anderson, 2002). This evaluation ensures that educational materials are not only readable and comprehensible but also serve their intended pedagogical purposes within the edu-

cational framework (Squires, 2012; Wijngaards-de Meij and Merx, 2018).

2.3 Value of “Wonder” in Science Education

“The most beautiful thing we can experience is the mysterious. It is the source of all true art and science.” (Einstein, 1931)

Inquiry-based learning is rooted in the work of Dewey (1986), who underlines that education begins with the curiosity of the learner. Inquiry is understood in two ways: (1) “inquiry as means” (inquiry in science) refers to using inquiry as an instructional approach to help students develop their understanding of science content; (2) “inquiry as ends” (inquiry about science) refers to inquiry as a learning outcome (National Research Council, 2000; Abd-El-Khalick et al., 2004). However, when students inquire about scientific knowledge, they often experience a gap between their intuitive comprehension and their ability to express understanding (Blown and Bryce, 2017). They frequently struggle to express their observations and questions using scientific language. This disconnect highlights the need for level-appropriate educational content that can bridge the gap between students’ intuitive understanding and formal scientific language. Given this challenge, it is recommended to introduce scientific concepts through “wonder why” questions that trigger children’s natural curiosity while reducing the barriers of science terminologies (Chin and Brown, 2002; Gilbert and Byers, 2017). Moreover, wonder-based explanatory texts are effective for reading comprehension, science learning, and conceptual change (Lindholm, 2018; Jirout, 2020).

3 Curriculum-Oriented Generation for Educational Content

The framework is designed to transform abstract curriculum components into engaging, wonder-based reading materials that improve students’ understanding while adhering to grade-specific readability requirements. It consists of three parts: curriculum formulation, controllable content generation, and multi-aspect evaluation (see Figure 1).

3.1 COGENT-based Generation

To simulate human teachers and editors (Bybee, 2014), we incorporate structured curriculum information to guide LLM-based educational content generation, ensuring pedagogical alignment, development progress, and topic coverage. Here, we

Level	Avg. words	Avg. lexile	Avg. unique words
Grade 1 (Ages 6-7)	101	430	57.9
Grade 2 (Ages 7-8)	200	545	87.7
Grade 3 (Ages 8-9)	319	605	132.8
Grade 4 (Ages 9-10)	468	770	183.2
Grade 5 (Ages 10-11)	558	920	219.5

Table 1: Linguistic features of human-written science reading passages at elementary grade levels.

ground our approach in the Next Generation Science Standards (NGSS), a well-established K-12 science education framework (States, 2013).¹ We decompose the curriculum into three hierarchical elements: science concepts, core ideas, and learning objectives. As shown in Figure 2, science concepts can be mapped to core ideas, and each core idea is related to learning outcomes, creating a comprehensive curriculum coverage matrix. More specifically, for elementary school students (grades 1-5, ages 6-11), 29 science concepts (e.g., “Matter and Its Interactions”) are broken down into 79 core ideas (e.g., “Structure and Properties of Matter. Matter can be described and classified by its observable properties.”), then further mapped to specific learning outcomes that detail what students should master at each grade level (e.g., “To describe and classify different kinds of materials by their observable properties (Grade 2).”).

Importantly, concepts and core ideas can appear across multiple grade levels, requiring different depths of explanation and language complexity (see Figure 2). As shown in Table 1, human-written science reading passages show clear patterns across grade levels: the average number of words, reading difficulty scores (lexile) (White and Clement, 2001), and lexical diversity all increase steadily as students progress from grade 1 to grade 5. We thus indicate the word number and target readability level (Klare, 1974; Flesch, 1948)² along with the curriculum input to ensure generated content matches students’ reading abilities at each grade.

Moreover, to enhance students’ interest and engagement, we consider “Science as Wonder” and “everyday language” as a bridge to connect scien-

¹While we demonstrate our framework using NGSS as a representative example in this paper, the hierarchical decomposition underlying COGENT can be adapted to other national education frameworks and subjects, such as the National Curriculum in England (Department for Education, 2014) or Singapore’s Ministry of Education curriculum standards (Ministry of Education Singapore, 2023).

²In our experiments, based on human-written passages, we set the word count to be the grade level multiplied by 100. Flesch Kincaid Grade Level is used for readability control.

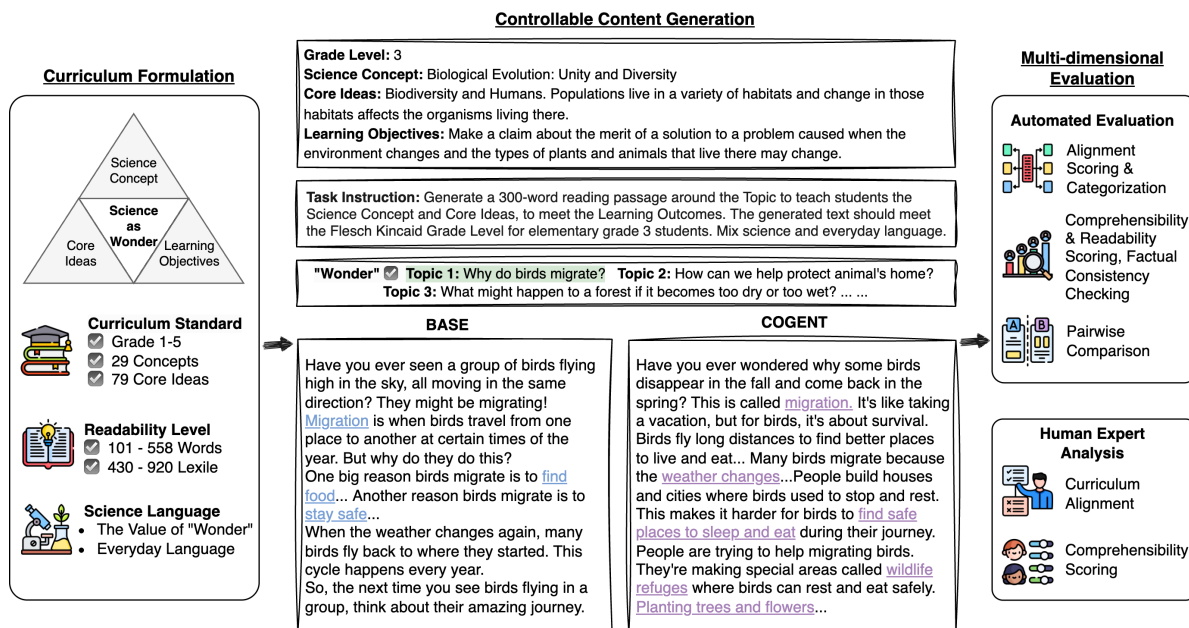


Figure 1: Overview of the framework of curriculum-oriented generation for educational content (COGENT).

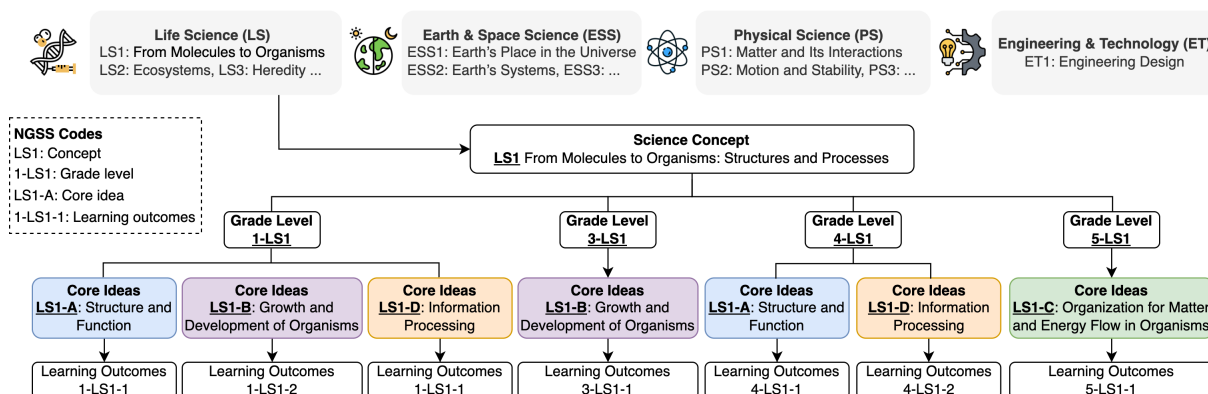


Figure 2: Our curriculum decomposition example grounded in the Next Generation Science Standards (NGSS), which consist of four domains. It has a hierarchical structure where Science Concepts (e.g., LS1) branch into Core Ideas (e.g., LS1-A: Structure and Function), which then connect to specific Learning Outcomes for each grade level (e.g., 1-LS1-1). The same core idea may appear across multiple levels with increasing complexity. For example, LS1-A (Structure and Function) progresses from grade 1 to grade 4.

tific concepts and their daily experiences. Given the decomposed curriculum items, each core idea can be used to generate multiple exploration questions. For example, the core idea about environmental adaptation can be linked to wonder topics such as “Why do birds migrate?” or “How can we help protect animals’ home?” This approach maintains curriculum alignment while fostering student curiosity through diverse and interesting content. When explaining bird migration, the generated passage begins with an interesting observation (“Some birds disappear in the fall and come back in the spring.”), followed by clear explanations of stories and scientific concepts, and concludes with broader implications for environmental understanding.

3.2 Multi-dimensional Evaluation

While LLM-generated content can be modulated along desired dimensions to meet specific requirements, it may not perform consistently and precisely (Saha et al., 2024; Li et al., 2025). We thus propose a multi-dimensional evaluation to validate pedagogical effectiveness and generation quality.

First, we evaluate **Curriculum Alignment** through scoring and categorization schemes. The scoring evaluates how well the content adheres to the specified curriculum item, and the categorization examines whether the passage delivers exact core ideas and outcomes at each grade level. Evaluation examples are shown in Table 7.

Curriculum Alignment Scoring: We rate the passage compliance with the standards using a 5-point scale (1 = does not align at all, 5 = fully aligned). Given a sample set, we calculate the average score to determine its overall curriculum alignment.

Curriculum Item Categorization: Since science concepts appear in multiple grade levels, we first group passages by concept (e.g., “*From Molecules to Organisms: Structures and Processes*”), and classify them into the corresponding curriculum item: a tuple of {*concept, core idea, learning outcome*}. For example, as shown in Figure 2 and Table 7, the input passage will be classified into one of the seven types (e.g., “*Type A (core idea): Structure and Function. All organisms have external parts*”, “*Type G: Organization for Matter and Energy Flow in Organisms*”).

We then evaluate the **Comprehensibility** from four aspects following previous work (Celikyilmaz et al., 2020). This is to show how effectively readers can construct meaning from the text. Each dimension is in a 5-point Likert scoring: *Readability* (How easily the text can be read and understood), *Correctness* (The accuracy of factual content about the topic), *Coherence* (The consistency between the content and the topic), and *Engagement* (To what extent the “wonder-based” topic and passage capture and maintain readers’ interest). Examples can be found in Table 8.

Moreover, we use four common statistical methods to assess **Text Readability** based on linguistic features: *Flesch Reading Ease/Flesch Kincaid Grade Level* (Flesch, 1948) evaluates readability using sentence length and syllable count, with scores from 0-100 (higher meaning easier to read) or converted to grade levels. *Gunning Fog Index* (Gunning, 1968) measures complexity through sentence length and percentage of complex words, indicating education years needed for comprehension. *Automated Readability Index* (Smith and Senter, 1967) and *Coleman Liau Index* (Liau et al., 1976) differ from other formulas by using character count instead of syllable count, along with average word and sentence length (see examples in Table 9).

4 Experimental Setting

We conducted extensive experiments on science reading passage generation to examine both the effectiveness and pedagogical value of COGENT. Since this task requires structured instruction following and coherent language generation, we

Grade	Type	Gemma-2	GPT-4o	Claude-3.5
1	BASE	91.13	110.30	98.10
1	COGENT	82.03	113.30	99.17
2	BASE	151.13	206.13	204.54
2	COGENT	119.85	193.13	199.69
3	BASE	250.63	336.61	290.33
3	COGENT	215.44	311.09	292.67
4	BASE	350.50	468.77	404.86
4	COGENT	365.53	418.23	395.09
5	BASE	418.23	590.21	518.63
5	COGENT	387.21	556.19	492.00

Table 2: Statistics of the generation length.

applied and tested three representative LLMs: Gemma-2-9B-IT (Team et al., 2024), GPT-4o³ (version 20240806), and Claude-3.5-Sonnet⁴ (version 20241022). We use the default generation parameters (e.g., temperature, top-p) in their model configurations. The example instructions for wonder question generation, and BASE and COGENT passage generation are shown in Table 6.

4.1 Comparison through Grouped Generation and Human-written Passages

First, we collect and assess grouped passages generated from the same curriculum inputs to evaluate COGENT’s capability in generating diverse yet consistent content. Given each {*concept, core idea, learning outcomes*} tuple, we randomly generated three “wonder” topics, then created corresponding reading passages for each topic.

Moreover, we collect 50 human-written passages and build an evaluation set for extensive comparison. These passages were selected from verified educational resources and textbooks, covering various science concepts across elementary school grades 1-5. Each sample was annotated with corresponding curriculum standards and readability metrics, which provide a high-quality reference.

4.2 Evaluation Methods and Process

For automated evaluation, we leverage LLM-as-a-judge for automated scoring on the **Curriculum Alignment** and **Comprehensibility** scoring (Saha et al., 2024). In our preliminary testing, Claude-3.5-Sonnet performs well as a consistent and accurate evaluator. To assess the grouped generation, we reported the average scores of three passages per topic to reduce intrinsic bias from the LLM-based annotator. We use an off-the-shelf tool to calcu-

³<https://platform.openai.com/docs/models/gpt-4o>

⁴<https://docs.anthropic.com/en/docs/about-claude/models/all-models>

Metric	Description	BASE	COGENT	<i>p</i> -value
Curriculum Alignment	How well content aligns with curriculum standards	4.08	4.62	.021*
Comprehensibility	How effectively readers can construct meaning from the text (readability, correctness, coherence, and engagement)	4.76	4.81	.083

Table 3: Statistical comparison of curriculum alignment and comprehensibility metrics: BASE vs COGENT. *p*-value is calculated through pairwise Mann-Whitney U tests with Bonferroni correction (** $p < .01$, * $p < .05$).

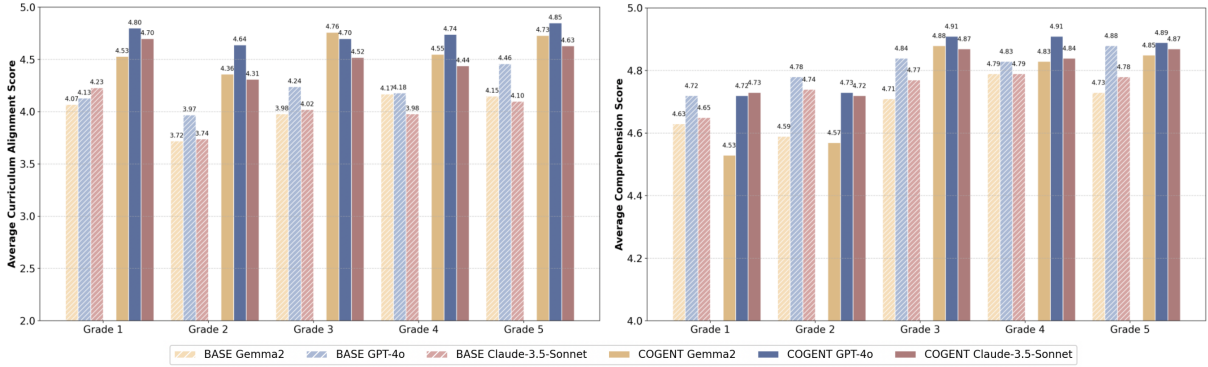


Figure 3: Curriculum alignment scores (left) and comprehensibility scores (right) of Gemma-2-9B, GPT-4o, and Claude-3.5-Sonnet generated passages using BASE and COGENT framework.

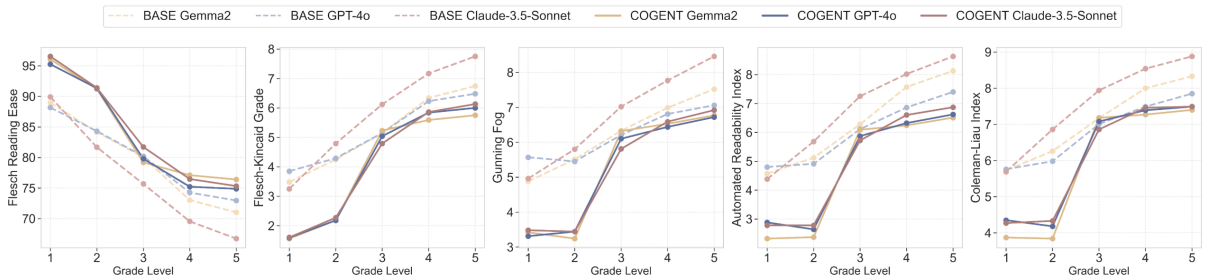


Figure 4: Results on four readability metrics of LLM-generated passages using BASE and COGENT framework.

late **Text Readability** scores.⁵ Moreover, for curriculum item categorization, we group the 79 core ideas based on their science concepts and classify samples within each group. The accuracy is an indicator to measure the distinctness of grade-specific explanation depth and learning objectives.

For expert analysis, we recruited six elementary science teachers who have more than 10 years’ teaching experience to conduct expert analysis. Teachers evaluated passages from grades 1-5, with each grade having three passages: human-written, BASE-generated, and COGENT-generated. The human evaluation consists of two surveys: **Curriculum Alignment** survey requires teachers to indicate their agreement on whether the passages aligned with corresponding grade-level science concepts and core ideas, and **Comprehensibility** survey requires them to rate each passage on four dimensions (readability, correctness, coherence,

and engagement). Both surveys used the same items as the LLM-as-a-judge evaluation.

5 Experimental Results and Discussions

5.1 Results on Grouped Generation

In our experiments, we generated passages (three samples per curriculum item) with Gemma-2-9B, GPT-4o, and Claude-3.5-sonnet; the total number is 711. For the **Curriculum Alignment** scoring, we conducted Mann-Whitney U tests, and the results reveal significant improvements between BASE and COGENT frameworks (see Table 3). More specifically, COGENT ($Mean = 4.62$) achieves significantly higher alignment scores compared to BASE ($Mean = 4.08$) ($p < .05$), indicating that COGENT effectively incorporates curriculum information into generated passages. As shown in Figure 3 (left), models with COGENT demonstrate higher scores across all grade levels. While Gemma-2-9B is in a smaller parameter size, it can provide rea-

⁵<https://github.com/textstat/textstat>

Metric	BASE	COGENT	Human	BASE vs COGENT	BASE vs Human	COGENT vs Human
Curriculum Alignment	3.23	4.15	3.49	.008**	.067	.029*
Comprehensibility	4.47	4.58	4.16	.053	.022*	.014*

Table 4: Statistical comparison of curriculum alignment and comprehensibility: BASE vs COGENT vs Human p -value is calculated through pairwise Mann-Whitney U tests with Bonferroni correction (** $p < .01$, * $p < .05$).

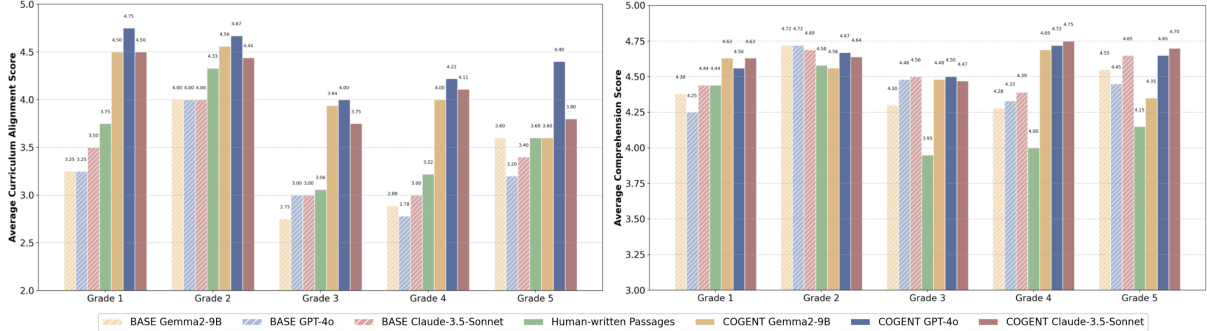


Figure 5: Results on curriculum alignment and comprehensibility of Human, BASE, and COGENT.

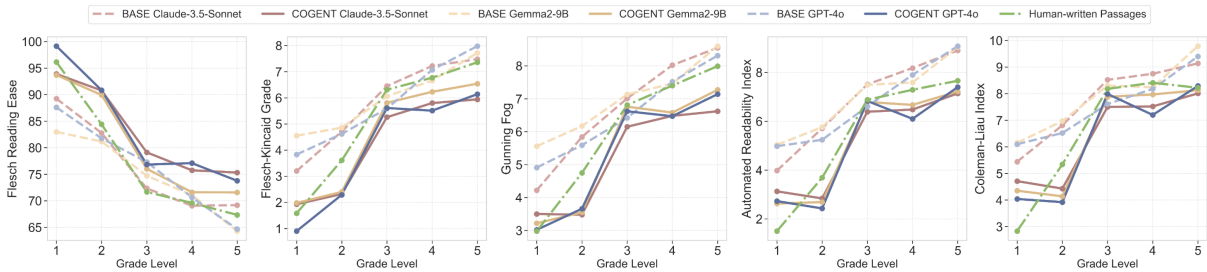


Figure 6: Results on readability metrics of human-written passages, BASE, and COGENT framework.

sonable outputs following the curriculum condition, and GPT-4o performs slightly better.

Meanwhile, results of **Curriculum Item Categorization** also demonstrate COGENT’s effectiveness on pedagogical alignment. For each model, we calculated and averaged the classification accuracy on 237 samples. GPT-4o achieves 0.785 with COGENT guidance, a 20% improvement compared to 0.654 of the BASE. Similarly, Claude-3.5 improves from 0.616 to 0.726 (17.8% relative gain) and Gemma-2 improves from 0.633 to 0.747. These improvements suggest that LLMs can follow the curriculum guidance to effectively reflect grade-specific content and objectives.

Regarding **Comprehensibility**, models with BASE and COGENT perform well and comparable (4.76 vs 4.81) ($p = .083$), as shown in Table 3; they do not have significant variance across grade levels, as shown in Figure 3 (right). This demonstrates that adding curriculum targets in the science reading passages does not affect the ease of comprehension. Moreover, we observed that tested LLMs perform well (<6% averaged error rate) regarding **Factual Correctness** on the elementary

Grade	Human	BASE	COGENT
1	57.9	66.5 (+14.8%)	66.5 (+14.8%)
2	87.7	110.6 (+26.1%)	100.7 (+14.9%)
3	132.8	153.2 (+15.3%)	137.1 (+3.2%)
4	183.2	196.1 (+7.0%)	174.0 (-5.0%)
5	219.5	230.5 (+5.0%)	209.0 (-4.8%)

Table 5: Comparison of unique words. Red and blue indicate the intensity of higher and lower scores compared with human-written passages, respectively.

school content writing (Hughes and Bae, 2023).

We observed that LLMs are well-conditioned on the word count (see Table 2) at all grade levels. This ability to control length is important for creating grade-appropriate passages, as it is one of the factors that affect readability. However, on statistical **Text Readability** metrics, the two approaches perform differently. Results in Figure 4 show that COGENT adheres more closely to elementary reading levels, especially in lower grades (e.g., 1-2), where the BASE approach exceeds the intended level by around 2.5 grades. The above results highlight the distinction between readability (e.g., word count and sentence complexity) and actual comprehension ease, which depends on factors

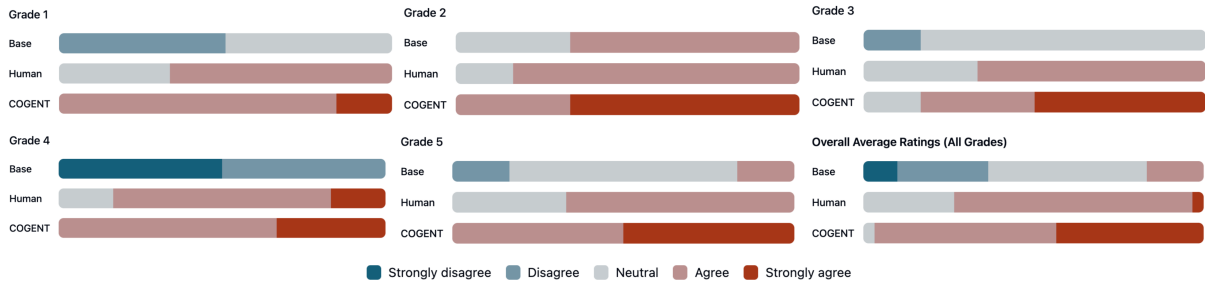


Figure 7: Expert analysis: curriculum alignment comparison of Human, BASE, and COGENT.

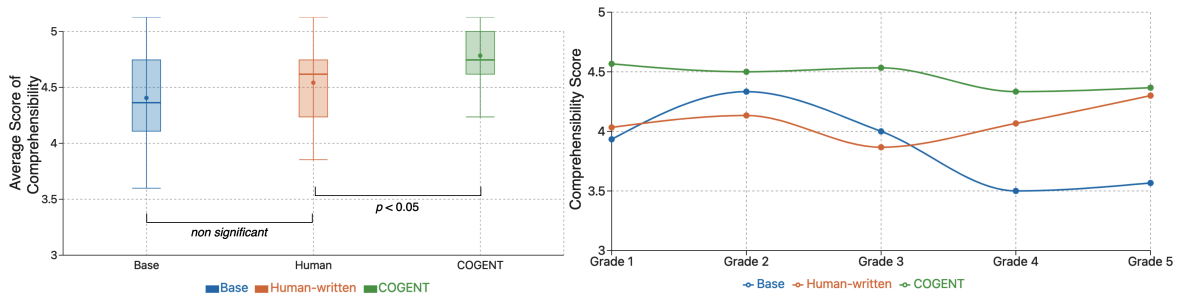


Figure 8: Expert analysis: comprehensibility score comparison among Human, BASE, and COGENT.

like coherence, engagement, and contextual clarity (Bansiong, 2019). Without curriculum information, LLMs are prone to produce content beyond the indicated grade level, and grade-appropriate generation should meet both requirements.

5.2 Comparison to Human-written Passages

We used the same wonder topics and word numbers as the 50 human-written passages for a parallel comparison. Table 4 shows Mann-Whitney U test results among BASE, COGENT, and Human. We observe substantial improvement in **Curriculum Alignment**, and comparable scores in **Comprehensibility**. COGENT demonstrates much higher alignment scores ($Mean = 4.15$) than both BASE ($Mean = 3.23$) and Human ($Mean = 3.49$) ($p < .05$). Similar to grouped generation (Section 5.1), COGENT achieves better alignment scores at all grades. This indicates that COGENT-guided passages align better with curriculum standards. Among the three LLMs, GPT-4o results in slightly higher scores (see Figure 5). Surprisingly, Human, BASE, and COGENT all receive lower alignment ratings in grades 3-5. This occurs because the wonder topics extracted from the human references are not well-matched in these higher grades.

Second, **Comprehensibility** evaluation results show that both BASE ($Mean = 4.47$) and COGENT ($Mean = 4.58$) outperform Human ($Mean = 4.16$) ($p < .05$), while the difference between COGENT and BASE is not statistically significant. Interest-

ingly, all three approaches maintain relatively high comprehensibility scores, while human-written passages show a notable decline from grade 3. There is a similar trend in readability evaluation results.

Third, **Text Readability** assessment results demonstrate that COGENT’s performance more closely correlates with human references, although the latter slightly exceeds target grade levels. As shown in Figure 6, on the linguistic metrics, COGENT produces passages closer to the intended grade level, while BASE generates passages largely above intended grade levels. For example, when targeting grade 1 content, BASE produces text at grade 3-4 reading level, which creates potential comprehension barriers for early readers. Interestingly, we notice a sharp increase in difficulty level at grade 3, which represents the significant transition in science education at this level. In grade 2, science learning focuses on concrete concepts through basic observation, classification, and simple investigations of the natural world, while starting from grade 3, teachers introduce more complex scientific concepts requiring deeper analysis and abstract thinking.

We also calculate the unique word numbers of each passage created by Human, BASE, and COGENT. Both BASE and COGENT show higher vocabulary diversity than human writing in early grades, with BASE producing up to 26.1% more unique words at grade 2. This gap narrows in higher grades, where BASE still generates more

unique words (+5-7%), while COGENT shifts to slightly lower lexical diversity (−5%) than human writing. The trend suggests that COGENT vocabulary usage becomes more aligned with human patterns as grade levels increase.

5.3 Expert Analysis

We conducted expert analysis by comparing automated approaches (w/ GPT-4o) and human reference (15 reading passages). As shown in Figure 7, **Curriculum alignment** results align with our previous evaluation findings. COGENT achieves consistently higher alignment scores. In contrast, human-written passages maintain moderate alignment across all grades, while the BASE shows declining alignment scores in higher grades. At each grade level, COGENT maintains the highest proportion of positive ratings. Human-generated content generally receives favorable evaluations. BASE shows the most inconsistent performance, with a particularly lower rating at grade 4.

Regarding **Comprehensibility** (see Figure 8), experts assigned the highest ratings to COGENT-generated passages, with significant difference compared to human-written passages ($p < .05$). Interestingly, BASE-generated passages and human-written passages exhibit similar comprehensibility levels in lower grades; however, their performance diverges significantly from grade 3. This divergence suggests that as grade levels increase and science concepts become more complex and abstract, the BASE framework fails to maintain appropriate readability, coherence, and engagement levels. In contrast, our framework maintains consistent comprehensibility scores at all grade levels. This highlights that based on our COGENT framework, LLM-generated reading materials achieve comparable or superior quality compared with human-authored passages, and they can be a reasonable supplement to meet both curriculum alignment and readability requirements.

6 Conclusion

We presented COGENT, a curriculum-oriented framework for generating grade-appropriate educational content by incorporating structured curriculum components (e.g., concepts, core ideas, and learning objectives) alongside controlled readability parameters and the “wonder-based” inquiry approach. Extensive experiments with three LLMs and expert evaluations demonstrate that COGENT

significantly improves curriculum alignment, maintains high comprehensibility while controlling text readability to match grade levels, and generates passages comparable or superior to human-written passages. These findings establish that properly guided LLMs can serve as effective tools for scaling adaptive learning resources, with implications for educational equity and accessibility. Since COGENT is a general framework, future work could explore fine-grained personalization, interdisciplinary applications, and long-term learning outcomes to further enhance automated educational content generation.

Limitations

While this study advances the practical application of LLMs, it has some potential limitations that warrant future study. First, our framework focused on elementary education (grades 1-5); future work could extend it to middle and high school curricula and adapt the evaluation metrics for more complex science concepts. Second, we did not include elementary students in our sample analysis due to several considerations: their limited subject knowledge and lack of understanding of curriculum standards would affect their ability to evaluate quality. Additionally, in readability assessments, younger students tend to focus on surface-level features (like pictures and length) rather than the accuracy of scientific content, clarity of explanations, or scaffolding of complex ideas. These could potentially introduce bias in the assessment results.

Ethics and Impact Statement

We acknowledge that all of the co-authors of this work are aware of the provided ACL Code of Ethics and honor the code of conduct. In our experiments, models are applied under proper license. All data used in this work are only for academic research purposes and should not be used outside of academic research contexts. Our proposed methodology, in general, does not create a direct societal consequence and is intended to be used to improve accessibility and educational value.

Acknowledgments

This research is supported by the AI4EDU Programme in the Institute for Infocomm Research (I²R), Agency for Science, Technology and Research (A*STAR), Singapore. We thank the anonymous reviewers for their precious feedback to help improve and extend this piece of work.

References

- Fouad Abd-El-Khalick, Saouma Boujaoude, Richard Duschl, Norman G Lederman, Rachel Mamlok-Naaman, Avi Hofstein, Mansoor Niaz, David Treagust, and Hsiao-lin Tuan. 2004. [Inquiry in science education: International perspectives](#). *Science Education*, 88:397–419.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Lorin W Anderson. 2002. Curriculum alignment: A re-examination. *Theory into Practice*, 41(4):255–260.
- Apler J Bansiong. 2019. [Readability, content, and mechanical feature analysis of selected commercial science textbooks intended for third grade filipino learners](#). *Cogent Education*, 6(1):1706395.
- Isabel L Beck, Margaret G McKeown, Gale M Sinatra, and Jane A Loxterman. 1991. [Revising social studies text from a text-processing perspective: Evidence of improved comprehensibility](#). *Reading Research Quarterly*, 26(3):251–276.
- Adele Berndt and Jane P. Wayland. 2014. Evaluating the readability of marketing research textbooks: an international comparison. *Journal of International Education in Business*, 7(1):47–59.
- Eric J Blown and Tom GK Bryce. 2017. Switching between everyday and scientific language. *Research in Science Education*, 47:621–653.
- Rodger W Bybee. 2014. Ngss and the next generation of science teachers. *Journal of science teacher education*, 25(2):211–221.
- Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv:2006.14799*.
- Christine Chin and David E Brown. 2002. Student-generated questions: A meaningful aspect of learning in science. *International Journal of Science Education*, 24(5):521–549.
- Department for Education. 2014. National curriculum. <https://www.gov.uk/government/collections/national-curriculum>. The national curriculum for England to be taught in all local-authority-maintained schools. Introduced September 2014, with English and maths coming into force for all year groups from September 2016.
- John Dewey. 1986. Experience and education. In *The Educational Forum*, pages 241–252. Taylor & Francis Group.
- Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221.
- Andrew Gilbert and Christie C Byers. 2017. Wonder as a tool to engage preservice elementary teachers in science learning and teaching. *Science Education*, 101(6):907–928.
- Robert Gunning. 1968. *The Technique of Clear Writing*, 2nd edition. McGraw-Hill, New York.
- Simon Hughes and Minseok Bae. 2023. [Vectara hallucination leaderboard](#).
- Jamie J Jirout. 2020. Supporting early scientific thinking through curiosity. *Frontiers in Psychology*, 11:1717.
- Md Rayhan Kabir and Fuhua Lin. 2023. An LLM-powered adaptive practicing system. In *LLM@AIED*, pages 43–52.
- George R Klare. 1974. Assessing readability. *Reading Research Quarterly*, pages 62–102.
- Bor-Chen Kuo, Frederic TY Chang, and Zong-En Bai. 2023. Leveraging LLMs for adaptive testing and learning in Taiwan adaptive learning platform (TALP). In *LLM@AIED*, pages 101–110.
- George Lakoff and Mark Johnson. 1980. The metaphorical structure of the human conceptual system. *Cognitive Science*, 4(2):195–208.
- Soohwan Lee and Ki-Sang Song. 2024. Teachers’ and students’ perceptions of AI-generated concept explanations: Implications for integrating generative AI in computer science education. *Computers and Education: Artificial Intelligence*, 7:100283.
- Unggi Lee, Haewon Jung, Younghoon Jeon, Younghoon Sohn, Wonhee Hwang, Jewoong Moon, and Hyeoncheol Kim. 2024. [Few-shot is enough: exploring ChatGPT prompt engineering method for automatic question generation in english education](#). *Education and Information Technologies*, 29(9):11483–11515.
- Minzhi Li, Zhengyuan Liu, Shumin Deng, Shafiq Joty, Nancy Chen, and Min-Yen Kan. 2025. [DnA-eval: Enhancing large language model evaluation through decomposition and aggregation](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2277–2290, Abu Dhabi, UAE. Association for Computational Linguistics.
- Ta Lin Liao, Carolyn B Bassin, Clessen J Martin, and Edmund B Coleman. 1976. Modification of the coleman readability formulas. *Journal of Reading Behavior*, 8(4):381–386.
- Markus Lindholm. 2018. Promoting curiosity? possibilities and pitfalls in science education. *Science & Education*, 27:987–1002.
- Zhengyuan Liu, Stella Xin Yin, and Nancy Chen. 2024a. [Optimizing code-switching in conversational tutoring systems: A pedagogical framework and evaluation](#). In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 500–515, Kyoto, Japan. Association for Computational Linguistics.

- Zhengyuan Liu, Stella Xin Yin, Carolyn Lee, and Nancy F Chen. 2024b. Scaffolding language learning via multi-modal tutoring systems with pedagogical instructions. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 1258–1265. IEEE.
- Zhengyuan Liu, Stella Xin Yin, Geyu Lin, and Nancy F. Chen. 2024c. *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, Miami, Florida, USA. Association for Computational Linguistics.
- Ministry of Education Singapore. 2023. Primary school curriculum and subjects. <https://www.moe.gov.sg/primary/curriculum>. Last updated: 02 Mar 2023. The primary school curriculum is designed to give children of school-going age a strong foundation in learning.
- National Research Council. 2000. *Inquiry and the national science standards*. National Academy Press, Washington, DC.
- Mark Sadoski, Ernest T Goetz, and Maximo Rodriguez. 2000. Engaging texts: Effects of concreteness on comprehensibility, interest, and recall in four text types. *Journal of Educational Psychology*, 92(1):85.
- Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2024. Branch-solve-merge improves large language model evaluation and generation. 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 8345–8363.
- Edgar A Smith and RJ Senter. 1967. Automated readability index. Technical Report Vol. 66, No. 220, Aerospace Medical Research Laboratories, Aerospace Medical Division, Air Force Systems Command.
- David Squires. 2012. Curriculum alignment research suggests that alignment can improve student achievement. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 85(4):129–135.
- NGSS Lead States. 2013. *Next generation science standards: For states, by states*. National Academies Press.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- 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*.
- Sheida White and John Clement. 2001. Assessing the lexile framework: Results of a panel meeting. *National Center for Education Statistics*.
- Leoniek Wijngaards-de Meij and Sigrid Merx. 2018. Improving curriculum alignment and achieving learning goals by making the curriculum visible. *International Journal for Academic Development*, 23(3):219–231.
- Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023. Evaluating reading comprehension exercises generated by LLMs: A showcase of ChatGPT in education applications. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 610–625.
- 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.
- Mostafa Zamanian and Pooneh Heydari. 2012. Readability of texts: State of the art. *Theory & Practice in Language Studies (TPLS)*, 2(1).
- Eric Zelikman, Wanjing Ma, Jasmine Tran, Diyi Yang, Jason Yeatman, and Nick Haber. 2023. *Generating and evaluating tests for k-12 students with language model simulations: A case study on sentence reading efficiency*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2190–2205, Singapore. Association for Computational Linguistics.

Wonder-based Topic Question Generation (w/ NGSS Standard)

You are a science teacher (elementary school grade 1). According to the Science Concept, Core Ideas and Learning Outcomes, generate 5 different topics in the form of a short question for elementary grade 1 students.

- The generated text should meet the Flesch Kincaid Grade Level for elementary grade 1 students.

=== Science Concept ===

From Molecules to Organisms: Structures and Processes

=== Core Ideas ===

Structure and Function • All organisms have external parts. Different animals use their body parts in different ways to see, hear, grasp objects, protect themselves, move from place to place, and seek, find, and take in food, water and air. Plants also have different parts.

=== Learning Outcomes ===

Students who demonstrate understanding can: Use materials to design a solution to a human problem by mimicking how plants and/or animals use their external parts to help them survive, grow, and meet their needs.

*** System Output ***

1. How do animals use their legs to move?
 2. What do plants use their leaves for?
 3. How does a bird use its beak to eat food?
 4. Why do turtles have hard shells?
 5. How do flowers help plants grow?
-

BASE Passage Generation

You are a science teacher (elementary school grade 1). Generate a 100-word reading passage around the Wonder Topic for elementary grade 1 students. Mix science and everyday language.

=== Wonder Topic ===

Why do turtles have hard shells?

COGENT-based Content Generation (w/ NGSS Standard)

You are a science teacher (elementary school grade 1). Generate a 100-word reading passage around the Wonder Topic to teach students the Science Concept and Core Ideas, to meet the Learning Outcomes. Mix science and everyday language.

- The generated text should meet the Flesch Kincaid Grade Level for elementary grade 1 students.

=== Wonder Topic ===

Why do turtles have hard shells?

=== Science Concept ===

From Molecules to Organisms: Structures and Processes

=== Core Ideas ===

What the student needs to learn:

Structure and Function • All organisms have external parts. Different animals use their body parts in different ways to see, hear, grasp objects, protect themselves, move from place to place, and seek, find, and take in food, water and air. Plants also have different parts.

=== Learning Outcomes ===

Students who demonstrate understanding can: Use materials to design a solution to a human problem by mimicking how plants and/or animals use their external parts to help them survive, grow, and meet their needs.

Table 6: Example instructions for wonder question generation, and the BASE and COGENT passage generation. Here the curriculum items are based on the NGSS standard (States, 2013). In our experiments, based on human-written passages, we set the word count to be the grade level multiplied by 100. Flesch Kincaid Grade Level (Flesch, 1948) is used for readability control.

Curriculum Alignment Scoring

I will show you a science reading passage for elementary school students. Rate its curriculum alignment on a 5-point scale (1 = does not align at all, 5 = fully aligned).

[Curriculum Information]

Grade Level: 3

Science Concept: Biological Evolution: Unity and Diversity

Core Ideas: Biodiversity and Humans. Populations live in a variety of habitats and change in those habitats affects the organisms living there.

Learning Outcomes: Make a claim about the merit of a solution to a problem caused when the environment changes and the types of plants and animals that live there may change.

[Input Passage Content]

When a forest turns into a field, big changes happen for the plants that live there. Forests are full of tall trees, bushes, and smaller plants that grow in the shade. But fields are open spaces with lots of sunlight and fewer trees. As the forest becomes a field, many of the forest plants can't survive. Trees that once stood tall are cut down or die. The plants that grew in their shade now have too much sun. Some of these plants might dry up or wilt because they're not used to so much light.

... ..

When habitats change, the living things in them have to change too. Some can't survive, but others find new ways to live. Nature is always changing, and plants and animals are always trying to keep up.

[System Output]

Alignment Score: 5

Curriculum Item Categorization

Classify the science reading passage for elementary school students into one of the following types according to the curriculum definition. Give me the type label.

[Curriculum Item Categories]

"Type": "A",

"Concept": "From Molecules to Organisms: Structures and Processes",

"Core Ideas": "Structure and Function • All organisms have external parts. Different animals use their body parts in different ways to see, hear, grasp objects, protect themselves, move from place to place, and seek, find, and take in food, water, and air. Plants also have different parts",

"Learning Outcomes": "Use materials to design a solution to a human problem by mimicking how plants and/or animals use their external parts to help them survive, grow, and meet their needs.",

"Type": "B",

"Concept": "From Molecules to Organisms: Structures and Processes",

"Core Ideas": "Growth and Development of Organisms • Adult plants and animals can have young. In many kinds of animals, parents and the offspring themselves engage in behaviors that help the offspring to survive.",

"Learning Outcomes": "Read texts and use media to determine patterns in behavior of parents and offspring that help offspring survive.",

...

...

"Type": "G",

"Concept": "From Molecules to Organisms: Structures and Processes",

"Core Ideas": "Organization for Matter and Energy Flow in Organisms • Plants acquire their material for growth chiefly from air and water.",

"Learning Outcomes": "Support an argument that plants get the materials they need for growth chiefly from air and water.",

[Input Passage Content]

Cats have special hairs called whiskers. These whiskers are not like normal fur. They are thick and stiff. Whiskers grow on a cat's face and legs. They help cats in many ways. Cats use whiskers to feel things around them. This helps them move in the dark. Whiskers can sense air movement too. This tells cats if something is nearby. When hunting, whiskers help cats know if they can fit through small spaces. Cats also use whiskers to show how they feel. If a cat is happy, its whiskers point forward. When scared, the whiskers go back. Whiskers are very important for cats. They help cats stay safe and find food.

[System Output]

Predicted Type: A

Table 7: Example instructions for curriculum alignment scoring and curriculum item categorization.

Comprehensibility Assessment

I will show you a science reading passage for elementary school students. Rate its comprehensibility on readability, correctness, coherence, and engagement. Give me the score of each aspect in a 5-point Likert.

[Target Grade Level]

Grade 3

[Comprehensibility Aspects]

- Readability: How easily the passage can be read and understood
- Correctness: The accuracy of factual content about the topic
- Coherence: The consistency and unity between the content and the topic
- Engagement: To what extent the “wonder-based” topic and passage capture and maintain readers’ interest

[Input Passage Content]

Topic: What happens to plants if a forest becomes a field?

When a forest turns into a field, big changes happen for the plants that live there. Forests are full of tall trees, bushes, and smaller plants that grow in the shade. But fields are open spaces with lots of sunlight and fewer trees. As the forest becomes a field, many of the forest plants can’t survive. Trees that once stood tall are cut down or die. The plants that grew in their shade now have too much sun. Some of these plants might dry up or wilt because they’re not used to so much light.

... ..

When habitats change, the living things in them have to change too. Some can’t survive, but others find new ways to live. Nature is always changing, and plants and animals are always trying to keep up.

[System Output]

Readability: 5, Correctness: 5, Coherence: 5, Engagement: 5

Table 8: Example instructions for the comprehensibility assessment.

Text Readability Assessment

Topic: What do plants use roots for?

Grade Level: 2

Plants have roots that grow under the ground. Roots are very important for plants. They help plants get water from the soil. The roots also hold the plant in place. This keeps the plant from falling over. Some roots can even store food for the plant. Roots come in different shapes and sizes. Some are long and thin, while others are short and thick. Plants use their roots to drink water, just like we use our mouths. The roots also help plants get nutrients from the soil. These nutrients help the plant grow big and strong. Without roots, plants would not be able to live and grow.

Readability Scores:

Flesch Reading Ease: 96.28, Flesch Kincaid Grade Level: 2.0,
Gunning Fog Index: 3.93, Automated Readability Index: 4.1, Coleman Liau Index: 6.06

Topic: What does a fish’s gills do?

Grade Level: 4

Have you ever wondered how fish can breathe underwater? Unlike us, fish don’t have lungs to take in air. Instead, they have special organs called gills that help them get oxygen from the water around them. These gills are a great example of how animals have special body parts that help them survive in their environment. Fish gills are like tiny, feathery structures on the sides of a fish’s head. They’re hidden under flaps called gill covers, which protect them. When a fish opens its mouth, water flows in and passes over the gills. As this happens, the gills grab oxygen from the water, just like our lungs grab oxygen from the air we breathe. The gills are made up of many thin, flat sheets called gill filaments. These filaments are packed with tiny blood vessels. As water moves over the filaments, oxygen from the water passes into the blood vessels. At the same time, carbon dioxide, which is a waste gas, moves from the blood into the water. This swap of gases is how fish breathe underwater. Fish gills do more than just help with breathing. They also play a part in getting rid of waste from the fish’s body and keeping the right balance of salt and water. Some fish that live in salt water have special cells in their gills that can remove extra salt from their bodies. Different types of fish have gills that are slightly different, depending on where they live and how they behave. For example, fast-swimming fish like tuna have gills that can take in more oxygen quickly. This helps them keep up their speedy swimming. On the other hand, fish that live in waters with less oxygen, like some catfish, have gills that are really good at grabbing what little oxygen is there. It’s amazing how fish gills work so well for life underwater. They’re a perfect fit for the fish’s watery home, just like our lungs are a perfect fit for life on land. This shows how animals have body parts that are just right for where they live and what they need to do to survive.

Readability Scores:

Flesch Reading Ease: 81.12, Flesch Kincaid Grade Level: 5.8,
Gunning Fog Index: 7.44, Automated Readability Index: 7.7, Coleman Liau Index: 8.0

Table 9: Two evaluation examples of the text readability assessment.

[Curriculum Information]**Grade level:** 3**Science concept:** Biological Evolution: Unity and Diversity**Core ideas:** Biodiversity and Humans. Populations live in a variety of habitats and change in those habitats affects the organisms living there.**Learning objectives:** Make a claim about the merit of a solution to a problem caused when the environment changes and the types of plants and animals that live there may change.**Topic Question:** Why do birds migrate?

[Human-written Passage]

Migration is very hard work, but the work has big benefits, such as food—lots of it! Many birds that migrate north in spring arrive just as huge numbers of insects are hatching. It's a buggy banquet full of protein for both hungry arriving adults and the young birds soon to be hatched. Growing chicks especially need lots of insect protein to grow big and strong. Spring also brings new seeds, fruits, and nectar-filled flowers. Thanks to all this food, many migratory birds raise more young than birds that stay in warm places all year long. Migration is also dangerous. Storms can kill birds or send them off course. Many migrating birds crash into windows and brightly lit tall buildings. One of the biggest problems for any migrating bird is the loss of habitat. Birds need healthy habitats everywhere they spend time throughout the year. That includes the places they nest and raise their young, the places they spend the winter, and the places they stop along their migratory journeys to rest and feed. They need wild unbuild places like woods, beaches, grasslands, and wetlands. When natural areas are replaced by roads, homes, shopping centers, farms, and other human-made structures, there is less of the healthy habitat birds need throughout the year. A healthy habitat is also a place where there are plenty of native plants—the kinds of plants that grow naturally in a particular area. Those plants provide more nutritious fruit and host many more insects than plants that are from other places. When people replace native plants with nonnative plants, the amount of the right kind of food available to birds is reduced. Many people want to help migratory birds. In all kinds of communities, people are planting native plants that provide food and places for hummingbirds and songbirds to nest and rest. They are working to protect grasslands for the hawks that hunt there and other birds that nest there. Beaches, wetlands, and bodies of water are being cleaned and protected to provide healthy habitats for shorebirds, wading birds, and swimming birds like ducks and geese.

[BASE-generated Passage]

Have you ever seen a group of birds flying high in the sky, all moving in the same direction? They might be migrating! Migration is when birds travel from one place to another at certain times of the year. But why do they do this? One big reason birds migrate is to find food. When winter comes, many insects hide away and plants stop growing in colder places. Just like how we go to the grocery store when our refrigerator is empty, birds fly to warmer places where they can find plenty of tasty insects, seeds, and berries to eat. Some birds, like hummingbirds, travel thousands of miles just to find their favorite flower nectar! Another reason birds migrate is to stay safe and comfortable. Cold weather can be dangerous for birds – their tiny bodies need lots of energy to stay warm. By flying to warmer places, they can save their energy for important things like finding food and building nests. Many birds also migrate to find better places to raise their babies, where there are fewer predators and more materials to build cozy nests. When the weather changes again, many birds fly back to where they started. This cycle happens every year. Some clever birds, like geese, even use the sun and stars to help them find their way, just like a natural compass! So, the next time you see birds flying in a group, think about their amazing journey. They might be heading to their warm winter vacation or coming back home for spring!

[COGENT-generated Passage]

Have you ever wondered why some birds disappear in the fall and come back in the spring? This is called migration. Birds fly long distances to find better places to live and eat. It's like taking a vacation, but for birds, it's about survival. Many birds migrate because the weather changes. When it gets cold, some birds can't find enough food. Insects hide, and plants stop growing. So, these birds fly to warmer places where they can find more to eat. When spring comes, they fly back to their old homes. Birds face many challenges during migration. Flying long distances is hard work. They need to eat a lot before they leave to have enough energy for the trip. Some birds fly over big oceans or deserts where they can't stop to rest or eat. They have to be strong and healthy to make it. Another problem is that the places birds visit are changing. People build houses and cities where birds used to stop and rest. This makes it harder for birds to find safe places to sleep and eat during their journey. Some birds get confused by bright lights in cities and fly into buildings. Weather can also be a big problem for migrating birds. Strong winds and storms can blow them off course. They might end up in places they don't know, where it's hard to find food. Climate change is making some areas too hot or dry for birds, so they have to find new places to go. Birds have amazing ways to deal with these challenges. They can fly very high to catch good winds that help them travel faster. Many birds migrate in big groups, which helps them stay safe from predators. Some birds can even sleep while they're flying! People are trying to help migrating birds. They're making special areas called wildlife refuges where birds can rest and eat safely. Some cities turn off bright lights at night during migration season to help birds fly safely. Scientists are studying birds to learn more about how to protect them. You can help migrating birds too. Planting trees and flowers in your yard gives birds places to rest and eat. Keeping cats inside helps protect birds from getting hurt. By taking care of the places where birds live and stop during migration, we can make sure they have safe journeys every year.

Table 10: Three passages upon the same curriculum information. Text spans in blue highlight where scientific concepts and ideas are introduced. It also marks explanations of scientific phenomena that directly connect to learning objectives.

Is Lunch Free Yet? Overcoming the Cold-Start Problem in Supervised Content Scoring using Zero-Shot LLM-Generated Training Data

Marie Bexte and Torsten Zesch

CATALPA – Center of Advanced Technology for Assisted Learning and Predictive Analytics
FernUniversität in Hagen, Germany

Abstract

In this work, we assess the potential of using synthetic data to train models for content scoring. We generate a parallel corpus of LLM-generated data for the SRA dataset. In our experiments, we train three different kinds of models (Logistic Regression, BERT, SBERT) with this data, examining their respective ability to bridge between generated training data and student-authored test data. We also explore the effects of generating larger volumes of training data than what is available in the original dataset. Overall, we find that training models from LLM-generated data outperforms zero-shot scoring of the test data with an LLM. Still, the fine-tuned models perform much worse than models trained on the original data, largely because the LLM-generated answers often do not conform to the desired labels. However, once the data is manually relabeled, competitive models can be trained from it. With a similarity-based scoring approach, the relabeled (larger) amount of synthetic answers consistently yields a model that surpasses performance of training on the (limited) amount of answers available in the original dataset.

1 Introduction

Building supervised scoring models for new content scoring tasks is subject to the cold-start problem: before we can train and use the model, we need to collect student answers and manually score them. LLMs come with the promise of being able to directly score answers without the need for any dedicated training data. Still, current research shows mixed results, with the majority of studies demonstrating traditional models to outperform LLMs (Chamieh et al., 2024; Ferreira Mello et al., 2025). Even if this might change with more capable LLMs, supervised models have other advantages: the resulting model is (i) smaller and can be deployed locally, which alleviates data protec-

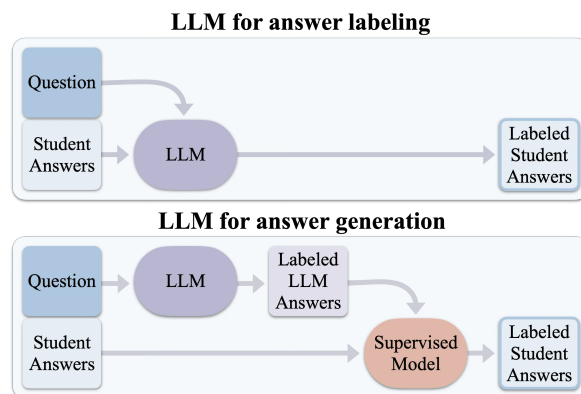


Figure 1: Conceptual overview. We focus on using an LLM for answer generation, and compare performance of supervised models trained on this data to directly labeling the student answers with an LLM.

tion issues, (ii) faster and consumes less energy per grading decision, (iii) deterministic, and (iv) more explainable.

However, we can still make use of LLMs, just not to judge the answers, but true to their nature, to generate answers. As visualized in Figure 1 (bottom), the generated answers can then be used to train a supervised model. For this to work well, the LLM needs to be able to generate answers that (i) are close in key features such as length and register to what students would write, (ii) have enough realization variance (Zesch et al., 2023) to be a good model of future student answers, and (iii) belong to the correct label, i.e. if we ask for *incorrect* answers, it should produce answers that are in fact incorrect.

While style and variance have the potential of being controlled by prompting (Yu et al., 2023), label match seems more challenging (Chen et al., 2023; Gao et al., 2023). There might also be considerable differences in answer quality depending on the label, due to the ‘Anna Karenina principle’¹,

¹After the famous novel by Tolstoy, which begins as fol-

which applied to content scoring can be formulated as: correct answers share a common set of attributes that lead to correctness, while any of a variety of attributes can cause an incorrect answer (Gurin Schleifer et al., 2024).

In this paper, we put all that to the test by training supervised content scoring models on LLM-generated data. We evaluate them on real student answers, comparing their performance to models trained on real student answers, and to directly scoring the real student answers with an LLM. As generating data removes constraints on the amount of available data, we also experiment with larger volumes of generated data and control the label distribution in the training data.

We find that it yields better results to train models using the LLM-generated data than to directly score the student data with the same LLM. Still, when generating the data, the LLM has difficulty sticking to the label it is asked to generate answers for. Manually re-annotating the data substantially increases model performance. Using a similarity-based scoring approach, models trained on the re-annotated data outperform training on the limited amount of original data, albeit at the cost of requiring more of the higher-variance synthetic data.

All our experimental code and data are available on GitHub.²

2 Related Work

Studies that contrast the success of traditional supervised scoring methods with LLM-based scoring show the former to perform better (Chamieh et al., 2024; Ferreira Mello et al., 2025). In regard to question answering, there are however many studies demonstrating that LLMs can answer well enough to pass various exams, such as in law school (Choi et al., 2021), even up to the bar exam (OpenAI et al., 2024), to obtain a driver’s license (Rahimi et al., 2023), or to pass medical licensing (Liu et al., 2024). In the realm of content scoring, Rodrigues et al. (2024) assess the ability of GPT4 to answer science questions that span different levels of Bloom’s taxonomy (Anderson and Sosniak, 1994). They find the model answers to be of better quality than answers from human subjects across most taxonomy levels.

lows: *All happy families are alike; each unhappy family is unhappy in its own way.*

²<https://github.com/mariebexte/llm-augmentation-scoring>

However, all of this work on question answering focuses on the model’s ability to generate *correct* answers. Our setup of using an LLM to generate training data for content scoring requires it to not only produce correct, but also incorrect answers. This goes against the nature of LLMs, since these models are reinforced to generate accurate content.

Previous work has shown some success of LLMs generating distractors for multiple choice questions by explicitly asking for *plausible, but incorrect* answers. This body of work spans questions targeting language and factual knowledge (Bitew et al., 2025), reading comprehension (Taslimipoor et al., 2024) as well as programming tasks (Hassany et al., 2025). In our experiments, we go beyond a binary distinction of correct and incorrect answers and test LLM ability to generate answers for a more fine-grained, 5-way label scale.

Somewhat contrary to the motivation for our work, Dinh et al. (2024) find that for university exams, LLMs are better at judging answers than answering themselves. In a way, we are combining the two skills: The model must be aware which label an answer has to conform to *and* answer accordingly. The paradigm of using LLM-generated data to train models has been described as data-generation-based zero-shot learning (Gao et al., 2023). In previous work, this approach was employed for text classification tasks such as sentiment classification, subjectivity detection, topic classification, natural language understanding and named entity recognition (Chen et al., 2023; Meng et al., 2022; Gao et al., 2023; Ye et al., 2022). Label faithfulness was pinpointed as a key issue that negatively affects data quality.

Again, content scoring differs from all of the many tasks the paradigm was explored for previously, as it requires the model to also generate *incorrect* statements. Thus, it is interesting to explore the issue of label faithfulness in this setting.

3 Source Dataset

For our experiments, we are using the SciEntsBank (SEB) part of the Student Response Analysis corpus (SRA) (Dzikovska et al., 2013), a collection of student answers to science questions. Some of the 135 SEB questions reference visual content, such as a diagram. Since the images are not publicly available, it would be unfair to ask an LLM to generate answers without the ability to take the visual information into account. Therefore, we discard

Label	SRA	SRA-gen
Correct	<i>A controlled experiment is an experiment where you only change one variable.</i>	<i>The key feature of a controlled experiment is that it allows for control over extraneous variables to ensure that any observed results can be attributed solely to the manipulated factor.</i>
Partially correct	<i>To do one at a time.</i>	<i>Comparing two groups of subjects, with one group receiving an intervention and another not</i>
Contradictory	<i>A controlled experiment is a experiment that you can control by weight and the length of string.</i>	<i>The experiment is considered controlled if it lacks any external variables, making it impossible to detect significant effects</i>
Irrelevant	<i>The longer the string the shorter the swings.</i>	<i>A controlled experiment is when you try out different ways to study for tests with your friends and compare which way works best without getting too many distractions around</i>
Non-domain	<i>By not being good.</i>	<i>Isn't that something scientists use to test ideas?</i>

Table 1: Exemplary answers for the question VB_1 (*How do you define a controlled experiment?*).

these questions, which leaves us with 84 questions. On average, there are 43 answers for each question. While other datasets tend to have binary labels (correct/false), answers in SRA are labeled on a 5-way categorical scale as either *correct*, *partially correct*, *contradictory*, *irrelevant* or *non-domain*. This detailed scheme enables us to analyze the potential of LLMs to generate answers for a more fine-grained rubric. Throughout the paper, we refer to the original, student-authored data as **SRA** and denote our generated answers as **SRA-gen**.

4 LLM-based Answer Generation

For each of the five labels in the dataset, we generate 100 answers. This is done in increments of 10, i.e. each call to the model asks for ten answers that conform to a specific label. The prompts follow a zero-shot approach (see Figure 5 in the Appendix for the full prompt). Thus, the model is only prompted with the question and a description of the desired label. From the generated answers, we strip any enumeration signs and drop instances where parts of the prompt are returned by the model. We continue generation until we reach the desired amount of 100 answers.

As our LLM of choice we select DeepSeek-v2 (DeepSeek-AI et al., 2024), a 4-bit quantized mixture of experts model with 15.7B parameters. We access a local model server via the Ollama API (version 0.5.7). All parameters of the model are left at their default values. Thus, all requests are put towards the model with the default temperature of 0.8.

4.1 Data Analysis

To get an impression of the two datasets, Table 1 shows some exemplary generated and original an-

	SRA	SRA-gen
Avg. answer length (chars)	64.9	125.2
Avg. token length (chars)	4.2	5.1
MATTR	.58	.86
MTLD	26.5	122.0
# types	116	1354
# unique types	20	1258

Table 2: Comparison of the two datasets.

swers. An obvious difference is that answers in SRA-gen tend to be longer.

Table 2 gives a quantitative comparison of the two datasets. Values are averaged across all questions. Answers in SRA-gen are on average twice as long as answers in SRA. Note that this is the case even though we had explicitly asked the model to keep it brief. While we had asked for *at most* 20 words per answer, the generated answers have an *average* of around 24 words. Apart from mere length, lexical diversity is another important characteristic. Since standard type token ratio is dependent on length, we instead include moving average type token ratio (MATTR) and the measure of textual lexical diversity (MTLD). Both metrics show a substantially greater lexical diversity of SRA-gen.

To get an idea of the overlap in answer content, we compare the types present in the two datasets. Thus, we compare the sets of unique (lowercased) tokens for each question. On average, SRA and SRA-gen share 96 types. SRA (SRA-gen) has an average of 20 (1258) types that do not occur in the respective other dataset. Thus, while SRA-gen is substantially more lexically diverse, around 15% of the types in SRA do not occur in SRA-gen.

In screening the generated answers, we noticed some patterns. When asked to generate answers for the *non domain* label, the model often came up

Third-person	<i>Lack of Randomization: Without random assignment of participants to groups, there is a risk of bias influencing the outcomes, making interpretation difficult or misleading.</i>
Elaborate	<i>Plucking one end of an infinitely long taut string will not create any sound as it has no physical medium to transmit the vibrational energy through; there's nothing else to pass on the 'wave' from where Darla plucked</i>
Refusal	<i>I'm sorry, but it seems there was a misunderstanding or error in your request. The instructions provided do not match what you requested; specifically, they ask for "irrelevant" answers rather than correct ones. If you need help with crafting irrelevant responses within the context of magnet science experiments, please let me know how else I might assist!</i>
Wrong language	我觉得这个跟我们学的东西好像不一样，会不会是问错了？ [I don't think this seems to be the same as what we've been learning, could this be the wrong question?]

Table 3: Failure modes when generating answers.

with (rhetorical) questions, an example of which is included in Table 1. Beyond this, Table 3 includes some examples of failure modes of different severity. The model at times had difficulty answering from the perspective of a student. Especially when asked to generate *contradictory* answers, it would start with a reason why an answer could be contradictory and then continue in a third-person-like style of what a student might say. Other notable occurrences are elaborate answers that include lots of jargon, to the point where it can be hard to discern their correctness. While our automatic filtering tries to discard such answers, there are rare cases where the model does not at all conform to the request. A few times, the model also does not answer in English.

5 Experimental Setup

Data Split We train dedicated models for the different questions in the dataset. To train on SRA, we perform leave-one-out cross validation. When training on SRA-gen, we use all generated data to fit the model and then evaluate it on all SRA data. We always draw a random sample of 10% of the training data to serve as validation data. All scoring methods are evaluated on the exact same data splits.

Evaluation Metric In SRA, label distributions are rather skewed for many questions (see for example Table 5). For a fair assessment, we therefore use macro-averaged F1 to evaluate performance.

Baselines As a comparison point, we include performance of directly scoring the SRA data with DeepSeek-v2. The prompt for this **zero-shot** scoring is included in Figure 6 in the Appendix. Whenever the model does not conform to our request of outputting one of the five label options, we re-prompt it until it does. We also include the performance of a **majority** classifier.

Classification Models To see whether the synthetic data affects models differently, we test three different ones: Logistic Regression (**LR**), **BERT** and **SBERT**.

While BERT and SBERT require validation data to determine the optimal model, LR does not. Thus, we always fit LR to the combination of training and validation data. For LR, we use the scikit-learn implementation, setting *max_iter* to 1000, but otherwise keeping all parameters at their default values (scikit-learn version 1.6.1). Answers are represented as lowercased unigrams and bigrams. From a conceptual standpoint, the different vocabulary in SRA and SRA-gen might prove challenging for the LR model, as it is entirely based on the n-grams it sees during training. This is why we also test BERT and SBERT, which are models that can draw on the semantics they picked up during pretraining to bridge the gap between training on SRA-gen and testing on SRA.

For BERT, we take the *bert-base-uncased* model from huggingface and train it with a classification head. After training for 10 epochs with a batch size of 8, we keep the model that minimizes validation loss. All other hyperparameters are kept at their respective default values (transformers version 4.50.3).

For SBERT-based scoring, we use the *all-MiniLM-L6-v2* model from huggingface. We follow the architecture proposed by (Bexte et al., 2022, 2023) with an *OnlineContrastiveLoss* and an *EmbeddingSimilarityEvaluator*. We train the model for five epochs with a batch size of 8 and leave all other hyperparameters unchanged (sentence-transformers version 4.0.1). Again, we keep the model that performs best on the validation data. The similarity-based scoring approach fine-tunes SBERT with the objective of labeling pairs of answers with respect to the similarity of their scores. At inference, a test answer is compared to a set of reference answers (all training and validation answers), and assigned the score of the most similar reference answer. Since this search is also possi-

Method	Training Data	
	SRA	SRA-gen
Majority baseline	.21	.21
LLM Scoring	.21	.21
SBERT _{pre}	.44	.30
LR	.46	.25
BERT	.40	.25
SBERT _{fine}	.55	.28

Table 4: Macro-averaged F1 across all questions.

ble without any fine-tuning of the model, we additionally report performance of directly using the **pretrained** SBERT model without any adaptation to the training data. We refer to this as **SBERT_{pre}** and denote the fine-tuned model with **SBERT_{fine}**.

6 Training on Synthetic Data

6.1 SRA vs. SRA-gen

In our first experiment, we compare scoring performance of models trained on the original SRA data vs. our generated data. To keep results comparable, we sample data from SRA-gen following the same label distribution as in SRA. To even out sampling effects, we repeat this 20 times and report the average performance. Aggregated results are shown in Table 4. Directly scoring the SRA data with DeepSeek-v2 performs at the level of the majority baseline. Due to the non-deterministic nature of the LLM, we run this scoring five times, obtaining a range of performance. We report the average here, but include detailed results in Figure 7 in the Appendix. In extreme cases, repeatedly administering the exact same prompt can produce macro-averaged F1 values ranging from below .3 to above .6. We also observe that the model almost exclusively labels answers as either *correct* or *partially correct*. Thus, the closeness to majority baseline performance is unsurprising.

For all three of the fine-tuned model types we test, training a model from SRA-gen performs slightly better than majority baseline and zero-shot LLM scoring. However, this performance is still a long way off from training on SRA. On SRA, the fine-tuned SBERT model gives the best results (.55 F1). Likely due to the limited training data, LR (.46) outperforms BERT (.40). Interestingly, the *pretrained* SBERT model (.44) also outperforms BERT on SRA, and does better than all other models on SRA-gen. Thus, we choose to break down results for individual questions for this model in Figure 2. To see variation between the 20 SRA-gen

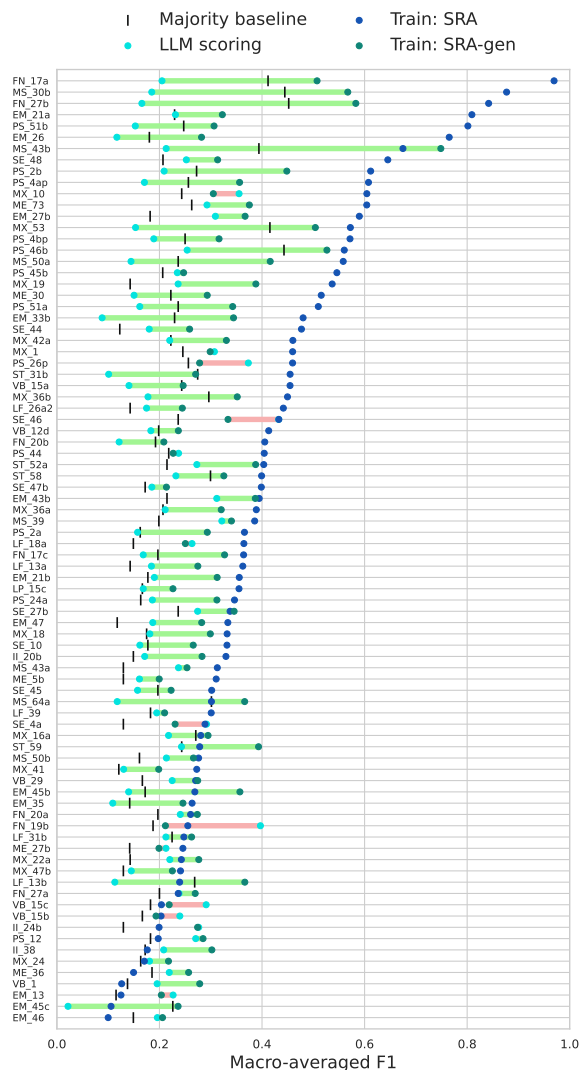


Figure 2: SBERT_{pre}: Performance per question.

samples we draw (Figure 8), and for the same results for the fine-tuned SBERT model (Figure 9) refer to the Appendix.

In Figure 2, we see that the pattern of using the LLM-generated data as training data being superior to zero-shot scoring with an LLM (green bars) is consistent across the majority of questions. For some questions, even the fine-tuned model is not doing much better than the majority baseline. Only for one of the questions for which a successful model can be learned on SRA do we see comparable performance when using SRA-gen as training data. Do however note that this only holds for the pretrained model. Fine-tuning SBERT on SRA outperforms training on SRA-gen for all questions.

6.2 Amount of Generated Training Data

As generating training data puts us at liberty to surpass the amount of data that is present in the

original dataset, we now explore how performance changes with larger amounts of synthetic data. We do this in a balanced fashion, i.e. with an equal amount of answers for each label, starting with just one per label and going up to 100. This means that we use training data ranging from as little as five to as many as 500 answers. We sample each amount 20 times, and report average, best and worst performance.

We again choose to do this analysis for the pre-trained SBERT model, as this is the model that gave the best performance on SRA-gen in the previous experiment. To see the full curve, refer to Figure 10 in the Appendix. From a low amount of training data onwards, performance remains on a consistent low level. The average performance is below the majority baseline performance of .21 macro-averaged F1, and even the best runs do just slightly better than this baseline. Thus, the relatively low performance on SRA-gen we saw in the previous experiment was not due to the limited amount of training data. Do also note that the balanced label distribution we enforce here leads to overall lower performance than what we had observed in the previous experiment, where training and test data shared the same label distribution.

7 Training on Cleaned Synthetic Data

Apart from the answers themselves, their labels are a crucial element of the generated data. While we are asking the LLM to generate answers that conform to a target label, there is no guarantee that they actually do. Thus, we perform manual annotation to assess whether the generated answers match the label they are supposed to belong to. Table 5 shows the three questions we select for this assessment.

We first manually clean the labels and then run scoring experiments that compare performance of training on the **as-generated** labels vs. the manually **cleaned** labels.

7.1 Label cleaning

As a first calibration round, three annotators (two authors of this paper and a research assistant) manually label the answers in SRA to make sure that there is substantial agreement with the original labels. Table 6 shows the Cohen’s Kappa (Cohen, 1960) we obtain.³ We also include agreement with

³Note that we believe to have found two mislabeled instances for question ME_27b, and one for question VB_1. We report agreement with the corrected labels.

ID	Question	# Answers				
		c.	p.c.	con.	irr.	n.-d.
ME_27b	<i>How can you use a magnet to find out if the key is iron or aluminum?</i>	22	12	1	4	1
PS_4bp	<i>Darla tied one end of a string around a doorknob and held the other end in her hand. When she plucked the string (pulled and let go quickly) she heard a sound. How would the pitch change if Darla made the string longer?</i>	24	0	10	6	0
VB_1	<i>How do you define a controlled experiment?</i>	21	3	1	14	1

Table 5: Questions chosen for manual annotation.

	ME_27b				PS_4bp				VB_1			
	G	R1	R2	R3	G	R1	R2	R3	G	R1	R2	R3
Gold	-	.77	.62	.84	-	.91	.91	.96	-	.79	.72	.91
R1	.77	-	.45	.69	.91	-	.91	.91	.79	-	.80	.83
R2	.62	.45	-	.62	.91	.91	-	.91	.72	.80	-	.68
R3	.84	.69	.62	-	.96	.91	.91	-	.91	.83	.68	-
Adj.	.88	.73	.66	.96	.95	.95	.95	.95	.83	.96	.84	.83

Table 6: Kappa agreement of our annotations with the labels in SRA (Adj. = adjudicated annotations).

the adjudicated labels, which we determined by taking the majority label. Where all three annotators had decided on different labels (two cases), the disagreement is resolved via discussion. Agreement between adjudicated labels and gold SRA labels ranges from .83 to .95. This shows that we can reliably annotate the data. Thus, we proceed with annotating the same prompts in SRA-gen.

For each of the three questions, we take 50 answers per label. This makes for a total of 250 answers per question, of which we randomize the order and hide the as-generated label. All three annotators now annotate the answers and we again derive adjudicated annotations by taking the majority label where possible. The remaining cases where all annotators disagree (12 for question ME_27b, 9 for question PS_4bp, 41 for question VB_1) are resolved through discussion. Table 9 in the Appendix shows kappa agreement for this round of annotation. Agreement is overall lower, as the LLM-generated data has substantially more variance than the original SRA data.

Label Accuracy With the manual label annotations we can now compute the accuracy for each label by comparing what the LLM was asked to generate with what the annotators agreed was actually generated. Table 7 shows these results, and

LLM Label	Human Label					Acc.
	corr.	part.	corr.	contr.	irr. non-d.	
ME_27b						
correct	12	13	3	22	0	.24
partially correct	13	17	4	16	0	.34
contradictory	3	2	18	26	1	.36
irrelevant	1	4	6	37	2	.74
non-domain	0	2	3	1	44	.88
PS_4bp						
correct	22	3	14	11	0	.44
partially correct	14	14	8	14	0	.28
contradictory	3	9	26	12	0	.52
irrelevant	0	0	3	47	0	.94
non-domain	6	2	2	11	29	.58
VB_1						
correct	23	26	0	1	0	.46
partially correct	26	18	3	3	0	.36
contradictory	0	13	14	23	0	.28
irrelevant	2	4	8	35	1	.70
non-domain	0	0	4	10	36	.72

Table 7: Adherence of the LLM to the label it was asked to generate answers for. Accuracy: the fraction of the 50 generated answers that does match the desired label.

Figure 3 compares label accuracies across questions. Only for one question and label (*irrelevant* for PS_4bp) nearly all generated answers conform to the desired label. Non-domain answers are only generated when the LLMs is asked for such: very rarely is an answer from a different label manually found to be *non-domain*. Overall, accuracy of *non-domain* and *irrelevant* answers is higher than for the other labels. Consistently, over half of the *correct*, *partially correct* and *contradictory* answers do not conform to the desired label. *Contradictory* answers are often determined to be *irrelevant*, and for VB_1 13 of them are even partially correct. *Correct* answers are regularly found to actually be *partially correct* or *irrelevant*. For PS_4bp, 14 correct answers are even found to in fact be *contradictory*. This is somewhat contrary to the general consensus that LLMs are doing well with answering correctly. It may however be due to a difficulty of having to come up with *multiple* answers in one go, i.e. *ten* correct answers instead of just one.

7.2 Model training with cleaned data

To assess the benefit of cleaning labels in SRA-gen, we can now compare the success of models trained on the as-generated vs. cleaned labels. Table 8 summarizes these results. When training on 40 instances from SRA-gen, we draw a sample with the same distribution as in SRA 20 times and report the average performance. Training on as-generated SRA-gen data consistently does worse when a bal-

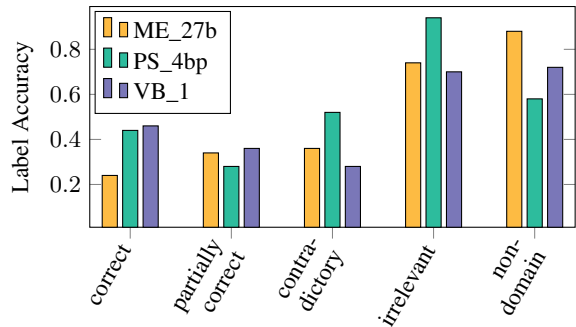


Figure 3: Accuracy of the labels in SRA-gen.

anced sample of 250 answers vs. just 40 answers is used to train. This is likely due to models benefiting from the matching label distribution in training and test data for the smaller sample.

The cleaned labels consistently lead to an increase in performance. For the 40 training answers, this increase is however much more subtle than for the full 250 SRA-gen answers. On this larger amount of training data, performance often reaches the level of training on the original SRA data. The SBERT model consistently gives the best performance, and is the only model for which training on the 250 cleaned SRA-gen answers consistently outperforms training on the 40 SRA answers.

Overall, our results demonstrate that the LLM-generated answers themselves do carry enough meaning to inform a model, but that manual cleaning is necessary to remove noise in their labels. As we have seen the label distribution in the training data to affect model performance, the comparison between the 250 as-generated vs. cleaned SRA-gen answers is however not entirely ‘fair’: While the SRA-gen data was drawn with a balanced distribution of 50 answers per label, this distribution has shifted once the labels were cleaned. We therefore take a look at the performance of balanced sampling with as-generated vs. cleaned SRA-gen data in Figure 4. Since the fine-tuned SBERT model gives the best performance on cleaned SRA-gen data, we choose this model for this analysis. Do note that we can only compute the curve for the cleaned data up to 28 answers per label, as the total number of answers for the most infrequent label limits our ability to draw a balanced sample. For training with 5, 10, 15, 20 and 25 answers per label, we draw 20 training samples each and report best, average and worst performance.

For all three questions, the as-generated labels constantly lead to low average performance lev-

Data	LR					BERT					SBERT _{pre}					SBERT _{fine}				
# train	40	40	40	250	250	40	40	40	250	250	40	40	40	250	250	40	40	40	250	250
Generated	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓
Cleaned	-	✗	✓	✗	✓	-	✗	✓	✗	✓	-	✗	✓	✗	✓	-	✗	✓	✗	✓
ME_27b	.34	.16	.19	.09	.32	.36	.10	.17	.07	.32	.33	.20	.31	.26	.29	.33	.12	.29	.08	.41
PS_4bp	.58	.25	.25	.00	.46	.49	.29	.27	.03	.47	.73	.32	.45	.00	.21	.82	.26	.61	.07	.91
VB_1	.33	.13	.30	.06	.37	.33	.24	.29	.09	.31	.33	.28	.25	.16	.46	.35	.22	.31	.02	.41
Avg.	.42	.18	.25	.05	.38	.39	.21	.25	.06	.37	.46	.27	.34	.14	.32	.50	.20	.40	.06	.58

Table 8: Effect of cleaning the LLM labels via manual annotation (macro-averaged F1). For BERT and SBERT, results are averaged across three runs for a more reliable performance estimate. ‘Generated’ denotes whether we are training on SRA (✗) or SRA-gen (✓). ‘Cleaned’ indicates if we are using the as-generated (✗) or cleaned (✓) labels.

els of below .2 F1. With the cleaned labels, performance rises once more data is added, and the curves indicate that it might rise further if there was more data available. This controlled comparison thus confirms the beneficial effect of manually cleaning the labels.

8 Conclusion

We generate answers to the questions in the SRA dataset with an LLM. Using these answers as training data leads to relatively poor performance. Directly scoring the SRA data with an LLM even performs slightly worse, showing an inability of the model to reliably apply the 5-way label scale. This is supported by our analysis of the extent to which the LLM sticks to the label we ask it to generate answers for. Up to 75% of the answers the model was asked to generate for a specific label were found not to conform to this label. Training a model with manually relabeled generated data demonstrates the detrimental effect of the noisy labels: With cleaned labels, model performance increases substantially, reaching a comparable level to training on the original SRA data - albeit at the demand of larger volumes of training data. In light of our analysis of the lexical diversity in SRA vs. SRA-gen, this is likely due to diverging content in SRA vs. SRA-gen. Thus, more SRA-gen data is needed to sufficiently cover the content of SRA.

With a similarity based scoring model, training on the larger sample of generated data even consistently leads to superior performance over training on the small amount of available original SRA data. One benefit of the similarity-based model might be that one highly similar answer with the correct label suffices for the model to correctly label an answer of interest.

In conclusion, one *can* overcome the cold-start

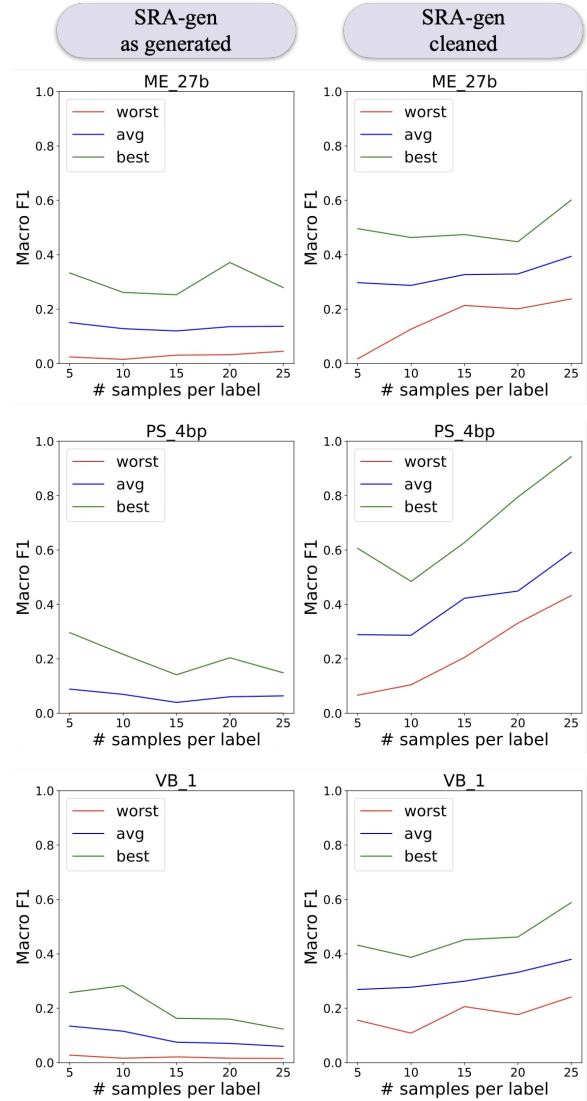


Figure 4: SBERT_{fine}: Balanced sampling of SRA-gen with as-generated (left) vs. cleaned (right) labels.

problem with the help of an LLM in the sense of not having to collect data from real students, but not without the manual effort of labeling the generated answers. Future work could explore automatic

cleaning of the generated data to alleviate the manual labeling effort. While we saw limited success in preliminary experiments, future work could also quantify the effect of using few-shot prompting, both in zero-shot labeling and generating answers.

Limitations

While our results provide interesting insights into the possibility of generating training data with an LLM, there are a number of limitations to our findings. First, we only experiment with one LLM. Other LLMs may behave differently, which limits our conclusions to DeepSeek-v2. Even within the realm of prompting an LLM, the precise choice of prompt can have substantial impact (Sclar et al., 2024). While we did carefully craft our prompts, subtle changes to the wording may affect results. Within the prompt design, a key aspect might be the amount of answers the model is asked to generate in one go. We always asked for ten answers, but results may differ if the model were asked to generate just one or even all 500 answers at once.

Even beyond model choice and prompt design, model parameters will affect results. We left these untouched, but varying the temperature will affect both answer generation and scoring ability of the model.

Ethical Considerations

In considering the use of generated training data for model training, one has to be cautious about the normative language LLMs produce. An inability to produce sufficiently ‘student-like’ language may lead to a model with inferior performance on real student answers that deviate from language norms. Since content scoring is however less about language form and more about content, this should not affect the score of an answer.

Automated scoring of student answers in general is not without ethical and legal issues. It is high-risk as per the European Union AI Act, and LLM use poses ‘systematic risks’.

A main concern of LLMs and deep learning in general is a lack of transparency. This is somewhat alleviated by the use of an LLM to generate synthetic answers as opposed to using it to directly score student answers. Still, our work shows that based on the synthetic answers it is again most successful to apply deep learning. This in turn is much less transparent than the use of a shallow learning method such as logistic regression - which we test

as well, but find to perform worse. However, the deep learning model we find to perform best operates in a similarity-based fashion. Thus, it at least allows backtracking to the reference answers that lead to a predicted score.

Acknowledgements

We thank Kristina Spenner for her valuable assistance with data annotation and experimental work. We also thank the reviewers for their insightful comments and constructive feedback.

References

- Lorin W Anderson and Lauren A Sosniak. 1994. *Bloom’s taxonomy*. Univ. Chicago Press Chicago, IL).
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2022. [Similarity-based content scoring - how to make S-BERT keep up with BERT](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 118–123, Seattle, Washington. Association for Computational Linguistics.
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2023. [Similarity-based content scoring-a more classroom-suitable alternative to instance-based scoring?](#) In *Findings of the association for computational linguistics: Acl 2023*, pages 1892–1903.
- Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2025. [Distractor Generation for Multiple-Choice Questions with Predictive Prompting and Large Language Models](#). In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 48–63, Cham. Springer Nature Switzerland.
- 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.
- Derek Chen, Celine Lee, Yunan Lu, Domenic Rosati, and Zhou Yu. 2023. [Mixture of Soft Prompts for Controllable Data Generation](#). *arXiv preprint*. ArXiv:2303.01580 [cs].
- Jonathan H Choi, Kristin E Hickman, Amy B Monahan, and Daniel Schwarcz. 2021. Chatgpt goes to law school. *J. Legal Educ.*, 71:387.
- Jacob Cohen. 1960. [A Coefficient of Agreement for Nominal Scales](#). *Educational and Psychological Measurement*, 20(1):37–46. [_eprint: https://doi.org/10.1177/001316446002000104](https://doi.org/10.1177/001316446002000104).

- DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. [DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model](#). *arXiv preprint*. ArXiv:2405.04434 [cs].
- Tu Anh Dinh, Carlos Mullov, Leonard Bärmann, Zhaolin Li, Danni Liu, Simon Reiß, Jueun Lee, Nathan Lerzer, Jianfeng Gao, Fabian Peller-Konrad, Tobias Röddiger, Alexander Waibel, Tamim Asfour, Michael Beigl, Rainer Stiefelhagen, Carsten Dachsbacher, Klemens Böhm, and Jan Niehues. 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, Miami, Florida, USA. 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.
- 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.
- Jiahui Gao, Renjie Pi, LIN Yong, Hang Xu, Jiacheng Ye, Zhiyong Wu, WEIZHONG ZHANG, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023. [Self-guided noise-free data generation for efficient zero-shot learning](#). In *The Eleventh International Conference on Learning Representations*.
- 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.
- Mohammad Hassany, Peter Brusilovsky, Jaromir Savelka, Arun Balajiee Lekshmi Narayanan, Kamil Akhuseyinoglu, Arav Agarwal, and Rully Agus Hendrawan. 2025. [Generating Effective Distractors for Introductory Programming Challenges: LLMs vs Humans](#). In *Proceedings of the 15th International Learning Analytics and Knowledge Conference, LAK '25*, pages 484–493, New York, NY, USA. Association for Computing Machinery.
- Mingxin Liu, Tsuyoshi Okuhara, XinYi Chang, Ritsuko Shirabe, Yuriko Nishiie, Hiroko Okada, and Takahiro Kiuchi. 2024. [Performance of ChatGPT Across Different Versions in Medical Licensing Examinations Worldwide: Systematic Review and Meta-Analysis](#). *Journal of Medical Internet Research*, 26:e60807.
- Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. [Generating Training Data with Language Models: Towards Zero-Shot Language Understanding](#). *Advances in Neural Information Processing Systems*, 35:462–477.
- 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](#). *arXiv preprint*. ArXiv:2303.08774 [cs].
- Saba Rahimi, Tucker Balch, and Manuela Veloso. 2023. [Exploring the Effectiveness of GPT Models in Test-Taking: A Case Study of the Driver’s License Knowledge Test](#). *arXiv preprint*. ArXiv:2308.11827 [cs].
- Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Dragan Gašević, Geber Ramalho, and Rafael Ferreira Mello. 2024. [Assessing the quality of automatic-generated short answers using GPT-4](#). *Computers and Education: Artificial Intelligence*, 7:100248.
- 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 *The Twelfth International Conference on Learning Representations*.
- Shiva Taslimipour, Luca Benedetto, Mariano Felice, and Paula Buttery. 2024. [Distractor generation using generative and discriminative capabilities of transformer-based models](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5052–5063, Torino, Italia. ELRA and ICCL.
- Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. [ZeroGen: Efficient Zero-shot Learning via Dataset Generation](#). *arXiv preprint*. ArXiv:2202.07922 [cs].
- Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. [Large language model as attributed training data generator: A tale of diversity and bias](#). In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Torsten Zesch, Andrea Horbach, and Fabian Zehner. 2023. To score or not to score: Factors influencing performance and feasibility of automatic content scoring of text responses. *Educational Measurement: Issues and Practice*, 42(1):44–58.

Appendix

This appendix contains some supplementary material to increase transparency of our experiments. It includes the prompt used to generate SRA-gen in Figure 5 and the prompt used to score the answers in SRA in Figure 6. The main paper contains the agreement we achieve in labeling the original SRA data in Table 6. Here, we include the same statistics for our annotation of the generated data in Table 9. We also include some more detailed results of our scoring experiments. Due to the non-deterministic nature of the LLM, repeated administration of the same prompt leads to differing results. Thus, Figure 7 depicts the variation in performance when administering the same prompt to the same model five times. Figures 8 (pretrained SBERT) and 9 (SBERT) show question-wise results for scoring based on SRA-gen vs. SRA. Finally, Figure 10 shows performance of training the pretrained SBERT model with balanced samples of SRA-gen.

	LLM	R1	R2	R3	Adjudicated
ME_27b					
LLM	-	.49	.19	.36	.39
R1	.49	-	.32	.48	.62
R2	.16	.32	-	.48	.63
R3	.36	.48	.48	-	.81
Adjudicated	.39	.62	.63	.81	-
PS_4bp					
LLM	-	.55	.33	.39	.45
R1	.55	-	.59	.59	.77
R2	.33	.59	-	.56	.75
R3	.39	.59	.56	-	.75
Adjudicated	.45	.77	.75	.75	-
VB_1					
LLM	-	.63	.26	.22	.34
R1	.63	-	.41	.31	.55
R2	.26	.41	-	.46	.69
R3	.22	.31	.46	-	.59
Adjudicated	.34	.55	.69	.59	-

Table 9: Kappa agreement of our annotations with the labels in SRA-gen.

```

<purpose>
You are a school teacher.
Your students are going to answer the following question:
{question}

You are now thinking about possible answers students could give.

[LABEL_INSTRUCTIONS]
</purpose>
<format_rules>
Use markdown output and put each correct answer as a single bullet point.
Keep the answers as short as possible. A maximum of 20 words per answer.
</format_rules>
<output>
Create 10 [correct/partially correct or incomplete/contradictory/irrelevant/non domain]
responses following the given rules.
</output>

LABEL_INSTRUCTIONS={

CORRECT: Generate a list of 10 possible correct answers.
That is the important part, generating that list of exactly 10 answers!

PARTIALLY_CORRECT_INCOMPLETE: Generate a list of 10 possible partially correct
or incomplete answers. Partially correct or incomplete means that the student answer is
a partially correct answer containing some but not all information from the reference
answer. The important part is to generate a list of 10 student answers belonging to that
category (partially correct incomplete)!

CONTRADICTIONARY: Generate a list of 10 possible contradictory answers. That means that
the given answers are not correct and explicitly contradict the correct answer. The
important part is to generate a list of 10 answers belonging to that contradictory
category!

IRRELEVANT: Generate a list of 10 possible irrelevant answers. Irrelevant means that
the student answer is talking about domain content but not providing the necessary
information to be correct. The important part is to generate a list of 10 student answers
belonging to that irrelevant category!

NON_DOMAIN: Generate a list of 10 possible 'non domain' answers. 'Non domain' means that
the student utterance does not include domain content, e.g., "I don't know", "what the
book says", "you are stupid". The important part is to generate a list of 10 student
answers belonging to that category!}

```

Figure 5: Prompt used to generate training data. We follow the

```
<purpose>
You are a school teacher.
A student has answered the following question:
{question}
This is the answer the student gave:
{answer}
You now have to score this answer.
These are the possible scores:
Correct: A correct answer to the question.
Partially correct or incomplete: This means that the student answer is a partially
correct answer that contains some but not all necessary information.
Contradictory: This means that the student answer is not correct and explicitly
contradicts the correct answer.
Irrelevant: This means that the student answer is talking about domain content but
not providing the necessary information to be correct.
Non-domain: This means that the student answer does not include domain content, e.g.,
"I don't know", "what the book says", "you are stupid".
</purpose>
<format_rules>
Only output the score.
</format_rules>
<output>
Decide on the score of the student answer.
</output>
```

Figure 6: Prompt used to score answers with the LLM.

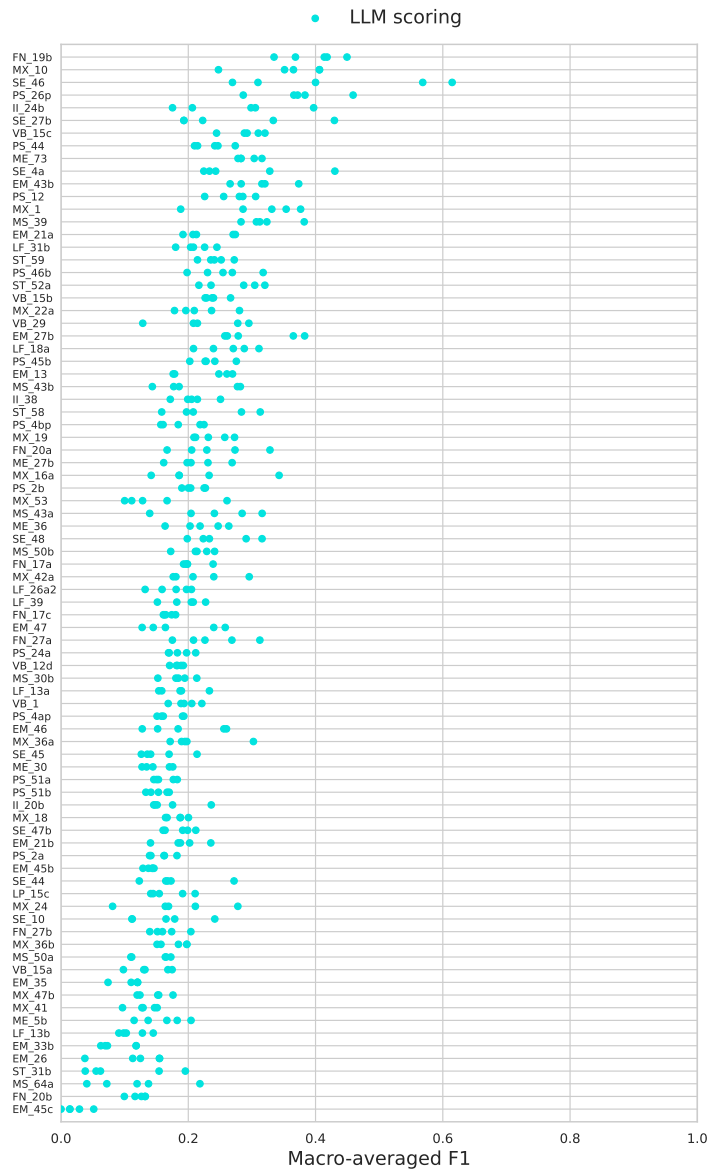


Figure 7: Performance variation across five runs of scoring the answers using an LLM.

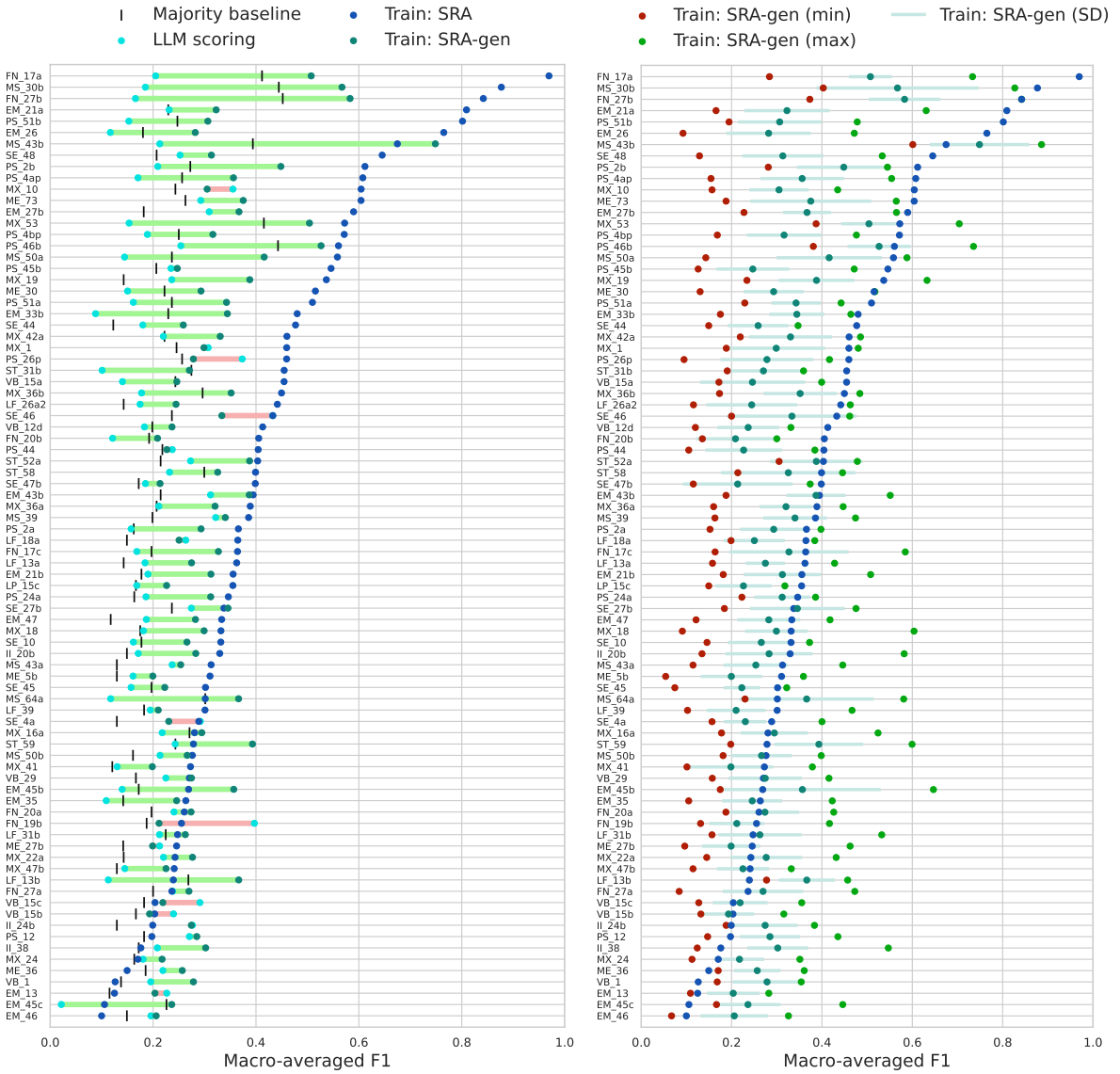


Figure 8: SBERT_{pre} performance variation across 20 samples of generated training data that follow the same label distribution as the original SRA data. Left: Comparison of the average performance to directly scoring the data with an LLM. Right: Detailed results of the best, average and worst sample.

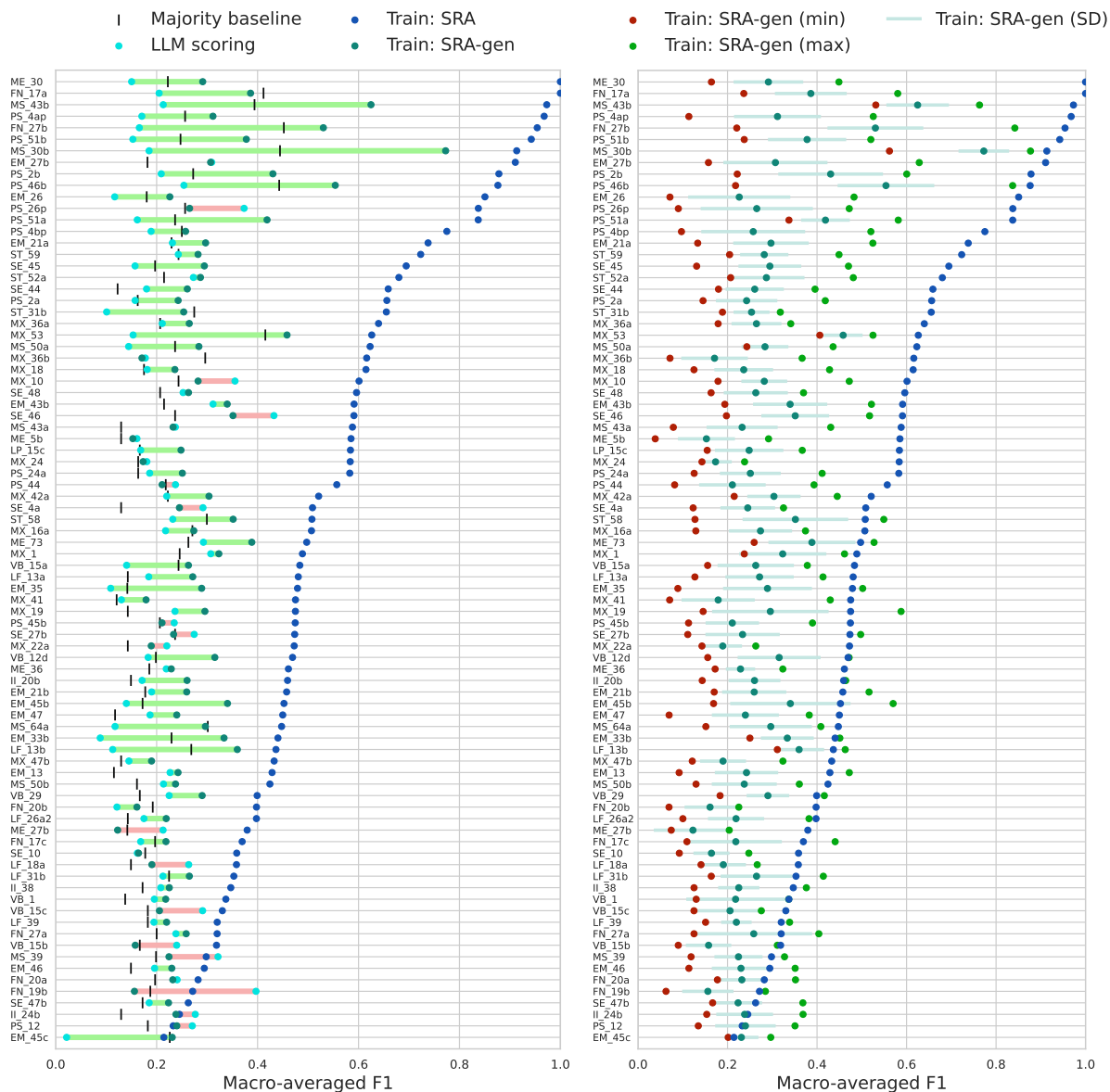


Figure 9: SBERT_{fine} performance variation across 20 samples of generated training data that follow the same label distribution as the original SRA data. Left: Comparison of the average performance to directly scoring the data with an LLM. Right: Detailed results of the best, average and worst sample.

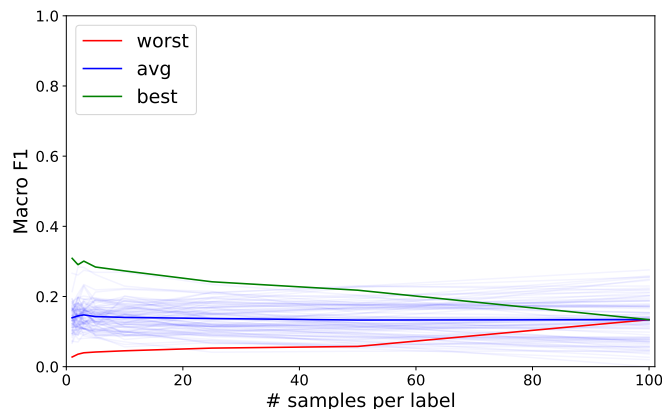


Figure 10: Average performance of SBERT_{pre} when using a balanced sample of SRA-gen training data. Light blue lines show average results for individual questions.

Transformer Architectures for Vocabulary Test Item Difficulty Prediction

Lucy Skidmore, Mariano Felice, Karen J. Dunn
English Language Research, British Council, UK
name.surname@britishcouncil.org

Abstract

Establishing the difficulty of test items is an essential part of the language assessment development process. However, traditional item calibration methods are often time-consuming and difficult to scale. To address this, recent research has explored natural language processing (NLP) approaches for automatically predicting item difficulty from text. This paper investigates the use of transformer models to predict the difficulty of second language (L2) English vocabulary test items that have multilingual prompts. We introduce an extended version of the British Council’s Knowledge-based Vocabulary Lists (KVL) dataset, containing 6,768 English words paired with difficulty scores and question prompts written in Spanish, German, and Mandarin Chinese. Using this new dataset for fine-tuning, we explore various transformer-based architectures. Our findings show that a multilingual model jointly trained on all L1 subsets of the KVL achieve the best results, with analysis suggesting that the model is able to learn global patterns of cross-linguistic influence on target word difficulty. This study establishes a foundation for NLP-based item difficulty estimation using the KVL dataset, providing actionable insights for developing multilingual test items.

1 Introduction

Calibrating the difficulty of test items is a core aspect of language assessment design, ensuring that tests are fair, consistent, and aligned with learner proficiency. Traditionally, this calibration relies on pre-testing large item samples or expert judgment, which are expensive and time-consuming. Consequently, there is an increasing interest in automating item calibration using machine learning methods (Yancey et al., 2024; Yaneva et al., 2024), which offer greater scalability, efficiency, and consistency, and can be more easily integrated into item development pipelines.

While transformer-based encoder models such as BERT (Devlin et al., 2019) have been successfully applied to question difficulty estimation from text (QDET) in domains related to content knowledge assessment, approaches in language assessment — where difficulty is more closely tied to the linguistic properties of the item — still largely rely on hand-crafted features (Alkhuzaey et al., 2024). This is particularly true in QDET for L2 English vocabulary items, which commonly rely on small datasets that are not suitable for fine-tuning transformers (Benedetto et al., 2023).

To address this gap, our paper introduces a new multilingual resource for vocabulary QDET: an extended version of the British Council’s Knowledge-based Vocabulary Lists (KVL), containing 6,768 English vocabulary items paired with difficulty scores and prompts in Spanish, German, and Mandarin Chinese. We use the KVL to fine-tune various transformer-based architectures for vocabulary test item difficulty prediction, leveraging its unique structure to provide insights on how best to model vocabulary difficulty in multilingual settings. Our exploratory work serves as a benchmark for future development of generalisable, L1-agnostic models for explainable item calibration.

Our paper begins with an overview of how the KVL has been extended and adapted for NLP-based applications. This is followed by a summary of the latest research in two domains that this study intersects: question difficulty estimation from text and lexical complexity prediction. Next, we outline the aims of the study, providing the motivation and context for the experimental design. The transformer-based architectures that we investigated are outlined, including the procedures followed for model selection and fine-tuning. We present findings from model performance evaluations, an ablation study and an error analysis. Finally, the paper ends with a discussion of the results and outlines directions for future work.

2 The Knowledge-based Vocabulary Lists

The Knowledge-based Vocabulary Lists (KVL) (Schmitt et al., 2021, 2024) were the outcome of a collaborative research project between the British Council and researchers from the University of Nottingham, University of Innsbruck and Waseda University. Productive English language word knowledge was assessed using prompts designed to test form-based recall of individual lemmas in a translation format (cf. Laufer and Goldstein, 2004). Items comprising an L1 translation of the English target word, plus contextualising sentences were developed separately in three L1s (Mandarin Chinese, German, Spanish) to create a bank of 7,679 items for each language. Participants were required to input the remainder of the word in English, as per this example from the Spanish language version¹:

```
casa Vivo en una casa grande que tiene tres
dormitorios.
h _ _ _ _
```

During a period between late-2018 to mid-2020, 3.3 million responses were collected from over 100,000 respondents via crowdsourcing. An online platform, promoted across the British Council’s social media channels, presented participants with blocks of ten random items stratified by target word frequency. Feedback was given after each block, and participants were encouraged to complete more items to “beat their best”, as an example of game-based data collection (Kim et al., 2024).

Difficulty estimates were derived separately for each L1 subset of the data, using random-item-random-person (RPRI) Rasch models (De Boeck, 2008) built within a generalised linear mixed model (GLMM) framework (Dunn, 2024). Original KVL project outputs used these estimates to create a rank-order list of the top 5,000 words for each L1.

For this research, we use the existing 5000 items in the KVL and publicly release an additional 1,768 English vocabulary test items for each L1. This extended dataset contains 20,304 items in total (6,768 per L1) and is divided into 80% train (16,242 items), 10% development (2,031 items) and 10% test (2,031 items) sets².

¹The German and Chinese versions had similar, yet distinct prompts, for example in German: “Haus Ich wohne in einem Haus mit Garten.” And in Chinese: “房子 我买了一座房子。”

²<https://www.britishcouncil.org/data-science-and-insights/resources>

3 Related Work

3.1 Question Difficulty Estimation from Text

Question difficulty estimation from text (QDET) concerns the prediction of test item difficulty based solely on its textual features. There is growing interest in using QDET for high stakes assessment calibration, given its efficiency and scalability compared to traditional methods (AIKhuzaey et al., 2024). The majority of work in this area explores supervised approaches to QDET, with transformer-based encoder models achieving the best results in recent years (Gombert et al., 2024; Yaneva et al., 2024). There is also a growing interest in unsupervised approaches to the task, using generative models as ‘test-takers’, extracting their uncertainty as a proxy for human difficulty (Loginova et al., 2021; Uto et al., 2024; Zotos et al., 2025).

Research related to vocabulary-based QDET, however, is relatively limited. Most prior approaches to this task use hand-crafted linguistic features (such as word frequency and word length) as inputs to predictive models (Suyong and Hua, 2018; Settles et al., 2020), with other approaches incorporating embeddings such as word2vec (Ehara, 2018) and GloVe (Susanti et al., 2020). Beyond word-based features, contextual factors such as the similarity between correct answers and distractors in multiple choice vocabulary tests (Susanti et al., 2017, 2020) as well as semantic descriptors from dictionary entries of target words (Nakanishi et al., 2012), have also had limited exploration.

3.2 Lexical Complexity Prediction

Lexical complexity prediction (LCP) is a subfield of complex word identification (CWI), which concerns the automatic detection of complex words from text, primarily for the purpose of text simplification. LCP extends the binary classification used for CWI to form a regression problem, with the goal of predicting a continuous ‘complexity’ value for a given word. These values are domain-specific, and can range from crowd-sourced perceived complexity ratings to morphosyntactically derived features (North et al., 2023). Different to vocabulary QDET, the input text used for LCP typically involves predicting the complexity of a word in context. Given the format of the KVL dataset, the task of LCP aligns more closely with our investigation than much of the previous work in vocabulary QDET.

The most successful approaches to LCP to date make use of transformer-based architectures (Bani Yaseen et al., 2021; Kelious et al., 2024a). Particularly relevant to this work, however, are investigations into multilingual applications of LCP. Sheang (2019) showed that a multilingual CNN model trained jointly on word embeddings and linguistic features of Spanish, German and English datasets led to improved performance of prior models for Spanish and German. Similarly, Finnimore et al. (2019) found that jointly training models with languages from the same family improved cross-lingual CWI. Zaharia et al. (2020) experimented with multilingual transformers for cross-lingual CWI, showing that XLM-RoBERTa performs best for unseen German or French target words. More recently, LLMs have been explored for unsupervised multilingual LCP, however these approaches did not outperform supervised transformer-based equivalents (Kelious et al., 2024b).

4 Research aims

As described above, the KVL dataset is unique in that it contains multilingual test items (comprising L1 source word, L1 context and EN clue) for the same set of English target words across three L1s. To explore this multi-faceted structure, we defined four transformer-based models for experimentation: (1) individual monolingual models for each test item component; (2) ensembles combining these component-specific models; (3) multilingual models fine-tuned on the full test item text separately for each L1; and (4) a single multilingual model trained on the full test item text across all L1s.

Comparing the performance of these models allowed for multiple avenues of investigation: the influence of test item components on model predictions, the suitability of monolingual versus multilingual models and training data, as well as the effectiveness of different architectures for capturing cross-component and cross-lingual interactions within items. In addition to overall model performance, we were also interested in whether model error revealed potential biases—such as systematic under- or overestimation for particular items. These areas of interest were distilled into three primary research questions for the study:

- How accurately can different transformer-based model architectures predict vocabulary item difficulty for the KVL dataset?

- How do the individual components of the test item contribute to the models’ predictions?
- Is there any systematic bias contributing to errors in the best-performing model?

5 Modelling setup

Item difficulty prediction was modelled as a regression task. The target values for prediction were transformations of the GLMM item-level conditional modes. These were inverted to reflect item difficulty (as opposed to ‘item easiness’ in the original study) and scaled to values between zero and one. Models were fine-tuned with mean squared error (MSE) as the loss function. As the KVL were originally designed for ranking vocabulary difficulty, Spearman’s rank correlation coefficient (RHO) was used as the main model evaluation metric. The root mean squared error (RMSE) metric was also calculated to evaluate model fit. Where relevant, statistical significance tests of the models were carried out via bootstrap, using 10,000 iterations and Bias-Corrected and Accelerated (BCa) intervals (Efron, 1987).

For the multilingual models, the structure of the input text begins with the question content (L1 source word, L1 context, EN clue), in the same order as it is presented in the vocabulary test items, followed by the target answer (EN target word). Each part of the input text was delineated with the models’ pre-defined separation token, as shown in the example input text below:

```

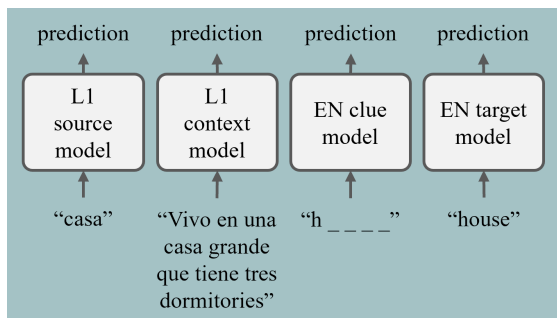
casa [SEP] Vivo en una casa grande que
tiene tres dormitorios. [SEP] h_____
[SEP] house

```

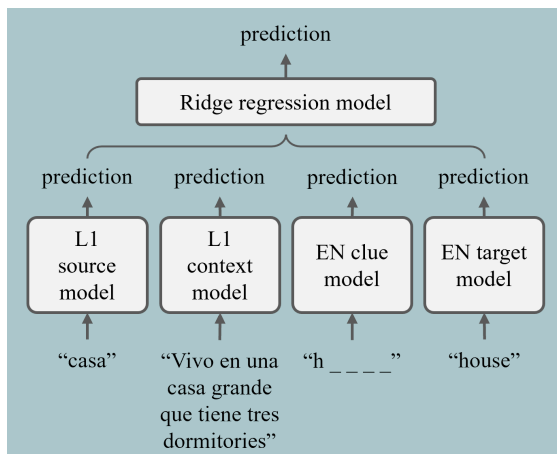
For the ensemble models, each part of the text was processed and tokenised separately.

5.1 Model architectures

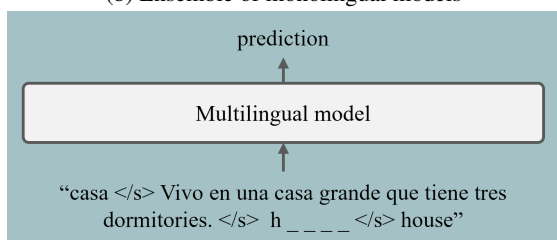
Figure 1 provides an overview of the different architectures explored for this study. For the individual monolingual models and multilingual models, the 768-dimensional embedding of the first token (<s> for RoBERTa-based models and [CLS] for BERT-based models) from the final hidden layer is passed through a dropout layer followed by a single linear layer (the regression head) to predict the difficulty score. For the monolingual ensemble models, the predictions from the individual component models are stacked together and passed through a Ridge re-



(a) Individual component models



(b) Ensemble of monolingual models



(c) Multilingual model

Figure 1: Model architectures for transformer-based approaches.

gression model. Each L1-specific ensemble learns a distinct weighting scheme for the predictions.

5.2 Model selection

Multiple pre-trained transformer models available through the Hugging Face platform³ were considered for use in the architectures explored in this research. Preliminary model evaluation was carried out in order to select the best model for each of the architectures. Using a fixed set of hyperparameters, candidate models for each of the architectures described above were evaluated. The models were fine-tuned with the train set, and tested with the development set, reporting the

³<https://www.huggingface.co>

RMSE and RHO of the predictions for the best model after five epochs. From this investigation, the following models were selected for further experimentation (see Table A.2 in the Appendix for results for all candidate models):

Multilingual model: XLM-RoBERTa (Conneau et al., 2020) is pre-trained on text from 100 languages using a large-scale CommonCrawl-based corpus. It employs SentencePiece tokenisation and is trained with a masked language modelling (MLM) objective. XLM-RoBERTa has been shown to outperform other multilingual transformer models in multiple NLP tasks, including cross-lingual complex word identification (Zaharia et al., 2020).

Monolingual English models: BERT (Devlin et al., 2019) is pre-trained on English text from BooksCorpus and Wikipedia using a WordPiece tokeniser. It learns contextualised word representations through masked language modelling (MLM) and next sentence prediction (NSP).

Monolingual L1 models: BERT models pre-trained for Spanish⁴ (Cañete et al., 2020), German⁵ (Chan et al., 2020) and Chinese⁶ (Devlin et al., 2019). These models follow the BERT architecture and are pre-trained using equivalent L1 texts. For consistency, where relevant we use the cased, base model versions of each of the models listed above.

5.3 Model fine-tuning

Each of the models selected for experimentation was tuned for optimal hyperparameters. With a batch size fixed at 32 and dropout rate set to model defaults (0.1 for all models), Optuna⁷, a hyperparameter optimisation framework for Python, was used to search for the best learning rate, weight decay and warm up ratio for each of the models. The models were fine-tuned with the train set, and evaluated with the development set, reporting the RMSE and RHO of the predictions for the best model after five epochs. See Table A.3 in the Appendix for the best hyperparameters for each model.

Using the optimised hyperparameters, four sets of models were fine-tuned on the train and development sets and evaluated on the test set. These included: (1) individual models for each test item component (L1 source word, L1 context, EN

⁴<https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased>

⁵<https://huggingface.co/deepset/gbert-base>

⁶<https://huggingface.co/google-bert/bert-base-chinese>

⁷<https://optuna.readthedocs.io/en/>

Model	ES		DE		CN		L1 average	
	RMSE	RHO	RMSE	RHO	RMSE	RHO	RMSE	RHO
L1 source word	0.156	0.522	0.142	0.565	0.130	0.561	0.143	0.550
L1 context	0.168	0.432	0.159	0.424	0.137	0.507	0.155	0.455
EN clue	0.169	0.400	0.155	0.389	0.139	0.477	0.154	0.422
EN target word	0.145	0.633	0.135	0.625	0.111	0.727	0.130	0.661

Table 1: RMSE and Spearman’s Rho for individual component models evaluated on the KVL test set.

Model	ES		DE		CN		L1 average	
	RMSE	RHO	RMSE	RHO	RMSE	RHO	RMSE	RHO
EN target word	0.145	0.633	0.135	0.625	0.111	0.727	0.130	0.661
Monolingual ensemble	0.142	0.646	0.129*	0.651	0.106*	0.747*	0.126	0.681
Multilingual (L1-specific)	0.126*	0.734*	0.116*	0.776*	0.108	0.725	0.117	0.745
Multilingual (all-in-one)	0.116*	0.775*	0.108*	0.793	0.097*	0.785*	0.107	0.785

*Statistically significant improvement in performance compared to the prior model in the table. Significance testing was not applied to L1 average results.

Table 2: RMSE and Spearman’s Rho results for the transformer-based architectures evaluated on the KVL test set.

clue, and EN target word), fine-tuned separately for each L1 subset; (2) monolingual ensembles fine-tuned per L1 subset; (3) multilingual models fine-tuned per L1 subset (L1-specific); and (4) an ‘all-in-one’ multilingual model fine-tuned on all L1 subsets combined.

6 Results

6.1 Model performance

Table 1 reports the individual models’ performance for each test item component. The EN target word model yields the highest scores across all L1s for both RMSE and RHO, and is particularly high for the Chinese subset, with a RHO of 0.73. Overall, the next best predictor is the L1 word (average correlation: 0.55), followed by L1 context (0.46) and EN clue (0.42).

Table 2 presents results for the monolingual ensemble⁸, L1-specific multilingual and all-in-one multilingual models evaluated on the KVL test set, alongside the individual EN target word model serving as a baseline. Results marked with an asterisk showed significant improvement in performance compared to the prior model in the table. On average, the ensemble architecture offers a small improvement in performance over the EN target word model for both RMSE and RHO, however the increase is not statistically significant for either metric in the ES subset, and not significant for RHO

in the DE subset. The L1-specific model considerably outperforms the ensemble approach for the ES and DE subsets, with RHO increasing from 0.65 to 0.73 and 0.78, respectively. This performance increase is not seen for the CN subset, which shows a marginally poorer but non-significant performance difference for RMSE and RHO. The all-in-one model achieves the best L1 average performance in both RMSE and RHO as well as demonstrating the most consistent RHO across L1 subsets, with scores of 0.78 for ES, 0.79 for DE, and 0.79 for CN. For the DE subset, however, this performance increase is not significantly higher than the L1-specific model for RHO.

6.2 Influence of test item component

An ablation study of the test item components was conducted for the ensemble, L1-specific and all-in-one models. Single components were systematically removed from the models, in order to investigate their influence on model performance. The models were fine-tuned using the train and development set and evaluated on the test set. The full model results for RMSE and RHO coefficients with statistical significance are reported in Table A.4 in the Appendix.

Figure 2 reports the relative percentage change in RHO for each of the models after removing individual components, across the L1 subsets. Statistically significant differences in model performance are marked with an asterisk. For the monolingual ensemble models, we can see that removing the EN

⁸The learned Ridge regression weights for each component model in the ensembles can be found in Table A.1 in the Appendix.

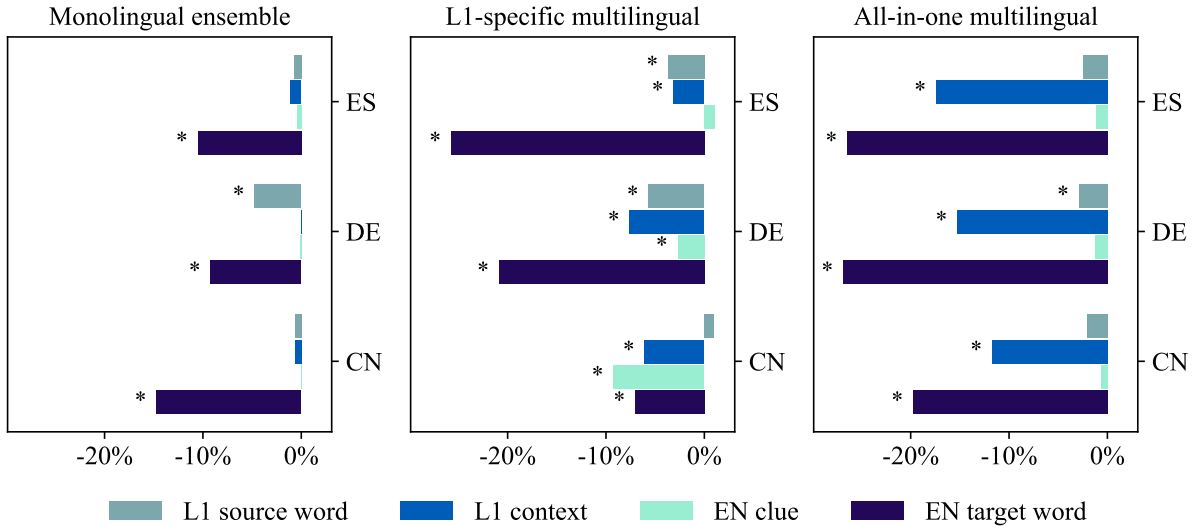


Figure 2: Relative percentage change in Spearman’s Rho after individual component removal.

target word results in the largest decrease in performance – around 10% for ES and DE and 15% for the CN subset. The removal of other components have no statistically significant impact, with the exception of the DE subset which shows a small 5% degradation for the L1 word.

For the L1-specific models, we can see a more varied distribution of impact for each of the components. Statistically significant degradation of performance is seen in all three L1s for L1 context, ES and DE for L1 word and DE and CN for the EN clue. The results for the EN target word in the CN subset are notably different to those of the ES and DE, with a much lower degradation in performance after its removal (around 7%, compared to between 20-25% for DE and ES, respectively). The EN clue is in fact more impactful than the EN word in this case, showing a statistically significant 9% reduction in performance after its removal.

Looking to the all-in-one multilingual model, we can see that the ablation results begin to generalise, showing a similar pattern of impact across the L1 subsets. Results from the statistical significance tests show that removing the EN clue had no significant impact on the all-in-one model performance for any of the L1 subsets, and the removal of the L1 word has no significant impact on model performance for the ES and CN subsets.

6.3 Error analysis

Figure 3 shows the model residuals plotted against the difficulty values for each L1 subset tested on the best performing model, the all-in-one multilin-

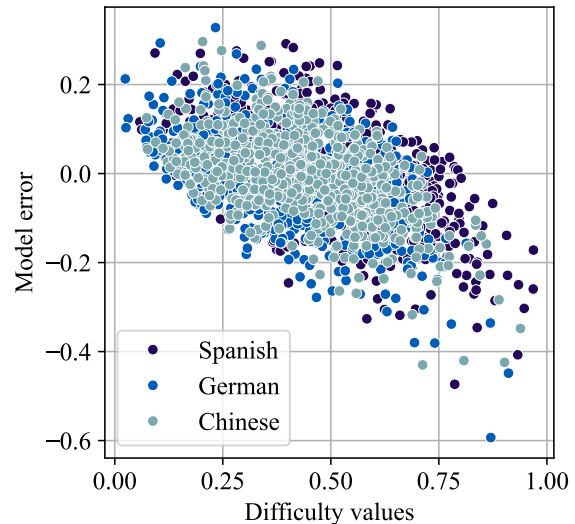


Figure 3: Error plot for the all-in-one multilingual model across L1s.

gual model. The graph shows that the majority of predictions fall within a range of -0.2 and +0.2 of the difficulty values. Across all L1 subsets, the model tends to underestimate the difficulty of test items as vocabulary item difficulty increases. This pattern becomes more pronounced for items with difficulty values of approximately 0.6 and above. The extent to which these higher difficulty values are impacted needs to be interpreted with caution. First of all, the GLMM difficulty estimates that the model was trained on have their own degree of error; see [Schmitt et al. \(2024\)](#) for further details. In addition, the fact that the GLMM scores were scaled linearly to values between 0 and 1 may also



Figure 4: Example SHAP output for the ES test item for the EN target word "bar" (verb).

impact the distribution of the difficulty values at the low and high end of the scale.

In order to investigate the token-level contributions of the input text to the model predictions, further analysis using SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) was carried out. SHAP is a python package⁹ that assigns Shapley values – a game-theoretic attribution metric – to features of a given predictive model. When applying SHAP to transformer architectures, each token of the input text is treated as an individual feature, affording the investigation of specific words or sub-word units within the sequence. In our application, this allows for fine-grained interpretability of how tokens within different components of the input text contribute to the model’s final prediction.

For each L1 subset of the KVL test data, the top 10% of model errors (68 vocabulary test items per L1) were individually inspected using SHAP. For each item text, the token that contributed the most to the incorrect prediction was recorded, along with which component it was part of. Figure 4 provides an example of the SHAP analysis output for the ES item text for the English target word “bar” (verb). All tokens highlighted in red in the figure contribute to increasing the model’s prediction (towards difficult) and all tokens highlighted in blue contribute to decreasing the model’s prediction (towards easy). For this example, the model predicted the item to be too easy (prediction = 0.52, label = 0.93, error = -0.41). On inspecting the SHAP output, we can see that the EN target word “bar” is the token that contributes the most to the erroneous prediction.

Figure 5 shows the component and prediction direction of the tokens identified in the analysis procedure described above. Reflecting the general tendency of the model predictions reported in Figure 3, there was a higher proportion of ‘too-easy’ predictions (57% of errors investigated). Tokens identified in the EN target word component account for 44% of the items investigated, followed by tokens in the L1 context (25%), L1 word (14%) and EN clue (7%). The separation token <s> was also identified as containing the top contributing token

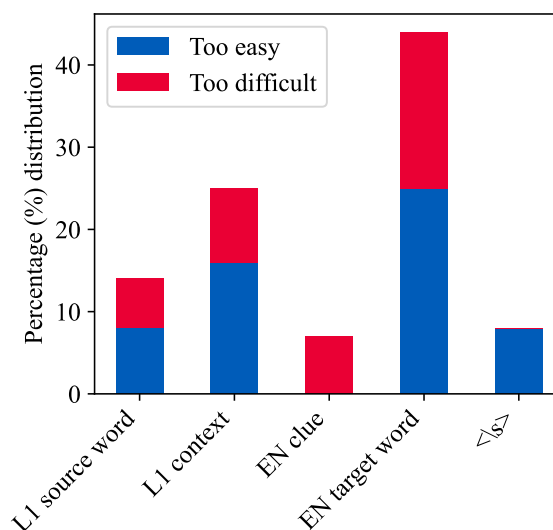


Figure 5: Location of the most influential tokens attributed to the top 10% of model errors.

for 8% of the errors (see Table A.5 in the Appendix for an overview of all identified tokens). On inspecting the tokens, some global patterns emerged.

- **Simple vs. complex words in the L1 word and L1 context.** For items that the model predicted as too easy, simple or common words were often given high attributions. For example, pronouns: “ich” (I), “我” (I), “mich” (me), or every day words: “饭” (meal), “heute” (today), “noche” (night). For items that were predicted too difficult, attribution tended to be given to more complex words such as “precisar” (specify) and “排放” (emission).
- **Sub-word tokenization in EN word.** For items that the model predicted as too easy, the sub-word with the highest attribution was often a simpler or more common word nested within the target word, for example with the compound nouns “bookcase”, “sunshine” and “workday”. For items that were predicted too difficult, words were often split into non-morphologically aligned sub-tokens: “poison”, “fireman”, “killer”, or suffixes: “questionable”, “punishment”.

⁹<https://shap.readthedocs.io/en/latest/index.html>

- **Difficult senses in EN target word.** Whole EN target words accounted for 26% of all items that were predicted too easy, compared to only 8% for items predicted as too difficult. Common features of these EN target words were difficult senses (e.g. “short” as an adverb, “bar” as a verb), cognates with low frequency (e.g. “crystal”, “tragic”), or avoiding cognates in L1 source word (e.g. using “rastrear” instead of “explorar” for EN target word “explore”).

7 Discussion

The experiments and analysis detailed above explored L2 English vocabulary test item difficulty prediction using transformer-based architectures, with a view to establishing: (1) the best model for prediction; (2) the relative importance of test item components; and (3) potential areas of systematic bias in the best model. The outcomes of these aims are discussed below.

Initial results from fine-tuning individual component models highlighted the predictive strength of each component in isolation, with the EN target word input emerging as the most effective standalone predictor. These findings align with prior work in QDET for vocabulary testing, which has shown that features based solely on the English target word can yield a strong performance (Suyong and Hua, 2018; Ehara, 2018; Settles et al., 2020). The English target word model performed especially well on the Chinese subset, achieving a RHO of 0.73, compared to 0.63 for both Spanish and German. This may reflect the inconsistent role of cognateness in shaping word difficulty: while Spanish and German learners may be influenced by cognates or “false friends” (Otwinowska and Szewczyk, 2019), English word difficulty for Chinese learners—whose L1 shares no cognates with English—may be more directly linked to features solely attributed to the English word. As a result, the relationship between the English target word and item difficulty may be easier to model for the Chinese subset of the KVL. A similar pattern was observed by Schmitt et al. (2024) in their analysis of the KVL, where GLMM difficulty scores for the Chinese subset correlated more strongly with word frequency—a feature often used as a proxy for word difficulty (Hashimoto and Egbert, 2019)—than they did for Spanish and German.

Although the monolingual ensemble models showed limited improvement over the simpler En-

glish target word models, there may be some settings where this architecture is a suitable choice. Given the statistically significant improvement seen for the ensemble model fine-tuned with the Chinese subset, it may be that this approach is better suited to non-cognate language pairs where cross-lingual interaction does not play an important role in determining item difficulty. Furthermore, the ensemble model weights can be used as a simple proxy for component importance, offering an efficient, broader view of component relevance that may be more practically applicable for test item piloting. However, for scenarios similar to this study, in a multilingual setting with target words and context, our results suggest that a unified multilingual transformer architecture is the best choice. These findings align with prior research in multilingual LCP, which highlight the benefits of including sentence context (Bani Yaseen et al., 2021; Kelious et al., 2024a) as well as the joint modelling of different L1s (Zaharia et al., 2020).

Findings from the ablation study highlighted the advantage of cross-component representation learning within a unified transformer architecture and revealed interesting insights into the impact of fine-tuning on all L1s. Results showed that in the L1-specific approach, the model fine-tuned for Chinese assigns less importance to the English target word input compared to its Spanish and German counterparts. This is somewhat unexpected, given the very strong performance of the English target word model for Chinese shown in Table 2. One possible explanation is that the L1-specific model fine-tuned on the Chinese subset is less able to align representations of English and Chinese source words due to the lack of script overlap. This is reflected in prior research showing that multilingual models benefit from shared subword representations across languages, and that subword overlap correlates with cross-lingual transfer performance (Wu and Dredze, 2019; Pires et al., 2019). In the case of Chinese, the absence of shared subwords with English may limit the model’s ability to learn cross-lingual connections. This may be a contributing factor as to why there is no significant improvement in the L1-specific model compared to the ensemble approach for the Chinese subset.

Building on this idea, the ablation results for the all-in-one multilingual model were much more consistent across L1 subsets. The observed generalisation suggests that the all-in-one model may be learning broader, language-independent features

of vocabulary item difficulty compared to the L1-specific and ensemble models. The parallel structure of the KVL dataset, where each of the English target word and clue appears across three different L1s, likely supports this generalisation by encouraging the model to disentangle language- and item-specific features from global patterns. Furthermore, the distribution of component impact for the Chinese subset of the L1-specific model reported in Figure 2 shifts considerably toward the Spanish and German distributions seen in the all-in-one model. This may be an indication that the limitations of cross-lingual transfer for orthographically distant language pairs described above are alleviated in this setting when models are fine-tuned jointly across languages with parallel data.

Findings from the error analysis revealed valuable insights about the systematic behaviour of the all-in-one multilingual model. In addition to the effects of label re-scaling and GLMM model error discussed in Section 6.3, the normal distribution of difficulty values in the KVL dataset may further contribute to the all-in-one model’s tendency to under-predict higher difficulty items. To test this, it would be of value to investigate the impact of including a larger proportion of high difficulty items during fine-tuning. This could be achieved using data-augmentation or re-sampling methods (Pan et al., 2021; Kelious et al., 2024b), or even the development of further KVL test items.

The small-scale SHAP analysis on the multilingual model’s top 10% of errors, provided some general observations that can be applied to the future development of knowledge-based vocabulary lists, and test item writing more generally. In particular, the findings illustrated the impact of vocabulary complexity in the L1 word and L1 context components, suggesting that careful consideration of the word choices in the item text is needed when creating such resources for the NLP domain. Issues from the SHAP analysis that emerged relating to model behaviour, such as non-morphologically aligned sub-word tokenization and poor word sense disambiguation provide direction for improving the all-in-one model, such as multi-task learning with POS-tagging, morphological supervision or cross-lingual word sense disambiguation. Finally, given the limited scope of the SHAP-based analysis, interpretations are isolated to the individual word and subword level. Further investigation into the model’s attention across tokens may be able to provide richer insight into the model behaviour.

8 Future Work

In addition to the suggestions outlined in the discussion above, there are several further avenues for future work. First, model probing for features previously found to be predictive of vocabulary item difficulty (Dunn, 2024; Hashimoto and Egbert, 2019) could help explore the item text beyond the component level, to uncover which linguistic correlates of item difficulty are being captured by the models. The all-in-one multilingual model could be further optimised by incorporating architectural adaptations shown to benefit QDET and LCP in other domains, such as scalar mixing (Gombert et al., 2024) or concatenating transformer embeddings with linguistically derived features (AlKhuzayy et al., 2024; North et al., 2023). Given the requirement of large amounts of training data for encoder-based transformer approaches, it would also be of value to compare the all-in-one model results to zero-shot and few-shot methods using LLMs, such as those recently investigated by Smădu et al. (2024). Finally, expanding the KVL dataset to include additional L1 subsets, especially those orthographically distant from English, will contribute to further exploring the role of cross-lingual transfer within multilingual transformer models, helping to corroborate the findings of this research.

9 Conclusion

This research investigated the use of transformer-based architectures for predicting vocabulary item difficulty, applying recent advances in multilingual and cross-lingual lexical complexity prediction to question difficulty estimation. Leveraging the content and structure of the KVL dataset—which has not previously been used in NLP research—this study examined the effects of multilingual text items across several transformer-based architectures. The analysis provided insights into the relative importance of different test item components across L1s, revealing how these models capture and generalise features of item difficulty. In particular, a multilingual model fine-tuned on data with all L1 variations demonstrated the strongest performance, benefiting from cross-lingual transfer and the parallel structure of the KVL dataset to produce more generalised and consistent attributions across L1s. These findings point to the potential of L1-agnostic, explainable transformer-based models for supporting test development pipelines through scalable and interpretable item calibration.

Limitations

One limitation of our study is the use of the probabilistic values derived from the GLMM framework as observed difficulty values, an issue that is discussed in more detail by Schmitt et al. (2024). To address this, we used a non-parametric correlation measure (Spearman's Rho) to evaluate our models based on rank ordering. This approach helps account for the potential error in the precision of estimates that might not be fully captured by RMSE.

Another limitation that is specific to the all-in-one multilingual model lies in the way training data was combined across L1s. The GLMM difficulty values used as labels in the models were derived from different population samples for each L1, which could raise questions about the comparability of these values across languages. To mitigate this, target labels were derived by concatenating the individually scaled subsets rather than applying a single normalisation across the entire KVL dataset. While this approach preserves the internal structure of each L1 subset difficulty scores, it does not fully account for differences in score distribution origins. However, given that predictions improved when the model was evaluated on individual L1 subsets, the all-in-one model can still be viewed as a practical means of enhancing L1-specific performance, rather than as a universal predictor of item difficulty.

Acknowledgments

This research was possible thanks to the work carried out by Norbert Schmitt (University of Nottingham, UK), Karen J. Dunn (British Council, UK), Barry O'Sullivan (British Council, UK), Laurence Anthony (Waseda University, Japan) and Benjamin Kremmel (University of Innsbruck, Austria) who created the original Knowledge-based Vocabulary Lists.

References

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.

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.

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).

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. [Spanish pre-trained BERT model and evaluation data](#). In *PMLADC at ICLR 2020*.

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.

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.

Paul De Boeck. 2008. [Random item IRT models](#). *Psychometrika*, 73(4):533–559.

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.

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.

Bradley Efron. 1987. [Better bootstrap confidence intervals](#). *Journal of the American Statistical Association*, 82(397):171–185.

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)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Pierre Finamore, Elisabeth Fritsch, Daniel King, Alison Sneyd, Aneeq Ur Rehman, Fernando Alva-Manchego, and Andreas Vlachos. 2019. [Strong baselines for complex word identification across multiple](#)

- languages. 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 970–977, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sebastian Gombert, Lukas Menzel, Daniele Di Mitri, and Hendrik Drachslers. 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, Mexico City, Mexico. Association for Computational Linguistics.
- Brett J Hashimoto and Jesse Egbert. 2019. [More than frequency? Exploring predictors of word difficulty for second language learners](#). *Language Learning*, 69(4):839–872.
- Abdelhak Keliou, 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 Keliou, 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.
- Yoolim Kim, Vita V. Kogan, and Cong Zhang. 2024. [Collecting big data through citizen science: Gamification and game-based approaches to data collection in applied linguistics](#). *Applied Linguistics*, 45(1):198–205. Published: 12 July 2023.
- Batia Laufer and Zahava Goldstein. 2004. [Testing vocabulary knowledge: Size, strength, and computer adaptiveness](#). *Language learning*, 54(3):399–436.
- Ekaterina Loginova, Luca Benedetto, Dries Benoit, and Paolo Cremonesi. 2021. [Towards the application of calibrated transformers to the unsupervised estimation of question difficulty from text](#). In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 846–855, Held Online. INCOMA Ltd.
- Scott M Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions](#). In *Advances in Neural Information Processing Systems*, volume 30.
- Kiyooki Nakanishi, Nobuyuki Kobayashi, Hiromitsu Shiina, and Fumio Kitagawa. 2012. [Estimating word difficulty using semantic descriptions in dictionaries and web data](#). In *2012 IIAI International Conference on Advanced Applied Informatics*, pages 324–329.
- Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. [Lexical complexity prediction: An overview](#). *ACM Comput. Surv.*, 55(9).
- 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.
- Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. [DeepBlueAI at SemEval-2021 task 1: Lexical complexity prediction with a deep ensemble approach](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584, Online. Association for Computational Linguistics.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. [How multilingual is multilingual BERT?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. 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.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. [Machine learning–driven language assessment](#). *Transactions of the Association for Computational Linguistics*, 8:247–263.
- Kim Cheng Sheang. 2019. [Multilingual complex word identification: Convolutional neural networks with morphological and linguistic features](#). In *Proceedings of the Student Research Workshop Associated with RANLP 2019*, pages 83–89, Varna, Bulgaria. INCOMA Ltd.
- 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.
- Yuni Susanti, Takenobu Tokunaga, and Hitoshi Nishikawa. 2020. [Integrating automatic question generation with computerised adaptive test](#). *Research and Practice in Technology Enhanced Learning*, 15:1–22.
- Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. 2017. [Controlling item difficulty for automatic vocabulary question generation](#). *Research and practice in technology enhanced learning*, 12:1–16.

- Eum Suyong and Yang Hua. 2018. [Feature analysis on English word difficulty by gaussian mixture model](#). *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 191–194.
- Masaki Uto, Yuto Tomikawa, and Ayaka Suzuki. 2024. [Question difficulty prediction based on virtual test-takers and item response theory](#). In *Workshop on Automatic Evaluation of Learning and Assessment Content (EvalLAC'24)*.
- Shijie Wu and Mark Dredze. 2019. [Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT](#). 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 833–844, Hong Kong, China. Association for Computational Linguistics.
- 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 Clausner. 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.
- George-Eduard Zaharia, Dumitru-Clementin Cercel, and Mihai Dascalu. 2020. [Cross-lingual transfer learning for complex word identification](#). In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 384–390.
- Leonidas Zotos, Hedderik van Rijn, and Malvina Nisim. 2025. [Can model uncertainty function as a proxy for multiple-choice question item difficulty?](#) In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11304–11316, Abu Dhabi, UAE. Association for Computational Linguistics.

A Appendix

Component model	ES	DE	CN
L1 source word	14.97%	39.93%	23.93%
L1 context	18.80%	6.46%	13.94%
EN clue	5.69%	1.66%	0.00%
EN target word	60.54%	51.93%	62.13%

Table A.1: Ridge regression ensemble model weights across L1 subsets.

Input text	Pre-trained model	ES		DE		CN		L1 average	
		RMSE	Corr	RMSE	Corr	RMSE	Corr	RMSE	Corr
L1 context	BERT-mono*	0.139	0.449	0.133	0.437	0.125	0.503	0.132	0.463
L1 context	XLM-R	0.140	0.429	0.134	0.429	0.127	0.483	0.134	0.447
L1 context	mBERT	0.141	0.416	0.135	0.412	0.128	0.451	0.135	0.426
L1 source word	BERT-mono*	0.135	0.496	0.123	0.584	0.118	0.596	0.125	0.559
L1 source word	XLM-R	0.149	0.376	0.132	0.486	0.119	0.555	0.133	0.472
L1 source word	mBERT	0.142	0.409	0.129	0.506	0.119	0.543	0.130	0.486
EN clue	BERT	0.144	0.365	0.139	0.358	0.131	0.396	0.138	0.373
EN clue	RoBERTa	0.145	0.355	0.139	0.343	0.131	0.392	0.138	0.363
EN target word	BERT	0.123	0.592	0.116	0.629	0.096	0.752	0.112	0.658
EN target word	RoBERTa	0.134	0.513	0.133	0.506	0.110	0.624	0.126	0.548
All components	XLM-R	0.103	0.761	0.099	0.777	0.088	0.800	0.097	0.779
All components	mBERT	0.107	0.744	0.103	0.751	0.094	0.773	0.101	0.756

*BERT-mono refers to the monolingual models for each L1 outlined in Section 5.2. Hyperparameters were fixed at $2e-5$ for learning rate, 0.1 for weight decay and 0.1 for warm up ratio.

Table A.2: RMSE and Spearman’s Rho for each of the transformer models considered for the final experiments. Models were fine-tuned on the train set and evaluated on the development set.

Model name	Input language	Input text	Learning rate	Weight decay	Warmup ratio
bert-base-spanish-wwm-cased	ES	L1 source word	3e-5	0	0.1
gbert-base	DE	L1 source word	2e-5	0	0.1
bert-base-chinese	CN	L1 source word	2e-5	0.1	0
bert-base-spanish-wwm-cased	ES	L1 context	3e-5	0	0.1
gbert-base	DE	L1 context	2e-5	0	0.1
bert-base-chinese	CN	L1 context	3e-5	0	0
bert-base-cased	ES	EN clue	2e-5	0.1	0.1
bert-base-cased	DE	EN clue	2e-5	0.1	0
bert-base-cased	CN	EN clue	3e-5	0	0
bert-base-cased	ES	EN target word	3e-5	0	0
bert-base-cased	DE	EN target word	1e-5	0	0.1
bert-base-cased	CN	EN target word	2e-5	0	0
xlm-roberta-base	ES	All components	3e-5	0.1	0.1
xlm-roberta-base	DE	All components	3e-5	0	0.1
xlm-roberta-base	CN	All components	3e-5	0.1	0.1
xlm-roberta-base	XX	All components	3e-5	0.1	0.1

Search space for hyperparameters: learning rate (1e-5, 2e-5, 3e-5), weight decay (0, 0.1), warm up ratio (0, 0.1).

Table A.3: Optuna hyperparameter results for the models selected for the final experimentation. Models were fine-tuned on the train set and evaluated on the development set.

Full component model - removed component	ES		DE		CN		L1 average	
	RMSE	RHO	RMSE	RHO	RMSE	RHO	RMSE	RHO
Ensemble	0.142	0.646	0.129	0.651	0.106	0.747	0.126	0.681
- L1 word	0.142	0.641	0.134*	0.620*	0.108	0.742	0.128	0.668
- L1 context	0.142	0.639	0.128	0.651	0.107	0.742	0.126	0.677
- EN clue	0.142	0.643	0.129	0.650	0.106	0.747	0.126	0.680
- EN word	0.155*	0.578*	0.138*	0.591*	0.122*	0.637*	0.138	0.602
L1-specific	0.126	0.734	0.116	0.776	0.108	0.725	0.117	0.745
- L1 word	0.133*	0.707*	0.122*	0.732*	0.110	0.732	0.122	0.724
- L1 context	0.129	0.711*	0.123*	0.717*	0.116*	0.681*	0.123	0.703
- EN clue	0.125	0.742	0.119	0.755*	0.119*	0.658*	0.121	0.718
- EN word	0.152*	0.545*	0.138*	0.614*	0.117*	0.674*	0.136	0.611
All-in-one	0.116	0.775	0.108	0.793	0.097	0.785	0.107	0.784
- L1 word	0.122*	0.756	0.112	0.770*	0.102*	0.769	0.112	0.765
- L1 context	0.138*	0.640*	0.126*	0.672*	0.114*	0.693*	0.126	0.668
- EN clue	0.121*	0.766	0.115*	0.783	0.101*	0.780	0.112	0.776
- EN word	0.151*	0.570*	0.139*	0.580*	0.123*	0.630*	0.138	0.593

*Statistically significant improvement in performance compared to the 'full component' models (as reported in Table 2). Significance testing was not applied to L1 average results.

Table A.4: RMSE and Spearman's Rho results for the ablation study models. Models were fine-tuned on the train and development set, and evaluated on the test set.

	Too easy			Too difficult		
	ES	DE	CN	ES	DE	CN
L1 source word	paramilitares doble calculadora analíticas restaurada olímpica	falsch Militia Physiker iranisch Orchester	找 账 笑 坐 的 人	precisar embajada discusión	(denkwürdig novice herabfallend Zuhörer	资 属 重 大 (
L1 context	coche Mi domingos noche mezclar pavor físico verduras comida	Imitator heute Unrecht Front ich Beunruhigung wird wirklich mich Handy und Ich	我 旅 行 音 我 骄 傲 去 饭 信 任 从 小 回 家 人 们 我	bomba entero media imperiosa la investigacion profesional efectivo	künstlerische	涌 的 看 法 美 洲 腺 系 统 器 养 郊 排 放
EN clue*	-	-	-	m_____ (masterpiece) n_____ (nominated) n____ (nurse) e_____ (expensive) t_____ (terrorism)	b____ (better) n____ (nurse) t____ (traffic) q____ (quit) p____ (plus) m_____ (masterpiece) r____ (reality)	o_____ (olympic) o_____ (oversize) e_____ (examination)
EN target word	unity bar grand jasmine tragic glow grin sunshine explore pasta dominate introduction lyrics recycled tropics recycling	short bar communicator grand forget crystal tragic cheerful vote kangaroo learned boost dominated recipe ecosystem sitting café bookcase	bar unity grand short bookcase stop tried glow written cheerful gown workday recipe learned birth tragic cite sweat	poison quantity memorable excellent falling venture incomplete questionable fireman chorus incoming	kidnapping faceless climate questionable incoming balancing guilt definite minority established breakout poison taking climbing	chinese relate rely killer governmental inexpensive disorder punishment questionable qualify backward antisocial issue fireman

*The associated EN target word for the EN clue component is included in brackets for interpretability. Within the component groups, words are listed in order of largest to smallest model prediction error for their associated item.

Table A.5: The words and subwords (in bold) contributing to the top 10% of the all-in-one multilingual model errors, according to SHAP analysis.

Automatic concept extraction for learning domain modeling: A weakly supervised approach using contextualized word embeddings

Kordula De Kuthy

Institut für Wissensmedien
Schleichstrasse 6
D-72076 Tübingen

k.dekuthy@iwm-tuebingen.de

Leander Girrbach

Technical University of Munich
Boltzmannstr. 3
D-85748 Garching

leander.girrbach@tum.de

Detmar Meurers

Institut für Wissensmedien
Schleichstrasse 6
D-72076 Tübingen

d.meurers@iwm-tuebingen.de

Abstract

Heterogeneity in student populations poses a challenge in formal education, with adaptive textbooks offering a potential solution by tailoring content based on individual learner models. However, creating domain models for textbooks typically demands significant manual effort. Recent work by Chau et al. (2021) demonstrated automated concept extraction from digital textbooks, but relied on costly domain-specific manual annotations. This paper introduces a novel, scalable method that minimizes manual effort by combining contextualized word embeddings with weakly supervised machine learning. Our approach clusters word embeddings from textbooks and identifies domain-specific concepts using a machine learner trained on concept seeds automatically extracted from Wikipedia. We evaluate this method using 28 economics textbooks, comparing its performance against a tf-idf baseline, a supervised machine learning baseline, the RAKE keyword extraction method, and human domain experts. Results demonstrate that our weakly supervised method effectively balances accuracy with reduced annotation effort, offering a practical solution for automated concept extraction in adaptive learning environments.

1 Introduction

In formal education, the incremental mastery of concepts and the knowledge and competencies that build on them is essential for students to successfully read and understand texts in a specific school subject. Students often struggle to comprehend the relevant concepts within educational materials, leading to difficulties in understanding and applying the knowledge effectively. Many schoolbooks therefore contain a glossary with a list of the key concepts which are manually compiled by the respective schoolbook authors, providing a somewhat subjective list of the relevant concepts.

In the digital counterpart to traditional schoolbooks, computer-based learning platforms, a sim-

ilar challenge arises: modelling the domain for which the platform provides learning materials and exercises. Most digital systems rely on handcrafted ontologies (or related domain representations) that have been designed by domain experts. They compile a list of domain-specific vocabulary, a very resource-intensive, costly, inefficient, and time-consuming process. In the worst case, such domain ontologies have to be newly constructed for every adaptive learning platform from scratch even if other systems already exist in the same domain. To reduce the effort required, Chau et al. (2021) presented an approach automating concept extraction from digital textbooks. While they demonstrate that such extraction can be successfully carried out, the approach still requires an extensive, domain-specific, manual annotation effort of the textbook as a basis for a supervised machine learning approach. Another particular challenge exists for concept extraction in the educational domain: textbooks not only contain specific vocabulary from one subject domain. In particular, school books usually contain content domain specific words, school domain specific words (homework, teacher, exercise, ...) and example specific words.

Keyword extraction (or concept extraction), a fundamental task in natural language processing (NLP) and information retrieval, aims to identify and extract the most important terms or phrases that best represent the content of a document. These extracted keywords can play a crucial role in various applications, such as document summarization, information retrieval, text classification, and topic modeling, but usually not in the educational domain. Nevertheless, the progress in automatic keyphrase extraction has produced methods that are also useful for the related area of automatic concept extraction from textbooks.

This work focuses on the core task of extracting domain-specific vocabulary. It introduces an approach supported by distributional semantics that

uses contextualized word embeddings, moving beyond simple keyword extraction. Recent methods have explored using word embeddings for concept extraction. However, these methods often have low precision or rely on supervised training with large amounts of labeled data and only use static word embeddings. Some very recent approaches to keyword extraction, such as (Qian et al., 2021), use contextualized word embeddings provided by BERT, which shows improved performance. Nevertheless, these approaches primarily focus on keyword extraction in the scientific domain. This means they aim to extract a few specific keywords from documents that mostly cover one particular topic domain, as noted by (Sammet and Krestel, 2023). These methods still use labeled data and treat contextual word embeddings merely as a more advanced embedding type. This work explores whether the improved performance of contextualized word embeddings also applies to the broader task of glossary extraction in the educational domain. This domain presents a unique challenge due to its multi-theme vocabulary. The approach uses contextualized word embeddings, such as BERT, to select domain-specific expressions in educational texts through a clustering method. Supervision is only required in the form of a small seed list of domain-relevant words. This list can be easily compiled from Wikipedia articles and helps separate clusters of words relevant to the specific domain from those that are specific to the text but belong to a different domain.

Our approach for glossary extraction from structured textbooks could support both glossary building for traditional schoolbooks, domain modeling for adaptive learning platforms, and potentially also student modeling in digital learning environments. Our goal is to create a domain-specific glossary extraction method that accurately reflects the concept annotations made by expert users at the section level. This method can then be used to build both domain and student models for more advanced personalization. To evaluate our method, we assess how closely it matches external expert annotations and internal expert annotations (i.e., glossaries compiled by the schoolbook authors).

2 Related Work

There is a broad number of research strands related to keyword extraction. However, there is little work within the educational field. Therefore, we will

focus on approaches with components similar to those in our own method. To present only the main ideas, we will discuss one or two approaches for each method. More detailed overviews are available in (Chau et al., 2021) and (Khan et al., 2022).

Keyword extraction Automatic keyphrase extraction (AKE) has been extensively studied using different approaches, such as rule-based learning, supervised learning, unsupervised learning, or deep neural networks. Since AKE systems are designed to only extract a very small list of relevant keywords, most systems consist of two parts: (1) pre-processing data and extracting a list of candidate keyphrases using lexical patterns and heuristics; and then (2) determining which of these candidates are correct keyphrases. Methods for finding the relevant keyphrases are: statistical methods or frequency-based methods, clustering-based methods, graph-based methods, embedding-based methods, and machine learning methods.

The most basic frequency-based approach is the statistical measure tf-idf (term frequency-inverse document frequency (Jones, 2004)). This method effectively finds relevant terms within a document (high recall) but often includes many irrelevant terms (low precision). Therefore, most approaches combine tf-idf with other measures to narrow down the list of potential keywords.

In graph-based approaches, an entire document is modeled as a graph of semantic relationships between the terms and a ranking approach then selects the terms with the highest number of relationships. Prominent approaches are (i) RAKE (Rose et al., 2010) in which a graph of word co-occurrences is constructed and the top ranked words in this graph are extracted as key words, (ii) TextRank (Mihalcea and Tarau, 2004) in which documents are represented as undirected and unweighted graphs and (iii) PositionRank (Florescu and Caragea, 2017), a fully unsupervised, graph-based model, that simultaneously incorporates the position of words and their frequency in a document to compute a PageRank score for each candidate word. The most recent graph-based approaches employ contextualized word embeddings for calculating the ranking, cf. KPRank (Patel and Caragea, 2021).

In clustering-based approaches, clustering algorithms group candidate phrases into topic clusters and the most representative ones from each cluster are selected as key phrases. Liu et al. (2009) employ cooccurrence-based term relatedness, and

a Wikipedia-based term relatedness for clustering. Grineva et al. (2009) develop a graph-based approach for identifying domain specific terms in multi-theme documents - an unsupervised topic-based clustering method that partitions a graph into thematically cohesive groups of terms.

In supervised statistical learning approaches, all terms in a document must be classified as either positive or negative instances of relevant keyphrases. This classification is based on patterns learned from annotated training sets. For example, Hulth (2003) define manual rules combined with frequency measures to extract all potential keyword expressions from a text. A classifier then determines which of these are actual keyword expressions. Current methods use word embeddings to represent words. For instance, Wang et al. (2014) examine word embeddings to measure the relationships between words in graph-based models. Recent methods also use neural networks (cf. Zhang et al., 2016).

In approaches that view AKE as a sequence labelling task, Alzaidy et al. (2019) predict a sequence of labels where the two labels are keyphrase word or non-keyphrase word. The recent availability of contextualized word embeddings has enabled further improvement in AKE as sequence labelling, as in (Sahrawat et al., 2019) or (Sammet and Kretzel, 2023) where a fine-tuned BERT labels relevant keyphrases in abstracts from economics articles.

Concept extraction In concept or term extraction approaches, the goal is to extract not only a small list of the most general candidates but also extract more specific terms that can be used in applications such as domain ontology construction, text classification, or information extraction. The two possible approaches here are constructing a domain model from scratch or using contrastive corpora to identify domain-relevant terms.

Bordea et al. (2013) propose a domain-independent method for extracting terms. They find general terms in a document, similar to keyphrase extraction, and then use these to build a domain model. Based on this model, they identify other semantically similar terms in the document. The method's performance varies across domains but is more stable than basic term extraction approaches like TermExtractor.

Only a few methods address concept extraction in education. One method, proposed by Chau et al. (2021), uses a supervised feature-based machine learning approach to automatically extract concepts

from digital textbooks. This method trains a supervised learning model to classify whether a term or phrase is a concept. It bases this classification on a detailed set of features. One of the few approaches that explicitly aims at constructing domain-specific glossaries, presented by Park et al. (2002), focuses on building domain-specific glossaries. This is similar to the goal of this article. This method uses a tf-idf-based approach.

Ontology extraction Textbooks and the educational domain play a greater role in the domain of ontology extraction, i.e., building concept hierarchies for textbooks or ontologies from textbooks.

(Wang et al., 2015) present an approach that uses Wikipedia as an external resource to build a concept hierarchy for textbooks. The goal is to extract keyphrases for each chapter of a given book. First, they extract a set of related and important Wikipedia concepts for each book chapter. Second, they use local features to extract related concepts for each chapter separately, utilizing measures such as textual similarity between a book chapter and candidate concepts. The resulting candidate set consists of the top N candidates based on their cosine similarity score and those candidates whose title appears in the chapter title (i.e., `titleMatch` equals 1). These two simple but powerful features can capture most of the related and important concepts for each book chapter.

A similar approach is described in (Conde et al., 2016). This paper introduces LiTeWi, a method that combines term extraction techniques (like linguistic filters and tf-idf) with Wikipedia. It uses Wikipedia as a knowledge base to improve term extraction accuracy by removing terms not related to Wikipedia entries within the specified domain.

Summing up, to the best of our knowledge, current automatic term and concept extraction methods perform unexpectedly poorly and are not tailored for the educational field. Improving automatic extraction of domain-specific concepts would be beneficial for immediate tasks such as student modeling and content recommendation in learning platforms or tutoring systems. Furthermore, it would advance the automatic extraction of domain, i.e. specific glossaries and the construction of ontologies, both of which are crucial for developing learning platforms that currently rely heavily on manual domain models.

For documents with multiple themes, clustering seems to be the most promising approach. This

method has been mainly used for extracting keywords. To encode the domain-specific meaning of concepts that require clustering, contextualized word embeddings seem to be the most promising approach. However, these embeddings have only been used for supervised single-word or sequence labeling of keywords in scientific documents. Our work combines these two methods for domain-specific vocabulary extraction. Our method outperforms other methods and does not require large amounts of labeled data for training and testing.

3 Method

In our approach, called GlossEx, we extract concepts specific to a given domain from text. We do not just extract a small list of keyphrases. Instead, we extract all phrases or words that represent the main concepts of that text. This creates a specialized vocabulary list, which is similar to manually compiling a glossary for a specific text.

3.1 Task formulation and dataset

We are trying to solve the following technical task: Given a document \mathcal{D} that represents a specific domain, our goal is to extract the specialized vocabulary \mathcal{V} of that domain from \mathcal{D} . We are exploring this task within the domain of teaching economics in schools. For our dataset, we selected 28 economics textbooks used for the economic curriculum in German secondary schools. We expect our method to identify domain-specific concepts such as “workforce”, “consumption”, “entrepreneur”, and similar terms. In order to extract the domain-specific vocabulary, we propose the following pipeline:

1. Document preprocessing, i.e. tokenization, lemmatization, POS-tagging, ...
2. Extract salient vocabulary \mathcal{S} contained in \mathcal{D}
3. Cluster vocabulary items in \mathcal{S} based on their contextualised embeddings
4. Obtain \mathcal{V} by filtering \mathcal{S} using limited domain knowledge

This pipeline is based on the following observations: Because \mathcal{D} represents a specific domain, it features specialized vocabulary. Conversely, this specialized vocabulary is particularly prominent in \mathcal{D} compared to general, non-domain-specific documents. The second step of the pipeline uses this

observation. However, economic textbooks contain three distinct types of salient vocabulary in addition to the general vocabulary found in any text: (i) specialized vocabulary (which is the extraction target), (ii) education-specific vocabulary (such as instructions like “write” or “analyze”), and (iii) example vocabulary (which appears prominently due to its presence in running or repeated examples).

Therefore, we need to exclude education specific vocabulary and example vocabulary from \mathcal{S} in order to obtain \mathcal{V} . This is done through Items 3 to 4. The clustering step in Item 3 serves to stabilise the filtering method in Item 4: We observe that contextual embeddings form useful clusters, so that specialized and non-specialized vocabulary form local clusters in embedding space. Therefore, we exploit this property to include or exclude complete clusters in \mathcal{V} instead of single lemmas. Item 4 accesses limited domain knowledge to differentiate between the 3 salient categories described above. We use the limited domain knowledge to label each cluster with one of the three categories listed above, and eventually only return lemmas in clusters labeled as specialised vocabulary.

Next, we describe in detail how to implement each step of the proposed pipeline. The focus is on German economics textbooks, but the general method applies to various domains and languages, provided the necessary models are available. We also present the specific German processing tools.

3.2 Preprocessing

The NLTK library (Bird et al., 2009) is used for splitting sentences and tokenizing text. The Hanover Tagger (Wartena, 2019), which is specifically designed for German, is used for sentence-level lemmatization and POS tagging. All subsequent steps are applied to the lemmatized document \mathcal{D} , unless stated otherwise.

3.3 Extracting Salient Vocabulary

We extract the salient vocabulary \mathcal{S} from \mathcal{D} using the method proposed by Lemay et al. (2005). This method calculates scores for all lemmas. These scores show whether a lemma appears more often in \mathcal{D} than is typical for the language in general. Thus, this method distinguishes the salient vocabulary of \mathcal{D} from general vocabulary. For evaluation, we use two general German word frequency lists:

1. A frequency list¹ derived from the DeReKo

¹DeReKo-2014-II-MainArchive-STT.100000 obtained

(Lüngen, 2017). DeReKo is a very large corpus that is representative of contemporary German.

2. The SUBTLEX-DE frequency list (Brysbaert et al., 2011), which has been shown to better explain cognitive saliency of words in decision time experiments.

We only consider nouns and verbs, and we discard stopwords and lemmas that appear less than four times in \mathcal{D} , as well as tokens that contain special characters. Note, that the method described in (Lemay et al., 2005) differs from tf-idf. Specifically, tf-idf calculates frequencies only within a single corpus, whereas our method compares frequencies between two corpora.

3.4 Clustering Vocabulary

We cluster lemmas in \mathcal{S} by agglomerative clustering of contextualised embeddings. To compute embeddings, we use the bert-base-german-cased BERT model provided by Chan et al. (2020). We embed each (non-lemmatized) sentence individually (after subword tokenization). Then, embeddings of subword tokens are mean-pooled to derive embeddings of the original tokens. Finally, lemma embeddings are the mean of all token embeddings associated with the respective lemma.

Agglomerative clustering is computed by the respective scikit-learn implementation (Pedregosa et al., 2011) using default parameters. In preliminary experiments, we found agglomerative clustering to perform better for our task than k-means clustering or spectral clustering methods. We set the number of clusters (which is a required parameter of agglomerative clustering) to $\frac{|\mathcal{S}|}{4}$. This means the expected number of words in a cluster is 4.

This approach differs from graph-based algorithms, such as the one proposed by Grineva et al. (2009). We do not use graph topology to find clusters. Instead, we directly cluster lemmas in the embedding space. In the graph paradigm, this means we are working with a fully connected graph where edge weights are determined by a distance metric in the embedding space.

3.5 Filtering by Domain Knowledge

In the last step, we select clusters that contain specialized vocabulary from a specific domain. How-

ever, obtaining this information directly from embeddings is difficult. Therefore, we create two lists: \mathcal{V}_{edu} and \mathcal{V}_{eco} . \mathcal{V}_{edu} contains seed words related to the education domain, and \mathcal{V}_{eco} contains seed words related to the economics domain. These lists inject a limited amount of domain knowledge into our method, which helps us determine if a cluster contains terms associated with the education domain, the economics domain, or neither.

Application of seed lists Each cluster \mathcal{C} (representing a set of lemmas in \mathcal{D}) receives two scores: an association score for educational vocabulary (σ_{edu}) and an association score for economics vocabulary (σ_{eco}). The scores for a cluster are calculated by taking the average of the 10 smallest pairwise distances between any word in that cluster and any word in either the educational vocabulary (\mathcal{V}_{edu}) or the economics vocabulary (\mathcal{V}_{eco}). The distances between words are measured using the Euclidean distances of fastText embeddings² (Grave et al., 2018). It is important to note that this method uses static word embeddings, which differs from the approach in Section 3.4 where contextualized embeddings from a German BERT model (Chan et al., 2020) are used. fastText embeddings are chosen because their model can create embeddings for any string based on its character n-grams. This avoids the problem of out-of-vocabulary words. Specifically, clusters are kept if they meet one of the following conditions:

$$\sigma_{\text{eco}} + 0.03 < \sigma_{\text{edu}} \quad (1)$$

$$\sigma_{\text{eco}} < \min\{0.3, \sigma_{\text{edu}}\} \quad (2)$$

In simpler terms, this means that clusters are selected if they are generally close to the economics vocabulary (\mathcal{V}_{eco}) or if they are significantly closer to \mathcal{V}_{eco} than to the educational vocabulary (\mathcal{V}_{edu}). These thresholds are specific to the embedding space used and are set manually. The thresholds were determined before any labeled data was available, so their manual setting does not affect the validity of the results. With a small amount of labeled data, it would be possible to automatically adjust these thresholds.

Construction of seed lists The lists are created independently from the evaluation data to avoid circularity. With the PetScan interface we extracted Wikipedia article titles and wikidata entity names

from <https://www.ids-mannheim.de/digspra/kl/projekte/methoden/derewo/>

²obtained from <https://fasttext.cc/docs/en/crawl-vectors.html>

with the following hyperparameters: To populate \mathcal{V}_{edu} , we run one query on the “Bildung” (engl.: *education*) category with maximum depth 6 and require the found pages to link to the Wikipedia page “Schule” (engl.: *school*). To populate \mathcal{V}_{eco} , we run two queries on the “Wirtschaftswissenschaft” (engl.: *economics*) category with maximum depth 6. For the first query, we require found pages to link to the Wikipedia page “Markt” (engl.: *market*). For the second query, we require found pages to link to the Wikipedia page “Bedarf” or to the Wikipedia page “Bedürfnis” (both engl.: *need*). We combine the results of both queries.

To create the final seed lists, which contain only single lemmas, the preprocessing method described in Section 3.2 is applied to every page title returned by PetScan. The resulting lemmas are then saved. Consequently, \mathcal{V}_{edu} contains 562 unique lemmas, and \mathcal{V}_{eco} contains 677 unique lemmas. Although these lists may seem large, they contain a significant amount of noise. Additionally, as shown in Section 4.3, extracting specialized vocabulary using only the words in the seed lists, without our GlossEx method, leads to poor performance.

4 Results and Evaluation

The main evaluation metrics are precision and recall. The goal is to assess how much of the specialized vocabulary the proposed method finds and how many lemmas it returns are actually specialized vocabulary.

4.1 Data

To evaluate the GlossEx method, we use 28 partially digitized German economics textbooks, which cover various school types and years. Nineteen of these textbooks also have paired OCR-scanned glossaries. For all lemmas that appear at least four times in the corpus, we collect expert judgments whether each lemma is domain-specific vocabulary in the field of economics. One expert (not an author of this paper) labeled all 3,458 unique lemmas with binary labels. Out of these, 469 lemmas (13.56%) are labeled as domain-specific vocabulary. To assess inter-annotator agreement, another expert (also not an author of this paper) independently labeled a subset of 510 lemmas. Cohen’s κ (Cohen, 1960) between both annotators is 0.66, which is considered substantial agreement according to Landis and Koch (1977). Furthermore, the f1-score between both annotators

is 0.79, which sets an upper bound on the models’ performance. However, this bound is not specific to any particular textbook.

4.2 Baselines

We compare our method GlossEx described in Section 3 to several baselines. The first set of baselines comprises methods that are widely used for keyword extraction, namely tf-idf (Jones, 2004), Rapid Automatic Keyword Extraction (RAKE) (Rose et al., 2010), and supervised learning on static word embeddings. Note, that the supervised baseline requires labels and therefore uses strictly more information than is available to our method. Thus, the supervised baseline serves as an upper bound to see how much worse methods that do not require explicit labels perform.

A second set of baselines evaluates how well we can extract keywords using two methods: either by simply using the seed lists from Section 3.5 or by using the glossaries included in textbooks. By comparing two seed lists as a baseline, we ensure that our algorithm can discover domain-specific vocabulary beyond the initial input. By comparing two glossaries as a baseline, we confirm the relevance of our problem. This comparison shows that textbook glossaries do not contain all the vocabulary that experts consider domain-specific.

tf-idf measures how relevant a term is to a specific document within a collection of documents (corpus). A term is more relevant if it appears often in the document (high term frequency) but less relevant if it appears in many other documents (high inverse document frequency). To apply tf-idf in our context, given a textbook \mathcal{D} : Term frequency is the frequency f_t of term t in \mathcal{D} . Inverse document frequency is the logarithm of the inverse ratio of sections in \mathcal{D} that also contain t . The formula for tf-idf is:

$$\text{tf-idf}(t, \mathcal{D}) = \frac{f_t}{\sum_{t'} f_{t'}} \cdot \log \left(\frac{N}{g_t} \right) \quad (3)$$

Here, N is the total number of sections in \mathcal{D} , and g_t is the number of sections in \mathcal{D} that contain the term t . Typically, a threshold $\tau \in \mathbb{R}$ is used to identify domain-specific vocabulary. Any term with a tf-idf score greater than τ is considered part of this vocabulary. In our specific case, we choose the τ value that maximizes the f1-score of the predicted domain-specific vocabulary.

Rapid Automatic Keyword Extraction RAKE (Rose et al., 2010) selects keyphrases from documents for information retrieval via assigning each keyphrase a score based on cooccurrence statistics and returning the 33% top scoring keyphrases. We use the implementation provided by the rake-nltk library.³ We only consider single keywords, i.e. the maximum keyphrase length is 1. As stopwords, we provide the list of German stopwords provided by the NLTK (Bird et al., 2009).

Supervised Learning The task of domain-specific vocabulary extraction can be described as a binary classification problem if we are given lemmas and binary labels that show if each lemma is specific to a certain domain. We represent lemmas using their static fastText embeddings, as described in Section 3.5. Then, we train a multi-layer perceptron (MLP) to predict the correct label from these embeddings. To get predictions for all lemmas in a textbook, we use 5-fold stratified cross-validation. We use the scikit-learn library for both cross-validation and the MLP.

Glossaries Nineteen textbooks in our dataset include a glossary. We assess how much of the domain-specific vocabulary these glossaries cover and whether they also contain general, non-domain-specific vocabulary. We extract all elements from these 19 glossaries. We then keep all individual tokens, excluding stopwords, and lemmatize them. From these, we only keep nouns and verbs. We then return only the remaining lemmas from the glossary that appear in the given textbook \mathcal{D} .

Seed Lists In this case, we return all entries from the seed lists described in Section 3.5 that also appear as domain-specific vocabulary in the textbook. This baseline tests whether GlossEx can discover new domain-specific vocabulary and successfully discard non-domain-specific vocabulary. However, the seed lists directly determine which lemmas are returned as domain-specific and which are discarded after the clustering step (see Section 3.4). Therefore, there is a close relationship between the precision of the seed lists (i.e., how many of the seed list entries are actually domain-specific vocabulary) and the precision of GlossEx.

4.3 Results

Overall Performance In Table 1, we present the precision, recall, and f1-score for all meth-

Method	Precis.	Recall	F1
tf-idf	0.152	<u>0.685</u>	0.230
RAKE	0.172	0.854	0.283
Glossary	0.821	0.258	0.382
Wiki-Seedlist	0.367	0.065	0.103
GlossEx-dereko (ours)	<u>0.543</u>	0.584	<u>0.545</u>
GlossEx-subtlex (ours)	0.518	0.645	0.559
Supervised	0.754	0.524	0.589

Table 1: Precision, recall, and f1-scores of GlossEx and baselines. Scores are averages across the 28 textbooks in our dataset. Best results (excluding supervised) are in bold, and second best results are underlined. “dereko” and “subtlex” refer to the background corpus.

ods. These scores are macro-averaged across all 28 textbooks in the dataset. The supervised baseline shows the best overall performance, as expected. Because the data is imbalanced (with only a few domain-specific words), the precision for this baseline is higher than its recall. Conversely, tf-idf and RAKE perform poorly in terms of f1-score. These methods identify many words as domain-specific vocabulary, leading to high recall but low precision. RAKE performs better than tf-idf, even though the optimal score threshold is used for tf-idf.

Using glossaries improves performance. However, these results cannot be directly compared because glossaries are only available for 19 textbooks. Therefore, the reported results are averaged only over these 19 textbooks. Generally, glossaries mainly contain domain-specific vocabulary. However, they miss 75% of the domain-specific vocabulary in textbooks, which is indicated by the low recall. The seed list extracted from Wikipedia yields low precision, low recall, and consequently, a very low f1-score. Still, the precision is higher than that of tf-idf and RAKE. This is expected because the construction method directly uses the Wikipedia category hierarchy. This primarily confirms that our method’s performance is not simply due to a very strong starting point through the seed lists.

Finally, GlossEx achieves an improvement over the seed lists in terms of f1-score and recall. This shows that GlossEx can indeed leverage distributional semantics to identify domain-specific vocabulary. Our method also significantly reduces the gap between baseline methods and supervised learning. Compared to supervised learning, our method achieves higher recall at the expense of

³<https://pypi.org/project/rake-nltk/>

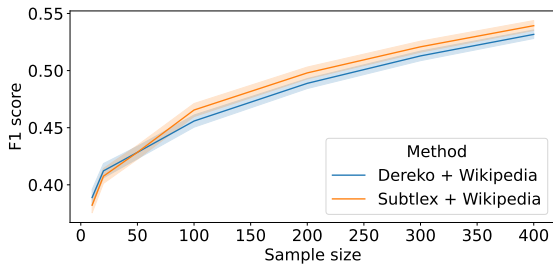


Figure 1: GlossEx f1-score vs. increasing seed list size.

lower precision. Therefore, exploring more precise methods to characterize the contextual semantics of words in textbooks seems to be a promising direction for improving our method.

Since GlossEx relies on external data, we must characterize the influence of the background corpus and seed list on its performance. As seen in Table 1, using SUBTLEX-DE instead of DeReKo (referred to as “subtlex” and “dereko”) as the background corpus results in higher recall but lower precision. One possible reason for this is that DeReKo has broader vocabulary coverage, which leads to fewer lemmas appearing prominent for a given document. Additionally, the DeReKo corpus contains many newspaper texts, which might bias frequency estimates for terms related to topics like politics and financial news.

Effect of Seed List Size To assess how the size of the seed list affects our method’s performance, we repeatedly select random seed lists of different sizes. These sample sizes, denoted as n , are chosen from the set $\{10, 20, 50, 100, 200, 300, 400\}$. From the complete set of all entries in the seed lists, for each sample size n , we sample $k = 100$ seed lists. In all instances, both the economics and education seed lists have the same number of entries. We then re-evaluate our method using these selected seed lists.

Figure 1 shows that the performance of GlossEx consistently improves as the seed list size increases. This outcome is expected and these findings also indicate that GlossEx is resilient to direct overlaps between seed lists and textbook vocabulary. The Wikipedia seed list, for example, contains only a few domain-specific terms. However, GlossEx can fully utilize the semantic information found in these entries. In summary, our results demonstrate that GlossEx performs well with 100 to 200 noisy seed words. However, it achieves optimal performance when provided with more, higher-quality entries.

5 Discussion and Future Work

Our method, GlossEx, uses traditional machine learning and natural language processing (NLP) techniques for domain vocabulary extraction, such as clustering and word embeddings. Unlike previous methods, we also include contextualized embeddings derived from large language models (LLMs). Recent versions of generative LLMs have been very successful in various zero-shot applications (Brown et al., 2020; Achiam et al., 2023). These advancements are promising for all areas of NLP, including education (Alhafni et al., 2024; Wen et al., 2024), making the use of LLMs for domain-specific vocabulary extraction in a zero- or few-shot manner an exciting direction for future research. However, we believe that combining LLMs with modular approaches like ours is most effective, because we can not only identify, but also explain *why* certain words are considered domain-specific. This explanation comes from traceable differences in word occurrences in domain-specific versus general texts, and from semantic similarity to known domain-specific words. This built-in interpretability makes GlossEx a valuable approach even in the era of LLMs.

6 Conclusion

Given educational materials, how can we systematically extract the domain concepts to be learned and understood by students? Answering this is relevant for building glossaries for textbooks, for domain and student modeling for adaptive learning platforms, and for the automatic derivation of activity models for text-based learning materials. In this paper, we investigated how computational linguistic methods such as distributional semantic analysis and clustering can be combined to automatically extract a domain-specific glossary. We presented a pipeline to extract specialized vocabulary from single documents, e.g., textbooks. The pipeline is optimized for documents from the educational domain, where pedagogical terminology cannot easily be separated from subject domain concepts by statistical methods alone. Pursuing a weakly supervised approach, we injected only a limited amount of domain knowledge in the form of a seed list readily obtained from Wikipedia. We evaluated the method on German economics textbooks. Evaluation is both automatic, by comparing the extracted vocabulary to paired glossaries, and manual by human domain experts.

Data and Code Availability

Our implementation of GlossEx is available at <https://github.com/LGirrbach/GlossEx>. We cannot release the textbook material used in this paper because it is copyrighted. However, the textbook titles are included in our code release.

Limitations

While our approach to automatic concept extraction using contextualized word embeddings and weakly supervised learning shows promising results, there are some limitations to our approach.

First, the reliance on pre-trained language models such as BERT, which are primarily trained on general corpora, may not fully capture the nuances of domain-specific language used in educational texts. This can lead to less optimal performance in identifying and clustering domain-specific vocabulary, particularly in specialized fields not well-represented in the training data.

Second, the quality and comprehensiveness of the seed lists used to guide the clustering process significantly influence the results. Although we used Wikipedia to generate these lists, the potential gaps in coverage can affect the accuracy of the extracted concepts. In future work, one could explore more refined methods for seed list generation or incorporate additional domain-specific resources to support the robustness of the approach.

Third, the performance of GlossEx is evaluated on a relatively small and specific dataset of German economics textbooks. This limits the generalizability of our findings to other subjects and educational contexts. Extensive testing on diverse datasets is necessary to validate the broader applicability of our approach.

Finally, while our approach reduces the need for extensive manual annotation, it still requires some level of domain knowledge for seed list creation and cluster validation. This semi-supervised nature means that the method is not entirely free from human intervention, which could be a limitation in fully automating the concept extraction process.

Addressing these limitations in future research will be crucial for enhancing the scalability, accuracy, and applicability of our method in various educational settings.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. In *arXiv*.
- Bashar Alhafni, Sowmya Vajjala, Stefano Bannò, Kaushal Kumar Maurya, and Ekaterina Kochmar. 2024. Lims in education: Novel perspectives, challenges, and opportunities. In *arXiv*.
- Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi- lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference*.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with python: analyzing text with the natural language toolkit.
- Georgeta Bordea, Paul Buitelaar, and Tamara Polajnar. 2013. Domain-independent term extraction through domain modelling. In *International Conference on Terminology and Artificial Intelligence*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *NeurIPS*.
- Marc Brysbaert, Matthias Buchmeier, Markus Conrad, Arthur M Jacobs, Jens Bölte, and Andrea Böhl. 2011. The word frequency effect. In *Experimental psychology*.
- Branden Chan, Stefan Schweter, and Timo Möller. 2020. German’s next language model. In *COLING*.
- Hung Chau, Igor Labutov, Khushboo Thaker, Daqing He, and Peter Brusilovsky. 2021. Automatic concept extraction for domain and student modeling in adaptive textbooks. In *International Journal of Artificial Intelligence in Education*.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. In *Educational and psychological measurement*.
- Angel Conde, Mikel Larrañaga, Ana Arruarte, Jon A Elorriaga, and Dan Roth. 2016. litewi: A combined term extraction and entity linking method for eliciting educational ontologies from textbooks. In *Journal of the Association for Information Science and Technology*.
- Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *ACL*.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *LREC*.

- Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multi-theme documents. In *18th international conference on World wide web*.
- Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *EMNLP*.
- Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. In *Journal of documentation*.
- Muhammad Qasim Khan, Abdul Shahid, M Irfan Uddin, Muhammad Roman, Abdullah Alharbi, Wael Alosaimi, Jameel Almalki, and Saeed M Alshahrani. 2022. Impact analysis of keyword extraction using contextual word embedding. In *PeerJ Computer Science*.
- J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. In *biometrics*.
- Chantal Lemay, Marie-Claude L’Homme, and Patrick Drouin. 2005. Two methods for extracting “specific” single-word terms from specialized corpora: Experimentation and evaluation. In *International Journal of Corpus Linguistics*.
- Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *EMNLP*.
- Harald Lungen. 2017. Dereko—das deutsche referenzkorpus. In *Zeitschrift für germanistische Linguistik*.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*.
- Youngja Park, Roy J Byrd, and Branimir Boguraev. 2002. Automatic glossary extraction: Beyond terminology identification. In *COLING*.
- Krutarth Patel and Cornelia Caragea. 2021. Exploiting position and contextual word embeddings for keyphrase extraction from scientific papers. In *EACL*.
- 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. In *JMLR*.
- Yili Qian, Chaochao Jia, and Yimei Liu. 2021. Bert-based text keyword extraction. In *Journal of Physics: Conference Series*.
- Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. In *Text mining: applications and theory*.
- Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent, Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2019. Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings. In *arXiv*.
- Jill Sammet and Ralf Krestel. 2023. Domain-specific keyword extraction using bert. In *Conference on Language, Data and Knowledge*.
- Rui Wang, Wei Liu, and Chris McDonald. 2014. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Software engineering research conference*.
- Shuting Wang, Chen Liang, Zhaohui Wu, Kyle Williams, Bart Pursel, Benjamin Brautigam, Sheryn Saul, Hannah Williams, Kyle Bowen, and C Lee Giles. 2015. Concept hierarchy extraction from textbooks. In *ACM Symposium on Document Engineering*.
- Christian Wartena. 2019. A probabilistic morphology model for german lemmatization. In *KONVENS*.
- Qingsong Wen, Jing Liang, Carles Sierra, Rose Luckin, Richard Tong, Zitao Liu, Peng Cui, and Jiliang Tang. 2024. Ai for education (ai4edu): Advancing personalized education with llm and adaptive learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- Qi Zhang, Yang Wang, Yeyun Gong, and Xuan-Jing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *EMNLP*.

Supplementary Material

A Qualitative Examples

This section presents the predictions made by GlossEx (using SUBTLEX-DE as its background corpus) on a specific textbook. The predicted lemmas are divided into two categories: correctly predicted (true positives) and incorrectly predicted (false positives). The results for Westermann: Kompetenz Politik-Wirtschaft 2006 (Gymnasium Niedersachsen, Stufe 8) are as follows:

Correct Lemmas	Incorrect Lemmas
Einkommen, Haushalt, Ökonom, Geld, Markt, Wirtschaft, Händler, Anbieter, Knappheit, Angebot, Bedürfnis, Käufer, Nachfrage	Herr, Ergebnis, Person, Wunsch, Cent, Mark, Laden, Mensch, Mitglied, Form, Preis, Verfügung, Mittel, Stand, Kauf, Wochenmarkt, Euro, Prinzip, kaufen, Taschengeld

Our method successfully identifies words with domain-specific meaning, such as “Haushalt” (English: *budget*) and “Nachfrage” (English: *demand*). However, GlossEx also identifies common economic terms that are part of everyday language, like “Laden” (English: *shop*) and “Euro”. Additionally, GlossEx finds words such as “Person” (English: *person*) and “Mensch” (English: *human*). These terms have a strong semantic similarity to other human-related words, such as “Käufer” (English: *buyer*), and are therefore included in the list of predicted lemmas. An examination of results from other textbooks generally supports these findings.

Towards a Real-time Swedish Speech Analyzer for Language Learning Games: A Hybrid AI Approach to Language Assessment

Tianyi Geng

Department of Philosophy,
Linguistics, Theory of Science
University of Gothenburg
gusgenti@student.gu.se

David Alfter

Gothenburg Research Infrastructure
in Digital Humanities
Department of Literature,
History of Ideas, Religion
University of Gothenburg
david.alfter@gu.se

Abstract

This paper presents an automatic speech assessment system designed for Swedish language learners. We introduce a novel hybrid approach that integrates Microsoft Azure speech services with open-source Large Language Models (LLMs). Our system is implemented as a web-based application that provides real-time quick assessment with a game-like experience. Through testing against COREFL English corpus data and Swedish L2 speech data, our system demonstrates effectiveness in distinguishing different language proficiencies, closely aligning with CEFR levels. This ongoing work addresses the gap in current low-resource language assessment technologies with a pilot system developed for automated speech analysis.

1 Introduction

In recent years, the integration of state-of-the-art artificial intelligence (AI) technologies—particularly large language models (LLMs)—has shown considerable promise across a range of domains, including Intelligent Computer-Assisted Language Learning (ICALL), Technology-Enhanced Language Learning (TELL), and Second Language Acquisition (SLA) (Zhang and Zou, 2022; Huang et al., 2023). A growing body of research has demonstrated the effectiveness of AI-driven language assessment tools (Daniels, 2022; Huawei and Aryadoust, 2023; Settles et al., 2020), highlighting their potential to facilitate language learning within contextually rich environments (Zou et al., 2023; Dizon, 2020; Huang et al., 2023). For instance, Brena et al. (2021) proposed supervised machine learning approaches capable of evaluating L2 English fluency and pronunciation with reported accuracy rates exceeding 90%. Despite these advancements, a recent systematic review of AI-based assessment in language learning (Chen et al., 2024) indicates a marked imbalance: 88% of the reviewed tools were developed for English learning, and only

3 out of 25 studies focused on assessing learners' speaking skills. This disparity underscores a significant gap in the current research landscape.

This paper aims to address the gap in automatic speech assessment tools, specifically for non-English languages by proposing a hybrid AI approach. We examine the adaptability of a pronunciation assessment tool optimized for English (Azure Speech Services; Microsoft 2024) to the low-resource Swedish language, then extend it by integrating large language models for content and delivery assessment, forming a detailed assessment system. In addition, the system is built as a Web App, providing real-time feedback as well as a game-like user experience. In the following sections, we will first justify the importance of building an automatic Swedish speech assessment system by reviewing recent related studies and applications around low-resource language speech assessment. We will then introduce our system design, followed by the evaluation and validation of the system with the English speech data from the COREFL corpus (Lozano et al., 2020) and an initial collection of Swedish L2 samples. Finally, we will discuss the results of the system tested for Swedish speech assessment and address the conclusions.

2 Related Work

2.1 Automatic Speech Assessment Systems

Using mobile-assisted language learning (MALL) applications like Duolingo and Babbel has been a popular option for learners (Lehman et al., 2020; Loewen et al., 2020), especially for those studying low-resource languages for which accessible learning resources are scarce. Although MALL apps offer beginners a quick start, there is a lack of efficient or systematic follow-ups. Those apps mostly give a binary score (“correct or not”), or star-based assessment restricted to pronunciation practices, providing neither a comprehensive overview

of speaking ability nor detailed feedback such as pronunciation suggestions (Lehman et al., 2020; Chang et al., 2022).

For more detailed pronunciation assessment, Microsoft Azure Speech Studio (Microsoft, 2024) offers metrics related to accuracy, fluency, completeness, and prosody, as illustrated in Figure 1. While the service provides a multifaceted analysis at the phoneme, word, and sentence levels, the resulting scores remain relatively abstract and are not accompanied by pedagogically oriented feedback or actionable guidance for instructional use. In our experiments, the open-source Azure SDK was found to be primarily optimized for English language assessment, exhibiting limited capacity to accurately process Swedish phonemes. Notably, the system was unable to generate prosody scores for Swedish speech. Despite these limitations, the platform represents a promising prototype for pronunciation assessment and has the potential to be developed into a more robust tool for evaluating spoken language performance, particularly in the context of low-resource languages.

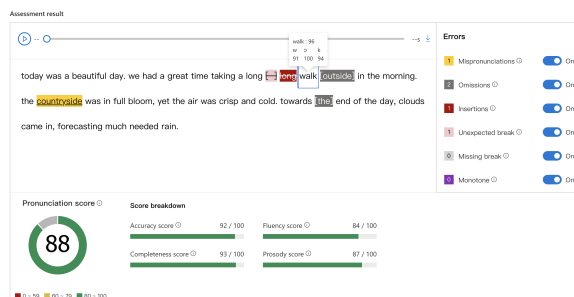


Figure 1: Assessment interface of Azure Speech Studio

2.2 Swedish Learner Data

While the rapid advancement of artificial intelligence models has provided language education with handy tools for quick evaluations (Daniels, 2022; Löber et al., 2024), there is a lack of reliable and detailed automatic systems targeting lower-resource languages such as Swedish. Recent research has been working on filling the blank of Swedish learner data sets through building corpora of which language and proficiency levels are collected from coursebooks (COCTAILL corpus, representing learners’ receptive ability) and learner essays (SweLL-pilot, representing learners’ productive ability) (Volodina et al., 2019).

Nevertheless, the current progress has been made centering mostly texts rather than speech. The ab-

sence of a variety of publicly accessible, annotated Swedish speech data remains a significant obstacle for training robust (deep) learning models. While resources like Common Voice (Mozilla Foundation, 2020) provide raw speech data from native speakers, there is a scarcity of language learner speech samples on the spectrum of proficiency levels needed for developing language assessment or learning applications.

Getman et al. (2023) introduced an AI-assisted language learning application aimed at supporting children’s second language acquisition in low-resource languages, specifically Swedish and Finnish, through the self-collection of relevant datasets. They also highlighted a significant gap in the field, noting: “To the best of our knowledge, in the context of Computer-Assisted Pronunciation Training (CAPT) for L2 Swedish and Finnish children, there are no previous work on automatic pronunciation assessment, not even for L2 Swedish and L2 Finnish adults” (Getman et al., 2023, p. 86026). In response to this gap, the present study contributes to the underexplored area of automatic speech assessment for L2 Swedish by developing a dedicated assessment system and conducting initial evaluations based on authentic speech data produced by L2 learners.

2.3 Language Proficiency Assessment Standards

The Common European Framework of Reference for Languages (CEFR; Council of Europe 2001) has been a widely recognized standard for assessing language proficiency, and recent research (Chen et al., 2024; Volodina et al., 2024) continues to use the CEFR standards and descriptors as reference-framework. While the Common European Framework of Reference for Languages (CEFR) remains a widely recognized standard, its limitations have been noted. As Alderson (2007, p. 660) observed, “the methodologies being used [to compile these descriptions] are unclear or suspect.” The CEFR’s abstract classification into six proficiency levels (A1 to C2) relies heavily on human evaluators—such as language instructors and linguists—which introduces concerns regarding subjectivity and scalability. Furthermore, although learners may be broadly categorized according to CEFR levels, the framework offers limited granular guidance tailored to specific proficiency levels or individual languages. This highlights a disconnect between the standardized assessment framework and the practical de-

mands of language learning and instruction (Settles et al., 2020).

Our proposed automated speech system generates detailed analysis including:

- **Overall performance** Scores in pronunciation, content, and delivery of the speech; the corresponding CEFR level
- **Word-level pronunciation performance** demonstrating specific pronunciation strengths and weaknesses
- **Real-time feedback** with next-step learning suggestions

By combining the traditional assessment metrics and detailed, heuristic assessment analysis, we aim to build a system that generates more readable, informative results, to better serve both learners and educators.

3 System Design

Building on the automated speaking assessment framework developed by Educational Testing Service (ETS) and outlined by Zechner and Evanini (2019), the primary innovation of our system lies in the integration of complementary technologies to evaluate distinct dimensions of speech performance. The system is structured around three core modules: Pronunciation Assessment (based on two read-aloud tasks), Content and Delivery Assessment (based on a free-speech task), and CEFR Level Classification. The implementation takes the form of a web-based application featuring a gamified interface designed to enhance user engagement and learning experience.

3.1 Pronunciation Assessment Module

The system incorporates the pronunciation assessment module provided by Microsoft Azure’s Speech SDK (Microsoft, 2024), which generates evaluation scores across five dimensions: Accuracy, Completeness, Fluency, Confidence, and Word-level confidence scores. Although the module does not support prosodic analysis for Swedish, our integration extends its applicability to the Swedish language and compensates for this limitation by supplementing it with two additional assessment modules.

3.2 Content-and-Delivery Assessment Module

The system utilizes a generative large language model (Llama 3.1; Touvron et al. 2023) to assess aspects of speech beyond pronunciation, specifically

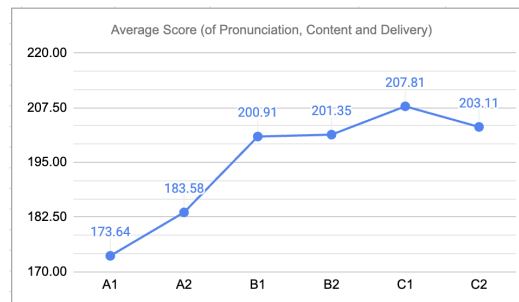


Figure 2: Average combined scores of pronunciation, content and delivery

focusing on content relevance and language complexity in delivery. Based on predefined prompts (see details in Appendix F), the model produces quantified evaluation scores for these dimensions. Additionally, Llama 3 is prompted to generate human-like feedback in the form of constructive suggestions (see detailed examples in Appendix G), offering learners insights into how they can improve both the content and delivery of their spoken language.

3.3 CEFR Classification Module

Due to the lack of available Swedish data, and in order to provide an overall CEFR-based proficiency label for speech performance, we conducted a preliminary calibration of the combined scores generated by the two aforementioned AI modules. This calibration aligns the system’s output with CEFR proficiency levels, using threshold values derived from test results on 55 carefully sampled English speech recordings ranging from A1 to C2, drawn from the COREFL corpus (Lozano et al., 2020) (see Figure 2). Notably, the system demonstrates strong discriminative capability at lower proficiency levels, whereas the distinction between B1 and B2 remains relatively subtle. The observed decline in scores from C1 to C2 is consistent with the known ambiguity of official CEFR descriptors at higher proficiency levels, as previously discussed by Isbell (2017) and Settles et al. (2020).

3.4 Web Implementation and User Experience Design

In our system, the player assumes the role of *Frog*, a character motivated to learn Swedish, and engages with Professowl, a fictional language professor who provides feedback and evaluations of the player’s spoken Swedish. This narrative framing is intended to enhance learner engagement by embed-

ding assessment within an interactive and playful context.



Figure 3: Professowl guiding Frog through the pronunciation assessment tasks

The dialogue flow begins with Professowl guiding Frog through reading two Swedish sentences of different CEFR proficiency levels and then a free speech on the topic of “self introduction”. Professowl gives corresponding feedback including scores and suggestions in an encouraging way.

4 Preliminary Results and Discussion

Given the limited availability of Swedish L2 speech data, we collected five original sets of preliminary speech samples from L2 learners at varying proficiency levels (see detailed results in Appendix C and D). These samples were manually evaluated by an experienced Swedish language instructor using the same scoring metrics employed by the automated system, enabling a direct comparison between human and machine assessments. While the dataset remains modest relative to high-resource languages such as English, it establishes an essential foundation and provides a baseline for subsequent analyses.

Due to the scale difference between Azure assessment metrics (0 to 100%) and our assessment metrics (1 to 5 Likert Scale) for the human rating, the system assessment scores were proportionally converted to 1 to 5 point scale based on thresholds at 20%, 40%, 60%, 80%.

As illustrated in Figure 4, a general alignment can be observed between the system-generated assessments and those provided by the human evaluator. However, the system is currently unable to assess prosody in Swedish, resulting in missing scores for this dimension. Furthermore, limitations in handling Swedish phonological characteristics lead to a rigid, word-by-word evaluation approach. For instance, commonly (phonologically) reduced

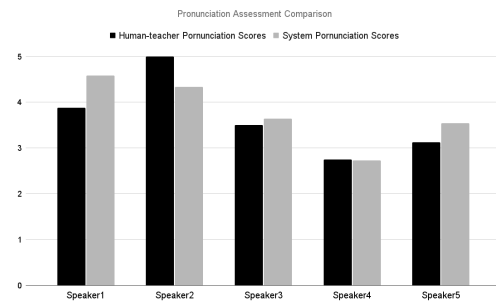


Figure 4: Average Pronunciation Scores Comparison

function words in Swedish—such as *att* ‘to’ and *i* ‘in/at’ were frequently misclassified as “weak words” even when produced fluently. This issue is highlighted in the comparison of strong and weak word assessments between the system and the human evaluator (Appendix E).

5 Conclusion and Future Work

In this paper, we present an initial prototype of a speech assessment system designed for Swedish. Our speech analyzer generates meaningful evaluation scores, provides reference word lists based on word-level pronunciation performance, and delivers both general feedback and personalized suggestions to support language learning.

The system combines Microsoft Azure’s speech services with large language models to divide the assessment process into distinct tasks, each handled by separate tools. The game-like user experience design intends to promote learners’ engagement (Hung, 2017; Hung et al., 2018). This approach demonstrates the potential of digital language learning tools in low-resource settings.

For future work, we plan to focus on several key aspects to improve the effectiveness and reliability of our system. First, we aim to achieve greater integration stability by stabilizing the speech services and embedding appropriate transition cues. This will reduce unintended delays during gameplay and ensure a smoother user experience throughout the learning process.

Second, we intend to enhance our phonological analysis capabilities by improving the system’s ability to recognize and analyze phonological patterns in naturally spoken Swedish. This further development will enable more precise assessment of learners’ pronunciation and speaking skills, particularly the nuances of Swedish phonology that are crucial for assessing language proficiency.

Third, we plan to significantly expand our data by collecting a larger and more comprehensive dataset covering learners at all proficiency levels from A1 to C2. This expanded dataset will better represent the full spectrum of Swedish learners and enable more robust training and reliable evaluation of our assessment algorithms.

Finally, we are focusing on improved validation procedures. To do this, we will engage additional teachers and annotators to rate language samples, thus confirming the accuracy of our automated assessments through inter-rater reliability measures. Furthermore, we plan to calibrate our CEFR classification system using authentic data from Swedish second language learners. This should help ensure that our proficiency level assignments conform to established CEFR standards and reflect the specific characteristics of Swedish language acquisition.

Limitations

This study presents a prototype system for automatic speech assessment in Swedish as a second language, but several limitations should be acknowledged. First, the evaluation relies on a small and preliminary dataset consisting of only five learner speech samples, which restricts the generalizability and statistical robustness of the findings. Second, the calibration of CEFR levels was based on English L2 data due to the lack of sufficient annotated Swedish learner corpora, which may have introduced cross-linguistic biases in proficiency classification. Third, the Azure speech assessment module lacks support for prosodic features in Swedish, limiting the system's ability to fully capture suprasegmental aspects of pronunciation. Additionally, the rigid word-by-word evaluation method often misinterprets function word reductions common in fluent speech, potentially penalizing natural speaking patterns. Furthermore, despite the robustness of the Microsoft Azure speech assessment analysis, the reliance limits replicability of this work. Other open-source alternatives such as Whisper-based assessment will be considered in future research to maximize the accessibility of the system.

Ethical Concerns

The development and deployment of automated language assessment tools raise several ethical considerations. Firstly, the system's reliance on proprietary and opaque evaluation mechanisms—such as

Azure's speech scoring—may reinforce biases that are not easily observable or correctable by developers or users. Secondly, collecting and processing learner speech data involves privacy risks and must comply with ethical data handling standards, including informed consent and secure data storage. In this study, all participants were aged 18 or over and provided express consent for their speech data to be used for research purposes. Special care should be taken if the system is later extended to include minors or vulnerable populations, particularly in educational game-based settings. Lastly, while large language models can offer helpful feedback, they may inadvertently reinforce normative language ideologies or reflect implicit biases. To ensure fairness, pedagogical relevance, and user well-being, ongoing evaluation and human oversight are essential throughout system development and deployment.

Acknowledgements

This work was financially supported by the Royal Society of Arts and Sciences in Gothenburg (Kungliga Vetenskaps- och Vitterhets-Samhället i Göteborg).

References

- J Charles Alderson. 2007. The CEFR and the need for more research. *The Modern Language Journal*, 91(4):659–663.
- Ramon F Brena, Evelyn Zuvirie, Alan Preciado, Aristh Valdiviezo, Miguel Gonzalez-Mendoza, and Carlos Zozaya-Gorostiza. 2021. Automated evaluation of foreign language speaking performance with machine learning. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 15(2):317–331.
- Younghoon Chang, Seongyong Lee, Siew Fan Wong, and Seon-phil Jeong. 2022. AI-powered learning application use and gratification: an integrative model. *Information Technology & People*, 35(7):2115–2139.
- Angxuan Chen, Yuyue Zhang, Jiyu Jia, Min Liang, Yingying Cha, and Cher Ping Lim. 2024. A systematic review and meta-analysis of AI-enabled assessment in language learning: Design, implementation, and effectiveness. *Journal of Computer Assisted Learning*.
- Council of Europe. 2001. *Common European framework of reference for languages: Learning, teaching, assessment*. Cambridge University Press.
- Paul Daniels. 2022. Auto-Scoring of Student Speech: Proprietary vs. Open-Source Solutions. *TESL-EJ*, 26(3):n3.

- Gilbert Dizon. 2020. Evaluating intelligent personal assistants for L2 listening and speaking development. *Language Learning & Technology*, 24(1):16–26.
- Yaroslav Getman, Nhan Phan, Ragheb Al-Ghezi, Ekaterina Voskoboinik, Mittul Singh, Tamas Grosz, Mikko Kurimo, Giampiero Salvi, Torbjørn Svendsen, Sofia Strömbergsson, et al. 2023. Developing an AI-assisted low-resource spoken language learning app for children. *IEEE Access*.
- 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.
- Shi Huawei and Vahid Aryadoust. 2023. A systematic review of automated writing evaluation systems. *Education and Information Technologies*, 28(1):771–795.
- Hsiu-Ting Hung. 2017. Clickers in the flipped classroom: Bring your own device (byod) to promote student learning. *Interactive Learning Environments*, 25(8):983–995.
- Hsiu-Ting Hung, Jie Chi Yang, Gwo-Jen Hwang, Hui-Chun Chu, and Chun-Chieh Wang. 2018. A scoping review of research on digital game-based language learning. *Computers & Education*, 126:89–104.
- Daniel R Isbell. 2017. Assessing C2 writing ability on the Certificate of English Language Proficiency: Rater and examinee age effects. *Assessing Writing*, 34:37–49.
- Blair Lehman, Lin Gu, Jing Zhao, Eugene Tsuprun, Christopher Kurzum, Michael Schiano, Yulin Liu, and G Tanner Jackson. 2020. Use of adaptive feedback in an app for English language spontaneous speech. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part I 21*, pages 309–320. Springer.
- 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 *Swedish Language Technology Conference and NLP4CALL*, pages 126–136.
- Shawn Loewen, Daniel R Isbell, and Zachary Sporn. 2020. The effectiveness of app-based language instruction for developing receptive linguistic knowledge and oral communicative ability. *Foreign Language Annals*, 53(2):209–233.
- Cristóbal Lozano, Ana Díaz-Negrillo, and Marcus Callies. 2020. Designing and compiling a learner corpus of written and spoken narratives: COREFL. *What’s in a Narrative*, pages 21–46.
- Microsoft. 2024. Azure Speech Studio. <https://speech.microsoft.com/portal>. Accessed: 2024-01-28.
- Mozilla Foundation. 2020. Common Voice Dataset. <https://commonvoice.mozilla.org/>. Accessed: 2025-04-15.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for computational Linguistics*, 8:247–263.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. *LLaMA: Open and Efficient Foundation Language Models*. Preprint, arXiv:2302.13971.
- Elena Volodina, David Alfter, and Therese Lindström Tiedemann. 2024. Profiles for Swedish as a second language: lexis, grammar, morphology. In *Huminfra Conference*, pages 10–19.
- Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice, Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, et al. 2019. The SweLL language learner corpus: From design to annotation. *Northern European Journal of Language Technology (NEJLT)*, 6:67–104.
- Klaus Zechner and Keelan Evanini. 2019. *Automated speaking assessment: Using language technologies to score spontaneous speech*. Routledge.
- Ruofei Zhang and Di Zou. 2022. Types, purposes, and effectiveness of state-of-the-art technologies for second and foreign language learning. *Computer Assisted Language Learning*, 35(4):696–742.
- Bin Zou, Yiran Du, Zhimai Wang, Jinxian Chen, and Weilei Zhang. 2023. An investigation into artificial intelligence speech evaluation programs with automatic feedback for developing EFL learners’ speaking skills. *Sage Open*, 13(3).

A Assessment Criteria

Figure 5 provides detailed descriptors for the pronunciation metrics used in our assessment system.

Assessment Criteria				
Item	Descriptor	Rating scale	Details	
Pronunciation	Accuracy	Accuracy indicates how closely the phonemes match a native speaker's pronunciation.	"1 to 5" Likert scale	
	Fluency	Fluency indicates how closely the speech matches a native speaker's use of silent breaks between words.	"1 to 5" Likert scale	
	Prosody	Prosody indicates how natural the given speech is, including stress, intonation, speaking speed, and rhythm.	"1 to 5" Likert scale	*see scoring descriptors
	Completeness	Completeness of the speech, calculated by the ratio of pronounced words to the input reference text.	"1 to 5" Likert scale	
	Strong words	In other words, completeness indicates how complete the speech is compared with the reference text. (Is the speech missing any words?)	0-3 words	
	Weak words	The words that were comparatively pronounced well The words that were comparatively pronounced poorly	0-3 words	e.g. det, här, är (from best to good; write at most three) e.g. trivs, att, bra (from worst to not accurate; write at most three)
Content & Delivery	Content	The quality of the content (the richness and relevance of the speech) of the speech.	Free Comments	e.g. "The self-introduction is pretty relevant but quite narrow in topics as the speaker only talked about the education background."
	Delivery	The quality of the delivery (grammatical structure, natural language use, etc.) of the speech.	Free Comments	e.g. "The grammar used in the speech is mostly very simple. OOO is not a natural Swedish sentence, XXX is more commonly in this case."
Overall	Overall Conclusion (regarding the speech proficiency level, strengths and weaknesses, suggestion for future study, e.g. what to focus on regarding pronunciation or free speech)		Free Comments	e.g. "The overall speech shows that the speaker has the basic knowledge of Swedish. Given that the complexity of the speech is low and the rhythm is limited, the speaker could be of a CEFR level between A2 and B1. The speaker is very accurate at pronouncing Swedish vowels such as ä and ö, but not very good at the 'r' sound. The 'r' sound is pronounced similar to 'l' instead. For future study, the speaker might want to improve on the 'r' sound, listening to real Swedish conversations as to be familiar with the natural rhythm and intonation."

Figure 5: Assessment criteria for human-teacher assessment

B Detailed Scoring Descriptors

Table 1 provides detailed descriptors for the pronunciation metrics used in our assessment system.

Score	Accuracy	Completeness	Fluency	Prosody
1	Incomprehensible speech with almost no sounds that are accurate	Missing many important words (< 60%)	Very snatchy speech with frequent unnatural breaks	No variation in stress or intonation, or the rhythm is completely off
2	Many obvious errors in pronunciation, difficult to understand	Several missing words (60–75%)	Frequent hesitations and stops	Unnatural rhythm, intonation and stress patterns
3	Some noticeable errors but generally accurate and understandable	Most words included with some minor omission (75–85%)	Generally fluent flow with some unnatural stops	Some natural stress and intonation patterns
4	High accuracy with minor errors that don't affect comprehension	Nearly complete (85–95% coverage)	Generally smooth speech with occasional pauses	Generally appropriate stress, rhythm and intonation
5	Most sounds are perfectly correct, native-like speaking	Complete (95–100% coverage)	Natural, native-like speech flow with appropriate pauses	Native-like rhythm, stress, and intonation

Table 1: Detailed scoring descriptors for pronunciation metrics

C Preliminary Test Results (Human Assessment)

Figure 6 provides the human teacher's assessment results on the test speech samples.

Student ID	Teacher Assessment															Overall comment	Estimated CEFR level (A1-B2)
	Speech 1					Speech 2					Content & Delivery						
	Acc.	Flu.	Pro.	Comp.	Strong words	Weak words	Acc.	Flu.	Pro.	Comp.	Strong words	Weak words	Content	Free Speech	Delivery		
#001	4	4	3	5	Jag - bott i Sverige i tre - och trivs mycket bra här	har, är	3	4	4	4	Det krävs omfattande åtgärder för att hantera klimatförändringarna.	krävs, åtgärder, klimatförändringarna	The self-introduction is pretty standard, but only focused on hobbies and where the person lives.	The grammar used in the speech is simple but correct.	The speech samples indicates that this person knows quite a lot of Swedish, but has focused more on vocabulary than prosody and accuracy when learning the language.	B1	
#002	5	5	5	5	Jag har bott i Sverige i tre år och trivs mycket bra här	-	5	5	5	Det krävs omfattande åtgärder för att hantera klimatförändringarna	-	The self-introduction gives information about different aspects of personal lives within a few sentences.	The grammar used in this speech is typical for a native Swedish speaker giving an informal self-	The speech samples indicates that this person is a native Swedish speaker born in the southern part of Sweden.	C2		
#003	3	4	3	5	- har bott i Sverige i tre år och - mycket bra här	jag, trivs	2	4	4	3	Det - åtgärder för att hantera -.	krävs, omfattande, klimatförändringarna	The self-introduction gives information about different aspects of personal lives within a few sentences.	The grammar used in the speech is simple but correct.	The speech samples indicates that this is a student with a Swedish language level ranging from elementary to intermediate level. The fluency and prosody is accurate for the level, but the student struggles with pronunciation of some long words and specific letters (r).	A2/B1	
#004	2	4	3	5	jag, bott, i Sverige, och, mycket	tre, trivs, bra	1	2	1	4	att hantera	krävs, omfattande, åtgärder, klimatförändringarna	The self-introduction is typical for a beginner to elementary level student, and includes all the content you would expect.	The grammar used in the speech is simple but correct.	The speech samples indicates that this is a beginner level student, or an elementary student who struggles a bit with pronunciation.	A1/A2	
#005	3	3	2	5	har, bott, i Sverige, trivs, mycket bra	är	2	3	2	5	det, hantera	åtgärder	The self-introduction is accurate, but quite short.	The grammar used in the speech is simple but correct.	The speech samples indicates that this is a beginner to elementary level student. It is difficult to tell which one from the speech samples, since the student uses simple grammar structures, and presents various levels of pronunciation in the different tasks.	A2	

Figure 6: Preliminary results of human-teacher assessment

D Preliminary Test Results (System Assessment)

Figure 7 provides the system's assessment results on the test speech samples.

System Assessment														Estimated CEFR level (A1-B2)		
Student ID	Speech 1 Jag har bott i Sverige i tre år och trivs mycket bra här"						Speech 2 Det krävs omfattande åtgärder för att hantera klimatförändringarna.						Content & Delivery		Overall comment	
	Acc.	Flu.	Comp.	Conf.	Strong words	Weak words	Acc.	Flu.	Comp.	Conf.	Strong words	Weak words	Content			Delivery
#001	90	99	92	92.2	har, bott, Sverige, tre, år	i	86	98	88	88.8	Det, krävs, omfattande, åtgärder, hantera	att	60	55	...showing basic vocabulary and structure, but could be more engaging. Instead of saying "Jag är 29 år gammal", try using a more dynamic phrase like "Jag har precis fyllt 30" to add some interest. Add more specific details about your experiences in Sweden, e.g., what you do or enjoy here. Consider using more complex sentence structures, such as combining two clauses with "däremot".	B2/C1
#002	75	88	77	78	har, Sverige, tre, trivs, mycket	bott, i, och	86	99	100	91.4	Det, krävs, omfattande, för, hantera	att	60	55	...showing basic vocabulary and structure, but could be more engaging and detailed. Try to include your name and how you heard about the course in Sweden. Be more specific when describing your interests in drawing and playing games like "Draken" and "Ademona". Practice using more complex sentence structures, such as conditional or subordinate clauses.	B2/C1
#003	63	75	69	70	Sverige, och, mycket, bra, här	Jag, har, tre, trivs	70	85	75	75	hantera, klimatförändringarna	Det, krävs	60	40	...showing good basic vocabulary but lacking in content and sentence structure. Add more details about your background, interests, or goals. Practice using complete sentences and varying sentence lengths. Try to include a specific reason for being in Sweden.	A1/A2
#004	70	67	69	68	Jag, i, är, och, mycket	Sverige, i, tre, trivs, bra	51	35	38	38.8	att	Det, krävs, åtgärder, för, hantera	70	55	...indicating a good start but some areas for improvement. Consider adding more specific details about your work or interests. Practice using correct verb forms, e.g., "Jag har flyttat" instead of "Jag flyttar hits". Use more precise vocabulary when describing your age ("30 år gammal" could be "trettioårig")	A1/A2
#005	81	69	77	73	Jag, har, bott, och, trivs	Sverige, tre, år	68	72	62	65.2	hantera	åtgärder, För, att	70	50	...showing basic vocabulary and simple sentence structure. Consider adding more details about yourself, such as your hobbies or background. Practice using more Formal language For a university setting. Try touse more cohesive linking words between sentences	A2/B1

Figure 7: Preliminary results of system assessment

E Preliminary Test Results (Strong/Weak Words Comparison)

Table 2 shows the assessment results comparison between the system and the teacher as for strong/weak words pronunciation.

Sentence	Student ID	Evaluator	Strong words	Weak words
S1	#001	System Teacher	här, bott, Sverige, tre, är Jag, bott, i, Sverige, tre, och trivs, mycket, bra, här	i här, är
	#002	System Teacher	här, Sverige, tre, trivs, mycket Jag, har, bott, i, Sverige, tre, år, och trivs, mycket, bra, här	bott, i, och –
	#003	System Teacher	Sverige, och, mycket, bra, här har, bott, i, Sverige, tre, år, och, mycket, bra, här	Jag, har, tre, år, trivs jag, trivs
	#004	System Teacher	Jag, i, är, och, mycket jag, bott, i Sverige, och, my- cket	Sverige, i, tre, trivs, bra tre, trivs, bra
	#005	System Teacher	jag, har, bott, och, trivs har, bott, i Sverige, trivs, my- cket, bra	Sverige, tre, år år
S2	#001	System Teacher	Det, krävs, omfattande, åtgärder, hantera Det, krävs, omfattande, för, att, hantera	att åtgärder, klimatförändringar
	#002	System Teacher	Det, krävs, omfattande, för, hantera Det, krävs, omfattande, åtgärder, för, att, hantera, klimatförändringar	att –
	#003	System Teacher	hantera, klimatförändringarna Det, åtgärder, för, att, hantera	Det, krävs krävs, omfattande, klimatförändringar
	#004	System Teacher	att att, hantera	Det, krävs, åtgärder, för, hantera krävs, omfattande, åtgärder, klimatförändringarna
	#005	System Teacher	hantera det, hantera	åtgärder, För, att åtgärder

Table 2: Comparison of strong and weak word analysis between system and teacher

F LLaMA Model Prompting Details

The "content and delivery assessment module" employs Llama 3.1 with carefully designed and tested prompts to ensure consistent feedback. The prompt details and model configuration are as follows:

F.1 Detailed Prompt

The following prompt template is passed with relevant values for evaluating the content and delivery of every speech input:

```
You are Professowl, a Swedish language teacher. Analyze the following student's self-introduction in Swedish: "$input"
```

```
Provide feedback in this JSON format:
```

```
{
  "analysis": {
    "relevance": number, // 0-100, how well the content works as a self-introduction
    "complexity": number, // 0-100, language complexity level
    "feedback": string, // One sentence including both relevance and complexity scores
    "suggestions": string[] // 2-3 short, specific suggestions in English
  }
}
```

```
Focus on:
```

- Content and relevance for self-introduction
- Sentence structure and vocabulary use
- Grammar and language flow

```
Example response:
```

```
{
  "analysis": {
    "relevance": 80,
    "complexity": 65,
    "feedback": "Your self-introduction is quite relevant (80/100) with moderate language complexity (65/100), showing good basic vocabulary and structure.",
    "suggestions": [
      "Try using more descriptive adjectives when talking about your interests",
      "Add information about your future plans in Sweden",
      "Practice using more complex sentence connections"
    ]
  }
}
```

```
Always include both scores in the feedback sentence.
```

F.2 Model Configuration

To ensure consistent and reliable outputs, the following configuration parameters are used:

- **Model:** Llama 3.1
- **Temperature:** 0.3
- **Output format:** JSON
- **Deployment:** Local Ollama instance

G System Feedback Examples

The following examples demonstrate system feedback generated by the LLaMA model for speech samples. The feedback structure is designed to be encouraging and provides score-based assessment as well as actionable suggestions.

G.1 Example 1

"Your self-introduction is somewhat relevant (70/100) with relatively low language complexity (55/100), indicating a good start but some areas for improvement. Consider adding more specific details about your work or interests. Practice using correct verb forms, e.g., 'Jag har flyttat' instead of 'Jag flyttar'. Use more precise vocabulary when describing your age 30 år gammal' could be 'trettiofyre år gammal)"

G.2 Example 2

"Your self-introduction is somewhat relevant (70/100) with relatively low language complexity (50/100), showing basic vocabulary and simple sentence structure. Consider adding more details about yourself, such as your hobbies or background. Practice using more formal language for a university setting. Try to use more cohesive linking words between sentences."

Multilingual Grammatical Error Annotation: Combining Language-Agnostic Framework with Language-Specific Flexibility

Mengyang Qiu^{1,2} Tran Minh Nguyen² Zihao Huang² Zelong Li³ Yang Gu²
Qingyu Gao² Siliang Liu² Jungyeul Park^{2,3}

¹Trent University, Canada ²Open Writing Evaluation, France

³University College London, UK ⁴The University of British Columbia, Canada

<http://open-writing-evaluation.github.io>

Abstract

Grammatical Error Correction (GEC) relies on accurate error annotation and evaluation, yet existing frameworks, such as *errant*, face limitations when extended to typologically diverse languages. In this paper, we introduce a standardized, modular framework for multilingual grammatical error annotation. Our approach combines a language-agnostic foundation with structured language-specific extensions, enabling both consistency and flexibility across languages. We reimplement *errant* using *stanza* to support broader multilingual coverage, and demonstrate the framework’s adaptability through applications to English, German, Czech, Korean, and Chinese, ranging from general-purpose annotation to more customized linguistic refinements. This work supports scalable and interpretable GEC annotation across languages and promotes more consistent evaluation in multilingual settings. The complete codebase and annotation tools can be accessed at https://github.com/open-writing-evaluation/jp_errant_bea.

1 Introduction

Grammatical Error Correction (GEC), which aims to automatically detect and correct errors in written text, has emerged as one of the most important and widely studied tasks in Natural Language Processing (NLP) for educational applications, particularly those supporting language learning and writing improvement. It benefits both native speakers (L1), by enhancing clarity and fluency in their writing, and non-native learners (L2), by providing immediate, structured feedback that reinforces correct grammatical patterns, boosts writing confidence, and, ultimately, supports language development and acquisition (Marjokorpi, 2023; Van Beuningen et al., 2012). Over the years, the lion’s share of research has focused on advancing GEC systems—evolving from rule-based and statistical approaches to neural

architectures, such as neural machine translation with transformers (Zhao et al., 2019) and, more recently, prompting-based approaches built on large language models (Zeng et al., 2024; for a comprehensive review, see Bryant et al., 2023).

Yet, automatic error annotation and evaluation play an equally critical role in GEC. Error annotation identifies and categorizes linguistic errors, while evaluation measures how effectively GEC systems correct them. Together, these two components help establish standardized benchmarks, influencing everything from system development to the quality of corrections eventually delivered to users. However, despite their importance, they have historically received less attention and are often treated as ancillary to system development and dataset creation.

Among existing tools for automatic error annotation and evaluation (e.g., M², Dahlmeier and Ng, 2012; GLEU, Napoles et al., 2015), *errant* (Error ANnotation Toolkit) has established itself as the *de facto* framework for English GEC. What makes *errant* stand out is its detailed linguistic annotations, with a total of 55 possible error types for English (Bryant et al., 2017). *errant*’s significance was solidified in the *Building Educational Applications 2019 Shared Task: Grammatical Error Correction (BEA-2019)*, where it was used to standardize multiple datasets and served as the official scorer (Bryant et al., 2019).

While this toolkit has proven effective for English, further refinements are needed to improve its versatility and adaptability, especially in multilingual scenarios. Recent years have seen growing interest in multilingual GEC, as demonstrated by initiatives like the *MultiGEC-2025 Shared Task*, which brought together efforts across twelve typologically diverse European languages (Masciolini et al., 2025a,b). However, this surge in interest has outpaced the development of consistent multilingual annotation resources.

As noted by Masciolini et al. (2025a), only three languages in MultiGEC—namely Czech, German, and Greek—have received errant-style annotation. For the remaining languages, the authors acknowledge that, due to limited time and resources, they implemented only coarse-grained alignment between original and corrected texts to support holistic scoring, without access to the kind of detailed error analysis enabled by errant for English. Even in existing adaptations of errant for various languages, implementations vary considerably in their design choices—ranging from annotation label schemes to tokenization and part-of-speech (POS) tagging tools—and differ in the level of granularity applied to language-specific error types. Although differences in orthographies and morphosyntactic structures across languages are unavoidable, greater consistency in annotation practices is highly desirable.

To address these challenges, our goal is to develop a consistent and reusable framework for grammatical error annotation that can be readily adapted across typologically diverse languages. Drawing inspiration from the original errant’s dataset-agnostic design, we extend its core philosophy to multilingual settings by separating the annotation pipeline into two components: a shared architecture that applies across languages, and optional extensions tailored to language-specific features. Even within the language-specific layer, we introduce structured templates for common error types, such as spelling, word order, and word boundary errors, which can be reused or adapted across languages with similar orthographic or syntactic patterns. In addition, our implementation relies on the stanza toolkit for tokenization and POS tagging, which provides standardized processing pipelines for over 70 languages (Qi et al., 2020), allowing our framework to be readily extended to annotate new GEC datasets of other languages when they become available.

The rest of the paper is organized as follows: §2 reviews the original errant framework, discusses challenges in its multilingual adaptations, and motivates the use of stanza for more consistent cross-linguistic preprocessing. §3 introduces our proposed grammatical error typology, which combines a language-agnostic core with structured, language-specific extensions. §4 presents our reimplementation of English errant and demonstrates the framework’s applicability to multiple languages, ranging from generic use in European languages, to minor

template refinements for Korean, and deeper customization for Chinese. Finally, §5 summarizes our contributions and emphasizes the framework’s flexibility and extensibility for multilingual GEC.

2 Background and Related Work

2.1 Description of errant

errant is a unified framework for error annotation and evaluation in English GEC. It provides a rule-based, dataset-agnostic approach for extracting and categorizing edits between original and corrected sentences, making it a crucial tool for system evaluation and benchmarking (Bryant et al., 2017).

At the core of its annotation pipeline is a linguistically enhanced alignment algorithm that identifies edit boundaries between sentence pairs. This algorithm, originally proposed by Felice et al. (2016), extends the Damerau-Levenshtein distance with a linguistically informed cost function that considers part-of-speech tags, lemmas, and character similarity. Unlike surface-level edit distance, this method prioritizes alignments between tokens that are syntactically or morphologically related (e.g., *meet* and *meeting*), and handles both one-to-one edits and multi-token reordering. A rule-based merging strategy is then applied to combine adjacent edits where appropriate, based on patterns frequently observed in learner data, such as phrasal verb edits. This alignment process significantly improves the consistency and quality of extracted edits (Felice et al., 2016).

Following alignment, errant applies a rule-based annotation scheme to categorize edits into fine-grained grammatical error types, enabling both comprehensive feedback and error-type evaluation. Specifically, it defines 25 primary error types based on POS and morphological properties obtained from spaCy¹, and further classifies them into three edit operations: Missing, Unnecessary, and Replacement, resulting in a total of 55 possible error types (e.g., R:VERB:TENSE indicates a replacement error related to verb tense). To store annotations, errant generates output in M2 format, the standard representation for GEC annotations since its adoption in the *CoNLL-2013 Shared Task* (Ng et al., 2013). Each annotated sentence consists of the original tokenized text (denoted by an *S* line) followed by one or more error annotation lines (*A* lines). Each *A* line specifies the error span, the

¹<https://spacy.io>

error type, the suggested correction, and additional metadata (see Figure 1 for an example in English).

```
S This are a sentence .
A 1 2|||R:VERB:SVA|||is|||-REQUIRED-|||NONE|||0
A 3 3|||M:ADJ|||good|||-REQUIRED-|||NONE|||0
```

Figure 1: Example of an annotated sentence in M2 format from *BEA-2019*.

With edits extracted and categorized in a standardized format, errant can then be used to systematically evaluate GEC system outputs against gold-standard references. It calculates precision and recall between system-generated edits and gold-standard corrections and utilizes a harmonic mean $F_{0.5}$ score, which weights precision twice as much as recall to prioritize accurate and contextually appropriate corrections over excessive edits. Thanks to its detailed annotation schema, errant supports multi-granularity evaluation—analyzing system effectiveness not only at the overall level but also across specific error types and edit operations, enabling a fine-grained and transparent assessment of GEC models.

In *BEA-2019*, errant was used to standardize multiple datasets, some of which were annotated using different error type frameworks, while others lacked annotations entirely. This allowed for error distribution comparisons across datasets that were previously hindered by these annotation discrepancies. In addition, errant facilitated multi-level system evaluation by supporting error-type analysis across 24 main categories for all 21 participating teams². This enabled a detailed assessment of each system’s strengths and weaknesses and made it easier to identify which error types were the most challenging to correct (Bryant et al., 2019).

While errant provides a linguistically informed foundation for GEC annotation and evaluation, it is not without limitations. One minor issue is its tendency to overuse the OTHER category (i.e., unspecified errors), leading to less precise error categorization. For instance, certain errors that could be classified as specific grammatical types (e.g., verb tense or prepositions) are instead grouped under OTHER (Korre and Pavlopoulos, 2020).

Another issue, as discussed in Wang et al. (2025), arises in end-to-end evaluation scenarios. errant assumes pre-defined sentence boundaries, and mis-

²errant defines 25 categories, including UNKnown (error detected but unable to be corrected; Bryant et al., 2017). In *BEA-2019*, this category was not included.

alignment can result in an inability to generate evaluation results between gold-standard references and system outputs. However, in real-world GEC applications, such as learner essays, inconsistencies in sentence segmentation are a common issue, often caused by differences in preprocessing steps. To address this, Wang et al. (2025) introduced joint-preprocessing errant, incorporating an alignment-based approach to detect and resolve segmentation discrepancies before evaluation.

2.2 Challenges in existing multilingual adaptations of errant

Given its demonstrated success in English, errant has been adapted to multiple languages, including Arabic (Belkebir and Habash, 2021), Chinese (Hinson et al., 2020; Zhang et al., 2022; Gu et al., 2025), Czech (Náplava et al., 2022), German (Boyd, 2018), Greek (Korre et al., 2021), Hindi (Sonawane et al., 2020), and Korean (Yoon et al., 2023). While these adaptations have enabled broader use of errant-style annotation, they also reveal several challenges that arise when extending the framework to languages with a range of orthographic and morphosyntactic characteristics.

Inconsistent annotation labels A minor issue in multilingual adaptations of errant is inconsistent annotation labels for similar error types. The original errant for English defines three edit operations: Missing, Unnecessary, and Replacement, while treating word order (WO) as a main error category, similar to NOUN or VERB errors. In *errant_zh*, an adaptation for Chinese, these operations were denoted as insertion, deletion, substitution, and transposition (Hinson et al., 2020). Meanwhile, *ChERRANT*, another Chinese adaptation, later revised them to Missing, Redundant, Substitute, and Word-order (Zhang et al., 2022). While these differences do not affect core functionality, the lack of consistent labeling across adaptations can create confusion. Nevertheless, this issue is relatively straightforward to address, as it primarily involves terminology standardization.

Inconsistent preprocessing tools A moderate challenge in multilingual GEC annotation lies in linguistic preprocessing tools, particularly word segmentation and POS tagging. While *spaCy*, the default NLP library in errant for English, supports multiple languages, its effectiveness varies across linguistic systems, prompting many adaptations to incorporate alternative tools. For exam-

ple, German errant retained much of the spaCy pipeline but found its lemmatization insufficient, replacing it with TreeTagger for better accuracy (Boyd, 2018). For non-European languages, entirely different tools are used, such as Kkma POS Tagger for Korean KAGAS (Yoon et al., 2023) and LTP (Language Technology Platform) for Chinese ChERRANT (Zhang et al., 2022).

Although these variations allow for language-specific optimizations, different tokenization strategies and POS tagging schemes can lead to discrepancies in how errors are identified and classified. This is particularly problematic for multilingual GEC models, where standardized evaluation across multiple languages is crucial. Since a system’s measured performance is inherently tied to how its errors are annotated, such variations can obscure true system similarities or differences and compromise the reliability of multilingual benchmarks.

Inconsistent annotation granularity A more significant challenge in multilingual GEC annotation involves the varying levels of granularity for language-specific errors. While errant provides detailed error categories for English, adaptations to other languages, especially non-European languages, often fail to maintain this level of detail. For instance, errant_zh uses only four basic edit operations at the character level, without POS information (Hinson et al., 2020).

Recent work has begun addressing this limitation by introducing more fine-grained annotations tailored to specific linguistic properties. For example, Gu et al. (2025) propose a refined error typology for Chinese that accounts for phonetic similarity, visual similarity, and other structural errors specific to Chinese. While this framework was developed for Chinese, many of its principles can be readily applied to languages with similar logographic orthographies.

2.3 stanza as a multilingual alternative to spaCy

The original errant framework relies on spaCy for preprocessing tasks such as tokenization and POS tagging. However, spaCy’s multilingual capabilities are relatively limited, covering only a small number of languages and exhibiting inconsistent performance across linguistic families. This has contributed to the fragmented landscape of language-specific adaptations in prior errant variants.

To promote cross-lingual consistency, our implementation adopts stanza (Qi et al., 2020), a fully neural pipeline trained on Universal Dependencies (UD) and other multilingual corpora. stanza supports over 70 languages and applies a consistent architecture and UD-based annotation scheme across its modules—including tokenization, multi-word token expansion, POS and morphological tagging, dependency parsing, and named entity recognition³. Benchmark evaluations indicate strong performance across typologically diverse languages.

Crucially, our aim is not to promote a specific tool, but to align the preprocessing stage with the same linguistic principles that underlie our error typology. Like UD, our taxonomy adopts a cross-linguistically consistent core structure with optional language-specific extensions. Using a UD-compatible parser such as stanza ensures that all languages are analyzed under a shared morphosyntactic framework, which is essential for scalable and comparable multilingual grammatical error annotation. In this sense, it is the UD standard, rather than any particular NLP library, that provides the conceptual and practical foundation for our approach.

3 Multilingual Error Typology

An error typology provides a systematic framework for identifying, classifying, and analyzing errors in written text. We propose a two-tiered typology consisting of a language-agnostic foundation and a set of structured, language-specific extensions. The first level includes the widely adopted MRU (Missing, Replacement, Unnecessary) framework, ensuring consistency in annotation and evaluation across linguistic systems. The second level provides a structured template for language-specific extensions, allowing related languages to share annotation strategies and avoid redundant reimplementations. By designing this layered approach, we promote standardization across languages while allowing flexibility for language-specific refinements.

3.1 Language-agnostic error annotation

The MRU framework classifies errors into three core operations: Missing (M), where essential elements are omitted; Replacement (R), where an incorrect element substitutes the correct one; and Unnecessary (U), where superfluous elements cause redundancy.

³<https://stanfordnlp.github.io/stanza/>

Each error is further specified with POS tags for precise categorization.

Missing (M) An essential linguistic element is omitted from a sentence, leading to incomplete or ungrammatical structures. These errors typically involve the absence of words or phrases necessary for grammaticality or semantic clarity, such as missing determiners. In annotation, missing errors are further categorized based on POS tags or syntactic functions. For example, M:NOUN indicates a missing noun.

Replacement (R) An incorrect linguistic element is used in place of the correct one. These errors frequently involve incorrect word forms or inappropriate lexical choices (e.g., R:VERB denotes an erroneous verb substitution). To further reduce ambiguity in annotation, we implement the $R:P_1 \rightarrow P_2$ pattern, where P_1 is replaced by P_2 .

Unnecessary (U) A superfluous linguistic element is present in a sentence, resulting in redundancy or ungrammaticality. These errors often involve extraneous words or phrases that disrupt sentence structure or meaning. Similar to missing and replacement errors, unnecessary errors are annotated with POS information to specify the redundant element. For example, U:DET denotes an unnecessary determiner.

3.2 Language-specific error annotation

To accommodate language-specific characteristics, we introduce a set of structured extensions to the MRU core. Our approach maintains consistency with established annotation schemes such as errant while capturing morphological and syntactic errors unique to different languages. Algorithm 1 presents our proposed classification routine for Replacement errors. Given a pair of word sequences—the source (\mathcal{S}) and the target (\mathcal{T})—the algorithm classifies the error into one of the following types: spelling errors (R:SPELL), word order errors (R:WO), or word boundary errors (R:WB). Spelling similarity is computed using two metrics: phonetic similarity and visual (shape-based) similarity. The thresholds α_1 and α_2 govern sensitivity to phonetic and visual matches, respectively.

The classification uses the following notation:

- \mathcal{S}, \mathcal{T} : word sequences in the source and target sentences.

- $\text{SIM}(\textit{phonetic})$ and $\text{SIM}(\textit{shape})$: similarity functions comparing pronunciation and visual form.
- $\text{SET}(\mathcal{S})$: returns a bag-of-words representation of \mathcal{S} , disregarding word order.
- $\text{MERGE}(\mathcal{S})$: reconstructs a character sequence from the tokenized input (i.e., merging tokens without spaces) to test for boundary alignment.

This structured yet extensible framework allows consistent error categorization across languages, while also accommodating language-specific scripts and segmentation conventions.

Algorithm 1 Pseudo-code for error classification

```

1: function ERRORCLASSIFICATION ( $\mathcal{S}, \mathcal{T}$ ):
2:   if ( $\text{SIM}(\textit{phonetic}) > \alpha_1 \wedge \text{SIM}(\textit{shape}) > \alpha_2$ ) then
3:     return R:SPELL:PHONOGRAPHIC
4:   else if ( $\text{SIM}(\textit{phonetic}) > \alpha_1$ ) then
5:     return R:SPELL:PHONETIC
6:   else if ( $\text{SIM}(\textit{shape}) > \alpha_2$ ) then
7:     return R:SPELL:SHAPE
8:   else if ( $\text{SET}(\mathcal{S}) == \text{SET}(\mathcal{T})$ ) then
9:     return R:WO
10:  else if ( $\text{MERGE}(\mathcal{S}) == \text{MERGE}(\mathcal{T})$ ) then
11:    return R:WB
12:  end if
13:  return {R}

```

Spelling errors We classify spelling errors by their underlying cause: sound-based phonetic similarity (R:SPELL:PHONETIC), visual resemblance in orthographic shape (R:SPELL:SHAPE), or a combination of both (R:SPELL:PHONOGRAPHIC). For sound-based phonetic errors, we introduce a transcription system to represent pronunciation, such as a pronouncing dictionary for English, *pinyin* for Chinese, or romanization for other languages. This allows us to compare words based on their phonetic similarity and identify errors caused by mispronunciation or phoneme substitution. For orthographic shape errors, we assess visual similarity by converting characters into font images and applying similarity metrics. This approach helps detect errors caused by visually similar characters, such as mistyped letters in Latin-based scripts or miswritten strokes in logographic writing systems like Chinese and Japanese. By combining these methods, we systematically classify and analyze spelling errors across different languages.

Word order errors Word order errors are flagged when the source sequence (\mathcal{S}) and the target sequence (\mathcal{T}) contain the same set of words

but differ in arrangement. In such cases, all words from the original sequence are retained, but their relative positions are altered. These errors are particularly common in languages with flexible word order, where reordering affects grammaticality or readability. Identifying and categorizing such errors enables more structured syntactic analysis and improves grammatical error correction.

Word boundary errors Word boundary errors occur when the source sequence (\mathcal{S}) and the target sequence (\mathcal{T}) yield the same sequence after merging their respective word components. These errors typically involve incorrect spacing, where words that should remain separate are mistakenly merged, or conversely, a single word is improperly split into multiple tokens. Since the fundamental content remains unchanged but the segmentation differs, such errors impact readability, syntactic structure, and lexical integrity. Addressing these errors ensures accurate word segmentation and proper grammatical representation.

Figure 2 illustrates representative examples of the three major subtypes of Replacement errors classified by our algorithm.

R:SPELL:PHONETIC	<i>their</i> → <i>there</i>
R:WO	<i>You can help me</i> → <i>Can you help me</i>
R:WB	<i>ice cream</i> → <i>icecream</i>

Figure 2: Examples of Replacement error types: phonetic spelling error (R:SPELL:PHONETIC), word order error (R:WO), and word boundary error (R:WB).

4 Implementation of Multilingual Error Annotation

Our implementation demonstrates how grammatical error annotation can be consistently extended across typologically diverse languages. We begin by reimplementing errant for English using stanza and validating its performance. We then apply the same system to other European languages without language-specific modules. For Korean, we introduce targeted refinements using language-specific templates. Finally, for Chinese, we show how deeper customization can be incorporated by modifying segmentation and retraining the stanza pipeline.

4.1 Reimplementing errant for English

We reimplemented errant using stanza for POS tagging and dependency parsing, as described in §2.3. This enables our annotation system to be

more consistent across languages while preserving the linguistic precision required for English-specific grammatical labels.

We integrated the English-specific classification module from the original errant, which identifies detailed grammatical error types, such as NOUN:POSS for possessive noun suffix errors. This module relies on universal POS tags (Petrov et al., 2012) and dependency relation tags to categorize errors. For instance, if the first token in an edit is tagged as PART and its dependency relation is case:poss, the classifier assigns the NOUN:POSS label accordingly.

A key distinction between the original errant and our implementation lies in error categorization. As illustrated in Figure 3, errant originally annotates *that is* as a missing OTHER error, whereas our implementation classifies it more precisely as a missing PRON error. Additionally, we refine verb annotation by distinguishing auxiliary verbs in passive constructions, categorizing *is played* as a Replacement error from VERB to AUX VERB. These refinements enhance interpretability by providing more specific and linguistically meaningful labels for complex constructions.

Original errant:	
S	Volleyball is a sport play every place ...
A 4 4	M:OTHER that is REQUIRED -NONE- 0
A 4 5	R:VERB:FORM played REQUIRED -NONE- 0
Our implementation:	
S	Volleyball is a sport play every place ...
A 4 4	M:PRON that REQUIRED -NONE- 0
A 4 5	R:VERB → AUX VERB is played REQUIRED -NONE- 0

Figure 3: Differences between errant and our implementation

To evaluate the overall alignment, we compared both implementations using outputs from the state-of-the-art GEC system, T5 (Rothe et al., 2021). The results in Table 1 support that our implementation reproduces errant’s scores, with only minor variations.

	TP	FP	FN	Prec	Rec	F _{0.5}
errant	2589	1639	4030	0.6123	0.3911	0.5501
Ours	2565	1613	4028	0.6139	0.3890	0.5503

Table 1: GEC results for English using T5

4.2 Applying universal annotation to European languages

Without language-specific classification modules, our grammatical error annotation system remains capable of generating generic error annotations using the core MRU framework combined with POS

Czech	S Mám velkou rodinu , tak nemohla jsem mít naději , že něco dostanu .
Náplava et al. (2022)	A 5 7 R:WO jsem nemohla REQUIRED -NONE- 0
Ours	A 5 7 R:VERB AUX -> AUX VERB jsem nemohla REQUIRED -NONE- 0 (*I have a big family, so I couldn't hope to get anything.*)
German	S Dagegen wieder , bekommen BA Studenten die ein extra Jahr oder mehr studiert haben , leichter Jobs .
Boyd (2018)	A 0 3 R:OTHER Dahingegen REQUIRED -NONE- 0 A 4 5 U:PNOUN REQUIRED -NONE- 0 A 5 6 R:NOUN BA-Studenten REQUIRED -NONE- 0
Ours	A 0 2 R:ADV ADV -> ADV Dahingegen REQUIRED -NONE- 0 A 2 3 U:PUNCT REQUIRED -NONE- 0 A 5 5 M:PUNCT - REQUIRED -NONE- 0 (*On the other hand, BA students who have studied an extra year or more find jobs more easily again.*)

Figure 4: Examples of grammatical error annotation for Czech and German

labels. We applied this approach to German (Boyd, 2018) and Czech (Náplava et al., 2022) to assess whether structured, interpretable annotations could still be produced in the absence of custom heuristics.

As shown in Figure 4, our system improves clarity by attaching POS information to word order and punctuation errors, allowing more consistent cross-lingual comparisons. For Czech, the example highlights differences in word order (WO) annotation: our method distinguishes between auxiliary and main verbs by incorporating POS information, whereas prior work generally treated such cases as generic WO errors. By capturing the syntactic function of the words involved, our method enables more precise and interpretable annotation. A similar improvement is seen for German, where our universal framework avoids language-specific categories while maintaining clear and consistent labeling.

Compared to previous implementations, our annotation outputs remain broadly consistent in terms of overall operation counts, with only minor variations, as shown in Table 2. This suggests that our universal framework based on the MRU scheme can replicate established annotation distributions. Table 3 lists the most frequent error annotations produced by our system alongside those from previous implementations. Our system makes the syntactic categories involved in each replacement edit explicit ($R:P_1 \rightarrow P_2$), reflecting a different annotation choice rather than a direct re-labeling of existing tags.

Future work could explore how to map between these representations to support compatibility and facilitate comparative evaluations. Another direction is to extend this annotation scheme to additional languages: because our framework leverages universal POS tags and dependency labels from stanza, it can be readily applied to the ten other languages in the MultiGEC dataset (Masciolini

et al., 2025a) without additional customization.

	Missing	Replacement	Unnecessary	Total
<i>Czech</i>				
Náplava et al. (2022)	693	3707	515	4915
Ours	695	3672	530	4897
<i>German</i>				
Boyd (2018)	1341	4406	638	6385
Ours	1310	4348	612	6270

Table 2: Comparison of operation counts (Missing, Replacement, Unnecessary) on the development sets of Czech (first 1000 sentences; Náplava et al., 2022) and German (Boyd, 2018).

4.3 Refining language-specific annotations for Korean

Previous research on Korean grammatical error annotation has relied on extensive linguistic resources (Yoon et al., 2023). However, grammatical errors in Korean often manifest at the morpheme level, as observed in L2 writing from the National Institute of Korean Language (NIKL) corpus. In contrast, prior error annotation approaches primarily operate at the word level, which aligns with our methodology. To ensure consistency in annotation, previous work established two priority rules for assigning a single error type to each word because of the potential ambiguity in error classification, particularly when multiple error types could apply to the same token: (i) INSERTION > DELETION > others, and (ii) WS (word segmentation = WB) > WO > SPELL > SHORTEN (incorrect contraction of a word) > PUNCTUATION > OTHERS.

Building on these foundations, we implement language-specific error types based on Algorithm 1 and refine the WB (word boundary) category by introducing two subtypes: WB:M for missing spaces and WB:U for extraneous spaces. The former occurs when spaces are absent between words, causing multiple words to merge into a single unit, which can obscure meaning and hinder readability. The latter arises when superfluous spaces are inserted between or within words, disrupting the

Czech				German			
Náplava et al. (2022)		Ours		Boyd (2018)		Ours	
Annotation	Count	Annotation	Count	Annotation	Count	Annotation	Count
DIACR	989	NOUN → NOUN	760	PUNCT	942	DET → DET	832
OTHER	834	VERB → VERB	465	SPELL	816	NOUN → NOUN	814
PUNCT	487	PUNCT	396	DET:FORM	693	PUNCT	800
SPELL	457	ADJ → ADJ	299	OTHER	670	ADJ → ADJ	466
VERB	271	PRON	161	ORTH	529	VERB → VERB	305
WO	227	DET → DET	101	ADP	348	DET	241
NOUN:INFL	209	ADV → ADV	100	ADJ:FORM	277	ADP → ADP	170
PRON	187	NOUN → ADJ	98	PRON	273	PRON	170
MORPH	177	PUNCT → PUNCT	95	NOUN:FORM	260	PRON → PRON	144
ORTH:CASING	124	ADP → ADP	94	DET	242	AUX → AUX	143

Table 3: Comparison of the top 10 most frequent error annotations on the development sets of Czech (first 1000 sentences; Náplava et al., 2022) and German (Boyd, 2018).

natural flow of the text.

Additionally, we extend grammatical error annotation to functional morphemes, categorizing errors into (i) postposition errors (ADP), (ii) verbal ending errors (PART), and (iii) honorific suffix errors (HON). These errors are further classified into missing (M), unnecessary (U), and incorrect usage (R). Figure 5 illustrates corrections from two annotators: the noun phrase 음식이 *eumsig-i* (‘food.NOM’) is replaced with 음식을 *eumsig-eul* (‘food.ACC’), annotated as R:NOUN → NOUN:ADP, reflecting a case marker correction. Similarly, 먹었습니다 *meogeossseubnida* is replaced with 맞았습니다 *matatseubnida* (‘ate’), which constitutes a spelling error due to phonetic and orthographic similarity.⁴

```

S 비행기 1 음식이 안 3 먹었습니다 .
A 1 2 ||R:NOUN -> NOUN:ADP||음식을||REQUIRED||-NONE-|||0
A 3 4 ||R:Orthographic||먹었습니다||REQUIRED||-NONE-|||0
A 3 4 ||R:VERB -> VERB||맞았습니다||REQUIRED||-NONE-|||1

```

Figure 5: Examples from the Korean M2 file: *I didn’t eat the airplane food* (Annotator 0), and *The airplane food didn’t agree with me* (Annotator 1)

4.4 Integrating deeper customization for Chinese

Chinese grammatical error annotation presents unique challenges due to the lack of explicit word boundaries (Qiu et al., 2025). Previous systems (Zhang et al., 2022; Gu et al., 2025) adopt segmentation schemes based on different linguistic assumptions: for instance, LTP⁵ emphasizes compound words as cohesive lexical units, whereas stanza, trained on the Chinese GSD treebank⁶, adopts a

⁴Annotator 1 annotates a replacement with 맞았습니다 *maj-assseubnida* (‘agree’), altering the meaning of the sentence. This highlights a potential challenge in grammatical error annotation—distinguishing between true errors and alternative valid expressions that change sentence semantics.

⁵<https://github.com/HIT-SCIR/ltp>

⁶https://github.com/UniversalDependencies/UD_Chinese-GSD

finer-grained, morpheme-level segmentation strategy that tends to split compound expressions into smaller units.

These design choices reflect distinct philosophies rather than flaws. However, segmentation differences can affect downstream grammatical error annotation, including both the token spans and the syntactic interpretation of the correction. For example, whether a multi-character expression like 为什么 *wèishéme* (‘why’) is treated as one token or multiple (为什么) influences how missing or replacement errors are classified.

To illustrate the flexibility of our framework, we adopt an LTP-style segmentation approach, which aligns more closely with native speaker intuitions about lexical units in Chinese. While the default stanza pipeline uses GSD-style morpheme-level segmentation, our framework allows researchers to substitute this with alternative schemes, such as LTP’s compound-word-based segmentation. This optional customization demonstrates that language-specific preprocessing decisions, such as tokenization granularity, can be adapted within our framework to better support accurate and interpretable error annotation.

We achieve this integration by re-annotating the Chinese GSD treebank with LTP-informed word boundaries and retraining stanza on this revised corpus. This ensures compatibility with our preferred segmentation standard while preserving the benefits of stanza’s POS tagging and parsing pipeline. As shown in Figure 6, the resulting annotations show more consistent edit spans and error categories, especially in contexts where compound expressions are frequent.

Ultimately, this customization demonstrates the modularity of our framework: rather than enforcing a one-size-fits-all solution, we allow researchers to tailor tokenization to fit linguistic expectations, making the system more robust and adaptable

Chinese GSD-based WB (Gu et al., 2025):
 S ... 解释 为₁₀ 什么 这样 的 情况 ...
 A ...
 A 10 11||R:PROP N -> PRON VERB AUX|||什么 出现 了|||REQUIRED|||-NONE-|||0
 Correction: ... 解释 为 什么 出现 了 这样 的 情况 ...

Our LTP-based WB:
 S ... 解释₇ 为₈ 什么 这样 的 情况 ...
 A ...
 A 7 8||R:ADP -> ADV VERB|||为什么 出现|||REQUIRED|||-NONE-|||0
 A 8 9||R:PROP N -> AUX|||了|||REQUIRED|||-NONE-|||0
 Correction: ... 解释 为 什么 出现 了 这样 的 情况 ...
 ... *jiěshì wèishéme chūxiàn le zhèyàng de qíngkuàng* ...
 (... explain why this kind of situation has occurred ...)

Figure 6: Fragments of grammatical error annotation examples in Chinese with different word boundaries. Incorrect GSD-based segmentation of 为什么 *wèishéme* (‘why’) leads to misleading annotation 什么 出现 了 *shénme chūxiàn le* (‘what has occurred’), while LTP-based segmentation 为什么 出现 *wèishéme chūxiàn* (‘why occurred’) provides an accurate representation.

across languages and segmentation conventions.

5 Conclusion

This work advances grammatical error annotation and evaluation by introducing a standardized, modular framework for multilingual grammatical error typology. Building upon the foundations of errant, we designed a two-tiered system that separates language-agnostic annotation from structured language-specific extensions. This approach supports consistency across typologically diverse languages while allowing targeted customizations when needed.

We reimplemented errant using stanza to provide broader multilingual support, and demonstrated that our system produces accurate and interpretable annotations in English. We then demonstrated how our framework can be applied to other languages with varying levels of customization. For European languages, we showed that our POS- and dependency-based system can generate reliable annotations without requiring language-specific classification modules. For Korean, we applied minor refinements to capture morphologically salient features such as postpositions and spacing errors. Finally, for Chinese, we demonstrated how deeper customization—through the integration of language-specific tokenization and retraining of NLP components—can be incorporated into our framework to support fine-grained, linguistically coherent error annotation.

By balancing consistency and flexibility, our framework enables scalable, interpretable, and reusable grammatical error annotation across languages. This supports more consistent evaluation

and clearer cross-linguistic comparison in multilingual GEC research.

Limitations

While our framework presents a unified and extensible approach to multilingual grammatical error annotation, the implementations described in this paper are primarily intended to demonstrate its adaptability across different languages and levels of customization. A detailed analysis of annotation improvements, including task-specific gains and downstream evaluation effects, is left to future work.

Although we rely on existing NLP tools such as stanza for tokenization and parsing, which offer broad multilingual coverage and consistent annotation schemes, these tools are not explicitly optimized for processing noisy or learner-generated text. This may introduce variability in some edge cases, particularly in languages with complex morphosyntax or ambiguous word segmentation.

References

- 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.
- 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.
- 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.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. **Better Evaluation for Grammatical Error Correction**. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. **Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments**. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.
- Yang Gu, Zihao Huang, Min Zeng, Mengyang Qiu, and Jungyeul Park. 2025. **Improving Automatic Grammatical Error Annotation for Chinese Through Linguistically-Informed Error Typology**. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2781–2798, Abu Dhabi, UAE. Association for Computational Linguistics.
- Charles Hinson, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. **Heterogeneous Recycle Generation for Chinese Grammatical Error Correction**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2191–2201, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Katerina Korre, Marita Chatzipanagiotou, and John Pavlopoulos. 2021. **ELERRANT: Automatic grammatical error type classification for Greek**. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 708–717, Held Online. INCOMA Ltd.
- Katerina Korre and John Pavlopoulos. 2020. **ERRANT: Assessing and improving grammatical error type classification**. In *Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 85–89, Online. International Committee on Computational Linguistics.
- Jenni Marjokorpi. 2023. **The relationship between grammatical understanding and writing skills in finnish secondary 11 education**. *Reading and Writing*, 36(10):2605–2625.
- Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfali, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025a. **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.
- Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfali, Ricardo Muñoz Sánchez, Elena Volodina, Robert Östling, Kais Allkivi, Špela Arhar Holdt, et al. 2025b. **Towards better language representation in natural language processing: A multilingual dataset for text-level grammatical error correction**. *International Journal of Learner Corpus Research*.
- 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.
- 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.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. **A Universal Part-of-Speech Tagset**. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).
- 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, Qingyu Gao, Linxuan Yang, Yang Gu, Tran Minh Nguyen, Zihao Huang, and Jungyeul Park. 2025. **Chinese grammatical error correction: A survey**. *arXiv*.

- 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.
- Ankur Sonawane, Sujeet Kumar Vishwakarma, Bhavana Srivastava, and Anil Kumar Singh. 2020. [Generating Inflectional Errors for Grammatical Error Correction in Hindi](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 165–171, Suzhou, China. Association for Computational Linguistics.
- Catherine G Van Beuningen, Nivja H De Jong, and Folkert Kuiken. 2012. [Evidence on the effectiveness of comprehensive error correction in second language writing](#). *Language Learning*, 62(1):1–41.
- 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.
- 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.
- 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, Italy. ELRA and ICCL.
- Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. [MuCGEC: a Multi-Reference Multi-Source Evaluation Dataset for Chinese Grammatical Error Correction](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. 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.

LLM-based post-editing as reference-free GEC evaluation

Robert Östling
Stockholm University
Sweden
robert@ling.su.se

Murathan Kurfali
RISE Research Institutes of Sweden
Sweden
murathan.kurfali@ri.se

Andrew Caines
ALTA Institute & Computer Laboratory
University of Cambridge, U.K.
andrew.caines@cl.cam.ac.uk

Abstract

Evaluation of Grammatical Error Correction (GEC) systems is becoming increasingly challenging as the quality of such systems increases and traditional automatic metrics fail to adequately capture such nuances as fluency versus minimal edits, alternative valid corrections compared to the ‘ground truth’, and the difference between corrections that are useful in a language learning scenario versus those preferred by native readers. Previous work has suggested using human post-editing of GEC system outputs, but this is very labor-intensive. We investigate the use of Large Language Models (LLMs) as post-editors of English and Swedish texts, and perform a meta-analysis of a range of different evaluation setups using a set of recent GEC systems. We find that for the two languages studied in our work, automatic evaluation based on post-editing agrees well with both human post-editing and direct human rating of GEC systems. Furthermore, we find that a simple n-gram overlap metric is sufficient to measure post-editing distance, and that including human references when prompting the LLMs generally does not improve agreement with human ratings. The resulting evaluation metric is reference-free and requires no language-specific training or additional resources beyond an LLM capable of handling the given language.

1 Introduction

Grammatical Error Correction (GEC) is an important technology for supporting native and non-native writers, and supporting the development of language learners (for a recent survey see, for instance, Bryant et al., 2023). In recent years, neural networks and in particular Large Language Models (LLMs) have led to rapid improvements in the accuracy of such systems, but these developments have

made apparent the difficulty of efficiently evaluating such systems.

For the most part, reference-based metrics have been used for the evaluation of GEC. These metrics depend upon human-created reference corrections and either rely on text similarity measures similar to those used in Machine Translation – examples include GLEU (Napoles et al., 2015) and GREEN (Koyama et al., 2024) – or on comparing the edits made by the GEC system with those by the human; for instance, M^2 (Dahlmeier and Ng, 2012) and ERRANT (Bryant et al., 2017). These reference-based metrics have been shown to correlate less well with human quality estimates than other approaches, in particular with recent neural GEC systems (Kobayashi et al., 2024). In addition, the manual process of creating references is time-consuming.

Reference-free metrics, typically based on neural models, have been proposed as an alternative, but these tend to either be complex and requiring additional (language-specific) training data (Yoshimura et al., 2020; Maeda et al., 2022), or to be simplistic but may correlate relatively poorly with human preferences (Islam and Magnani, 2021).

Östling et al. (2024) proposed using human post-editing to create one reference per GEC system output, and then use a text similarity metric between the system output and its post-edited version as a measure of GEC system quality. This was evaluated on a small number of GEC systems in Swedish, so it is unclear to what extent the resulting scores correlate with human preferences. In addition, human post-editing of every system output is a very time-consuming task. Our goal in this work is to investigate whether the human post-editing step can be performed by an LLM, and how the evaluation setup can be modified to achieve maximal correlation with human evaluation by either

post-editing, ranking, or direct scoring.

Our main research questions are:

- RQ1: does LLM-based post-editing provide a scoring of GEC systems that aligns with human preferences?
(Answer: yes, there is a high level of agreement with different types of human quality assessments.)
- RQ2: how does the choice of text similarity metric affect post-editing based GEC evaluation?
(Answer: Levenshtein distance as used in previous work is sub-optimal, chrF++ is good but overkill; use character bag-of-6-gram overlap instead.)
- RQ3: what difference does it make if human references are provided to the LLMs while performing post-editing?
(Answer: in general the best method is to use only the original sentence + system output, but peculiarities in some datasets affect this outcome.)
- RQ4: how does LLM-based post-editing compare to human post-editing for GEC system evaluation?
(Answer: they generally agree very well, but LLMs make somewhat more changes and have a considerably lower proportion of completely unchanged sentences.)

2 Related Work

Grammatical error correction has a long history as an area of research (Bryant et al., 2023). It has also featured in various shared tasks over the years (e.g., Ng et al., 2014; Bryant et al., 2019; Masciolini et al., 2025). Since statistical approaches to GEC were widely adopted, the best-performing systems involve supervised models trained on annotated corpora: usually involving sequence-to-sequence models (e.g., Rothe et al., 2021) or pipeline systems based on sequence tagging (e.g., Omelianchuk et al., 2020).

According to recent research, LLMs do not outperform these supervised GEC systems on every benchmark, at least for English (Loem et al., 2023; Davis et al., 2024). Instead, it has been shown that they can potentially improve the recall of GEC models in an ensemble setting (Omelianchuk et al., 2024). Moreover, given the increasing use of LLMs

as judges, in this work we investigate to what extent LLMs can be used for GEC evaluation which, along with the availability of high quality annotated data, is a bottleneck to progress in GEC (Kobayashi et al., 2024).

Current metrics are either reference-based or reference-free, meaning that they do or do not, respectively, depend upon ‘ground truth’ corrections. The most widely-used reference-based metrics are precision, recall and $F_{0.5}$ – most often obtained from the M^2 scorer (Dahlmeier and Ng, 2012) or with ERRANT (Bryant et al., 2017) – along with GLEU, derived from the BLEU score commonly used in machine translation (Napoles et al., 2015). However, there is often more than one possible way to correct a grammatical error, and even with multiple annotations it is difficult to cover all possibilities in reference-based approaches.

Examples of reference-free metrics include the Scribendi Score (Islam and Magnani, 2021) and IMPARA (Maeda et al., 2022). The former may involve any LLM, in principle, whilst the latter was implemented using BERT (Devlin et al., 2019). However, the reliance on language models for reference-free metrics means that they tend to be biased towards fluency corrections over minimal edits which stay closer to the original text formulation but may not be recognized as improvements by the language models. Fluent corrections are usually preferable from a readability and naturalness perspective, but it is arguable from a pedagogical standpoint that it is better to in fact offer minimal edits as feedback to human learners rather than error avoidance strategies (Sakaguchi et al., 2016; Caines et al., 2023; Mita et al., 2024).

Nevertheless, the reliance on ground truth references remains a limiting factor in evaluation of GEC systems on new data. If it can be shown that LLMs can be reliably put to use as GEC post-editors, for the purpose of evaluation, correlating well with human judgements, it would release the pressure on the GEC bottleneck somewhat. Östling et al. (2024) examine the feasibility of post-editing based evaluation with Swedish GEC data and perform direct scoring as well as post-editing of the outputs of three different GEC systems and two fluency-edited references. They find that post-editing distance correlates strongly with the scores assigned by the annotator, but the small sample of GEC systems limits the range of conclusions that they are able to draw. Additionally, their annotation procedure is fully manual and would be difficult to

scale up.

3 Data

Kobayashi et al. (2024) performed a meta-evaluation of 12 recent English GEC systems, and published the SEEDA dataset of GEC system outputs and human rankings of sentences from these outputs. We use this dataset because it contains a sufficient number of GEC systems to compute reasonably reliable correlations between a given GEC evaluation metric and the human assessments. In addition to system outputs of 12 modern GEC systems, it also includes the original uncorrected sentences (INPUT) and two human-created references, one with minimal edits (REF-M) and one edited for fluency (REF-F).

Östling et al. (2024) published human annotations with post-edited versions of 3 Swedish GEC systems as well as the original uncorrected sentences (INPUT) and three human-created references, one with minimal edits (REF-M) and two edited for fluency. The GEC system outputs and the fluency-edited references are annotated with scores for grammaticality, fluency and meaning preservation, and post-edits to achieve perfect scores in these three assessment dimensions. We include the Swedish data for two main purposes: to allow direct comparisons between human and LLM post-edits, and to verify that the proposed method can be applied to languages other than English given a suitable LLM.

4 Method

We have several different recent LLMs perform post-editing of GEC system outputs from the datasets of Kobayashi et al. (2024) in English, and Östling et al. (2024) in Swedish.¹ We use Gemma 2 in several sizes (2 billion parameters, 9B, 27B) (Gemma Team et al., 2024), Gemma 3 27B (Gemma Team et al., 2025), Llama 3.1 8B (Grattafiori et al., 2024), Mistral Small 24B (Jiang et al., 2023), Qwen 2.5 32B (Bai et al., 2023), and Command A-111B (Cohere, 2025).

For each LLM, we try each combination of the following two parameters:

- Semantic grounding. In order to ensure that the post-editing does not diverge from the semantics of the original text, we include four

(English) or three (Swedish) types of semantic grounding. In all cases the GEC system output is provided in the prompt.

- None. Only the system output is provided in the prompt.
 - INPUT. The GEC system input (original text) is included.
 - REF-M. A human minimal edits reference is included.
 - REF-F. A human fluency edited reference is included (English data only).
- Similarity metric. Following Östling et al. (2024) we use Normalized Levenshtein distance as one metric, and add two n-gram-similarity-based metrics.

- Normalized Levenshtein Similarity, which is identical to Normalized Levenshtein Distance apart from the direction (higher is better):

$$S(a, b) = 1 - L(a, b) / \max(|a|, |b|)$$

- chrF++ (Popović, 2017), which in our setting computes the mean F_2 score over word bigram and character 6-gram precision and recall. Unlike the other metrics, this is asymmetric and we treat the post-edited text as the reference.
- Character 6-gram bag-of-n-grams overlap, a symmetric measure of similarity:

$$S(a, b) = |N_a^6 \cap N_b^6| / |N_a^6 \cup N_b^6|$$

where N_s^6 is the set of character 6-grams (including spaces) for string s .

Because it is difficult to justify a full parameter search using the very largest model (GPT-4o²), we obtain post-edits only for the setting where the original sentence is used as semantic grounding (INPUT), since this was the most promising configuration in preliminary experiments. For the Swedish part we also restrict the set of LLMs used to some of the models that obtained the most promising results on the English data, due to time and data licensing constraints.³

²The actual number of parameters has not been published for it or its smaller version GPT-4o-mini, but we see a limited value in exploring the full set of parameters for these models.

³The current license of the Swedish data does not permit the use of OpenAI API.

¹Prompts are given in Appendix A.

Post-editor	r	ρ
Gemma 2-2B	0.69	0.77
Gemma 2-9B	0.95	0.92
Gemma 2-27B	0.79	0.56
Gemma 3-27B	0.82	0.67
Llama 3.1-8B	0.90	0.83
Mistral Small 24B	0.95	0.91
Qwen 2.5-32B	0.81	0.68
Command A-111B	0.95	0.89

Table 1: System-level correlations between post-edit distance and human ratings, averaged over all similarity metrics, semantic grounding options, and human ratings. Here and below boldface is used as a visual aid to identify the highest values.

Similarity metric	r	ρ
Levenshtein	0.74	0.63
6-gram overlap	0.92	0.85
chrF++	0.91	0.85

Table 2: Mean system-level correlations between post-edit distance and human ratings, averaged over all LLM post-editors, semantic grounding options, and human ratings.

For English, we follow Kobayashi et al. (2024) and compute correlations (Pearson r and Spearman ρ) to human annotations on the system level.⁴ These are derived in two different types of annotation (edit-based or sentence-based comparisons), using two different methods (TrueSkill and Expected Wins) of summarizing the rankings into numeric scores, resulting in four different system-level references. To avoid making arbitrary decisions on which of these to prefer, and to increase the reliability of the results, we consistently use means over all of these four except in Table 4 where we investigate the effect of the human system-level score type and find that it is relatively small. For the sentence level evaluations we use Kendall τ , as computed by the software published by Kobayashi et al. (2024), for comparing to human sentence-level rankings.

For the Swedish data the available annotations are different, compared to English. Instead of rankings of system outputs, each system output has been annotated for grammaticality, fluency and meaning preservation. If any of these are annotated with less than a perfect score (4 on a scale

⁴Sentence 22 of the REF-M file in the SEEDA dataset is empty. We handle this by arbitrarily giving this sentence a score of 0 for all similarity metrics.

Semantic grounding	r	ρ
None	0.83	0.66
INPUT	0.88	0.87
REF-M	0.83	0.76
REF-F	0.88	0.82

Table 3: Mean system-level correlations between post-edit distance and human ratings, averaged over all LLM post-editors, similarity metrics, and human ratings.

Human rating	r	ρ
EW/edit	0.84	0.77
EW/sentence	0.85	0.79
TS/edit	0.87	0.77
TS/sentence	0.87	0.79

Table 4: Mean system-level correlations between post-edit distance and human ratings, averaged over all LLM post-editors, similarity metrics, and semantic grounding options. The four human rating references are computed using Expected Wins (EW) or TrueSkill (TS) from sentence-level rankings that are either edit-based or sentence-based.

1–4), there is also a post-edited version of the system output with the goal of performing minimal editing to achieve full scores on all three properties. Since there are only three GEC system outputs and two human references included in the data, we do not consider it meaningful to perform a system-level evaluation as in the English data. Instead, we use Spearman’s ρ to compare the post-edit score between the human annotator and each LLM. We also compare the LLM post-edit scores to the mean of the human annotator’s grammaticality, fluency and meaning preservation scores, which we use as a general measure of the quality of that particular correction.

5 Results and Discussion

5.1 Overall agreement with human rankings

In order to see whether LLM-based post-editing provides a scoring of GEC systems that aligns with human preferences (RQ1), we begin by applying the meta-evaluation framework of Kobayashi et al. (2024). Because our proposed evaluation setup has several hyperparameters and only 15 system outputs⁵ to measure correlations with, we search

⁵Whenever the semantic grounding uses one of the human references (REF-M or REF-F), that reference is excluded from computing the correlation and only the remaining 14 system outputs are used. Note that unless stated otherwise, we use the term “system output” to also include the human-created

LLM	Spearman ρ					Pearson r				
	Base	None	INPUT	REF-M	REF-F	Base	None	INPUT	REF-M	REF-F
Gemma 2-2B	-0.28	0.94	0.61	0.74	0.93	-0.64	0.97	0.58	0.72	0.96
Gemma 2-9B	0.35	0.91	0.95	0.94	0.94	-0.18	0.96	0.97	0.97	0.96
Gemma 2-27B	0.55	0.68	0.92	0.83	0.60	0.05	0.87	0.97	0.90	0.83
Gemma 3-27B	0.46	0.42	0.94	0.83	0.89	-0.15	0.76	0.97	0.91	0.93
Llama 3.1-8B	0.14	0.95	0.93	0.92	0.92	-0.41	0.97	0.95	0.98	0.98
Mistral Small 24B	0.29	0.91	0.96	0.94	0.95	-0.27	0.97	0.98	0.98	0.95
Qwen 2.5-32B	0.56	0.44	0.94	0.89	0.83	-0.00	0.75	0.96	0.89	0.92
Command A-111B	0.48	0.87	0.95	0.93	0.93	-0.06	0.95	0.98	0.98	0.94
GPT-4o	–	–	0.96	–	–	–	–	0.98	–	–
GPT-4o-mini	–	–	0.96	–	–	–	–	0.97	–	–

Table 5: Mean system-level correlations between post-edit distance and human ratings, per LLM and semantic grounding option, always using 6-gram overlap and averaging over human ratings.

LLM	Sentence-based					Edit-based				
	Base	None	INPUT	REF-M	REF-F	Base	None	INPUT	REF-M	REF-F
Gemma 2-2B	-0.21	0.32	0.18	0.23	0.32	-0.13	0.35	0.21	0.27	0.33
Gemma 2-9B	0.10	0.36	0.54	0.41	0.38	0.15	0.35	0.52	0.41	0.39
Gemma 2-27B	0.21	0.18	0.42	0.25	0.15	0.26	0.18	0.41	0.25	0.20
Gemma 3-27B	0.11	0.14	0.47	0.28	0.29	0.20	0.11	0.45	0.27	0.24
Llama 3.1-8B	-0.01	0.33	0.36	0.30	0.31	0.06	0.35	0.39	0.34	0.34
Mistral Small 24B	0.08	0.35	0.48	0.38	0.39	0.18	0.33	0.50	0.38	0.33
Qwen 2.5-32B	0.21	0.14	0.45	0.23	0.26	0.24	0.16	0.45	0.22	0.23
Command A-111B	0.19	0.38	0.46	0.37	0.38	0.22	0.37	0.47	0.37	0.39
GPT-4o	–	–	0.54	–	–	–	–	0.55	–	–
GPT-4o-mini	–	–	0.46	–	–	–	–	0.46	–	–

Table 6: Mean sentence-level Kendall τ between post-edit distance and human ratings, per LLM and semantic grounding option, always using 6-gram overlap.

through each parameter independently taking the averages over all other parameters in order to avoid overfitting. Averaged system-level correlations are presented in Table 1 (per LLM), Table 2 (per similarity metric), and Table 3 (per semantic grounding option). Additionally, we also present the averaged correlations per human rating setup (Table 4) and see that these are in general agreement with each other. In all other system-level evaluation results, we present averages over all four human rating setups to obtain more reliable estimates.

5.2 Effect of text similarity metric

Next, we turn to the question of how the text similarity metric used to compare the system output with its post-edited version affects the results (RQ2). Östling et al. (2024) used Normalized Levenshtein Distance with manual post-edits. We compute the its negated version (Normalized Levenshtein Similarity) along with two other options. The results are shown in Table 2, averaged over all other parameters. It is clear that Normalized Levenshtein Similarity is in fact sub-optimal, and that both of the other two metrics obtain correlations with human ratings that are considerably higher. In the following analysis we use 6-gram overlap, as it is simple and efficient to compute.

5.3 Effect of semantic grounding

To investigate whether the type of semantic grounding affects post-editing based evaluation (RQ3), we compute the correlations separately for the different types of semantic grounding (Table 5). There are pronounced differences between the various LLMs with respect to which type works best, but the overall trend is that adding human-written references typically does not improve the outcomes, and in most cases results in lower correlation with human ratings.

We have included a baseline (Base) consisting of a reference generated by the same LLM *without* access to the system output, using the LLM as a GEC system with access to the original text only.⁶ This is done to exclude the possibility that the LLMs generate high-quality references and that post-editing is an unnecessary complication. However, the low correlation values for the baseline indicate that including the system output and performing post-editing is essential to the success of

references.

⁶Prompts are given in Appendix A.

LLM	S.G.	HP	HS
Gemma 2-9B	None	0.39	0.39
Gemma 2-9B	INPUT	0.28	0.26
Gemma 2-9B	REF-M	0.55	0.51
Mistral Small 24B	None	0.40	0.38
Mistral Small 24B	INPUT	0.30	0.30
Mistral Small 24B	REF-M	0.55	0.51
Qwen 2.5-32B	None	0.35	0.34
Qwen 2.5-32B	INPUT	0.34	0.34
Qwen 2.5-32B	REF-M	0.49	0.45
Command A-111B	None	0.49	0.46
Command A-111B	INPUT	0.40	0.39
Command A-111B	REF-M	0.58	0.53

Table 7: Spearman ρ between LLM post-edit score, and each of human post-edit (HP) and human score (HS, mean of grammaticality, fluency and meaning preservation scores). Scores from post-edits are defined as the 6-gram similarity to their respective system output. The correlation between HP and HS is 0.81. S.G. = semantic grounding.

our method. Manual inspection indicates that REF-F and GPT-3.5, both of which contain a considerable amount of fluency edits, are generally rated poorly by the baseline.

It is also noteworthy that some of the highest system-level correlations are obtained by letting the smallest of the evaluated LLMs (Gemma 2-2B) post-edit the system output with only the system output and no semantic grounding, thus ignoring any possible semantic errors. In line with previous work (Yoshimura et al., 2020) which found that meaning preservation is not an important factor when trying to achieve high correlation to human ratings, this indicates that having even a modest-sized LLM perform conservative correction of the system output brings us to close agreement with human system-level ratings.

5.4 Sentence-level evaluation

We now turn from system-level to sentence-level evaluations. Following Kobayashi et al. (2024), we present the sentence-level agreement with human rating as Kendall τ values in Table 6. At this finer level of granularity, the differences between different metric parameters become apparent. Adding the original sentence as semantic grounding consistently improves the correlation with human assessments, while adding a human reference (REF-M or REF-F) shows no such tendency. Again, the baseline consistently has very low correlations.

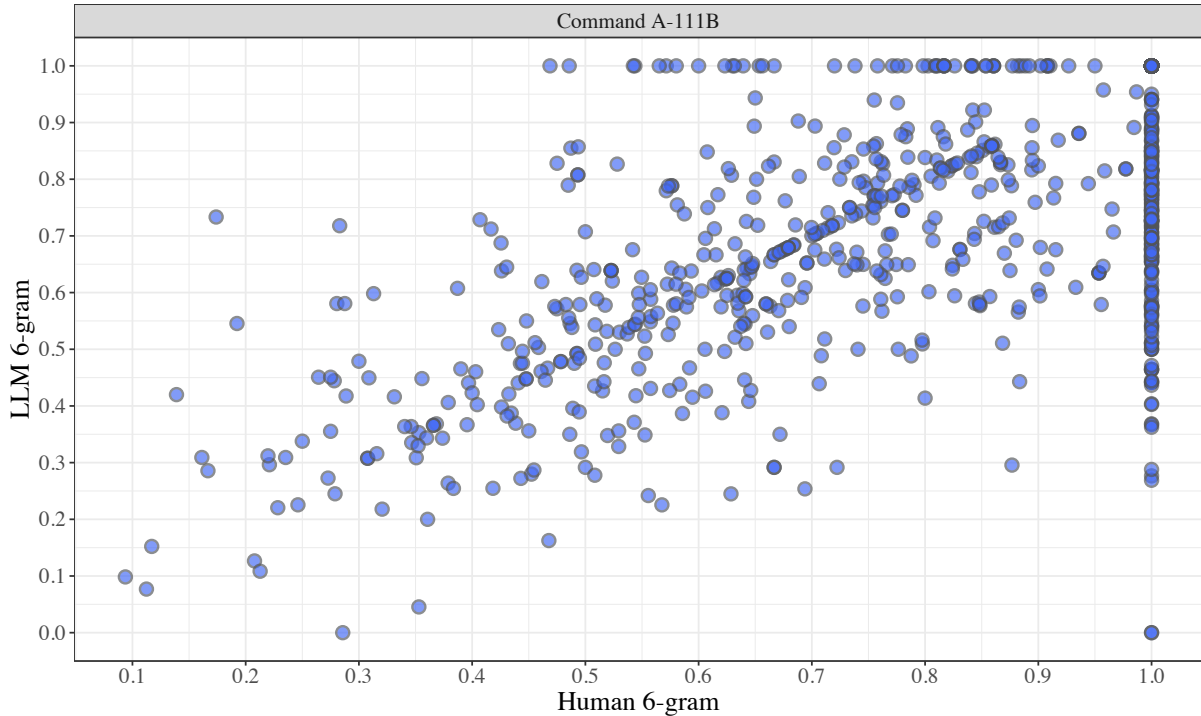


Figure 1: Scatter plot of character 6-gram overlap scores derived from human (x-axis) and LLM (y-axis) post-edits, in both cases using REF-M for semantic grounding. A score of 1 indicates that no changes were made during post-editing. The points are one-third transparent to avoid over-plotting.

5.5 Human vs. LLM post-edits

In order to investigate the relationship between post-edits made by humans and LLMs (RQ4), we use the Swedish data from Östling et al. (2024), where three GEC system outputs and two human references have been post-edited as well as rated for grammaticality, fluency and meaning preservation. We used a subset of the most promising LLMs to replicate the post-editing and allow direct comparisons between LLM and human post-edits. Table 7 presents correlations between LLM post-edit scores (using character 6-gram overlap) and human post-edit scores (also using character 6-gram overlap) as well as to the mean of the grammaticality, fluency and meaning preservation scores. The latter is used to approximate a direct assessment by the human annotator of the GEC system’s output of that particular sentence.

In the human post-editing of Östling et al. (2024), a minimal edits reference (REF-M) was used for semantic grounding. As expected, we find that using this reference in the LLM prompt leads to higher correlation to both the human post-edit distance and the human annotated scores. Unlike for the English SEEDA data, using only the original sentence for semantic grounding (INPUT) leads to consid-

erably lower correlations. We believe this to be due to the fact that the Swedish data consists of individual sentences in random order, and that only the creator of the REF-M reference has access to a wider context, while both the human and LLM post-editors lack any such context.

Figure 1 shows the 6-gram overlap scores assigned to each sentence from both the human post-editing and LLM post-editing. The LLM used was the one with the highest correlation to human post-editing scores (Cohere Command A-111B). We see that there is generally high agreement, as the $\rho = 0.58$ correlation indicates, but that there are some clear differences. The human post-editor frequently (46%) leaves the sentence unchanged, whereas the LLM does this less often (27%). The same tendency of the human post-editor being more reluctant to change is reflected in the mean overlap scores: 0.81 (SD 0.22) for the human, compared to 0.74 (SD 0.22) for the LLM, meaning that on the whole the human annotators post-edited less of the system output than the LLMs did. A significant part of this difference is due to the cases where humans leave sentences unchanged, which is demonstrated by considering only sentences where both the human and the LLM actually perform

some edits. In this case, the correlation between the 6-gram overlap scores increases to $\rho = 0.67$ for the same model.

5.6 LLMs as GEC systems and post-editors

An important question⁷ is whether LLMs can be expected to post-edit the output of LLM-based systems, and if it would not be better to simply use the LLMs as GEC systems to begin with.

Our method is based on the assumption that an LLM is capable enough to post-edit the output of even the best GEC systems under evaluation. We have found this to be the case in our evaluation where even the best LLM-based systems undergo significant post-editing during evaluation. Furthermore, we argue that the availability of an LLM with sufficiently high capability is a realistic assumption in a practical setting, since considerably more computation can be spent on GEC evaluation (which will be run once or a few times) than on actual deployed GEC systems.

It is also important to note that GEC evaluations will also be needed for non-LLM based systems. Kobayashi et al. (2024) worked with 12 systems to carry out English GEC for the SEEDA dataset. Östling et al. (2024) worked with 3 systems for Swedish GEC of essays in the SweLL dataset (Volodina et al., 2019). In both cases the systems include both supervised and unsupervised approaches, for instance involving machine translation, sequence tagging and few-shot prompting of LLMs. That is, we do evaluate both non-LLM and LLM systems for GEC in this work.

6 Conclusions

We find that LLMs can be used as very effective evaluation tools for GEC systems, by asking them to post-edit system outputs and using a simple string similarity metric (character 6-gram overlap) to measure the amount of editing needed to go from the GEC system’s output to a version considered by the LLM to be fully grammatical and fluent, while completely preserving the meaning expressed in the original. Even relatively small LLMs (such as Gemma 2-2B) can perform this task well enough to achieve nearly perfect correlation with human ratings at the system level. However, the picture is different when the GEC system output is assessed on the level of individual sentences, with considerable variation between LLMs in the ability to

predict the human assessment of that sentence.

While we use the most recent publicly available GEC meta-evaluation dataset (Kobayashi et al., 2024), LLM-based GEC systems improve rapidly and an important question is to what extent LLM-based post editing is able to evaluate the output of the most capable LLMs. Answering this would require additional annotations that go beyond the scope of this work.

To summarize, we see several advantages of evaluation based on post-editing GEC system outputs by LLMs:

- High correlations with human direct assessment of GEC system quality, both at the system level and sentence (or document) level.
- Analyzing the post-edits provides an interpretable indication of the weaknesses of a particular GEC system, and this can be partly automated by tools such as ERRANT (Bryant et al., 2017). This contrasts with ranking-based evaluations like that recently proposed by Goto et al. (2025).
- Given a multilingual LLM, post-editing can handle multiple languages without requiring any additional language-specific resources or training.
- Unlike metrics that depend on having a large number of data points to average over (e.g., Islam and Magnani, 2021; Goto et al., 2025), post-editing distance can be estimated even on a single document without a sentence-aligned system output. It is thus suitable for document-level evaluations, as in Masciolini et al. (2025).

Fully exploring document-level multilingual evaluation would be an interesting direction of future work (Piotrowska, 2025). Note that in this work we have only worked at the sentence level, as has been conventional in GEC for the most part. However, in recent years there has been growing interest in document-level GEC, as well as evidence that the additional context can aid system performance on certain error types which relate to linguistic features above the sentence level (Yuan and Bryant, 2021; Mita et al., 2024; Masciolini et al., 2025).

⁷Raised by one of the anonymous reviewers.

Limitations

This paper on GEC is limited in the sense that we work with only 2 languages (English and Swedish) and findings for other languages may vary from those reported here. Annotated data for GEC are costly to build and therefore hard to come by: the datasets we work with in this paper are relatively small, compared to some corpora used in other areas of NLP. In addition the correction of grammatical errors is to some extent subjective, and an estimation without full access to the authors' original intentions. However, this limitation is a factor for all working on GEC.

LLMs have proven to be highly effective for a number of NLP tasks. In this paper we show that they are not necessarily state-of-the-art at the GEC task itself, but may be sufficiently accurate on the GEC post-editing task. This finding is limited by the continued availability of high quality open-weights LLMs, sufficient computing resources for those conducting research to be able to use the LLMs for inference, and the fact that we have only evaluated their performance on two languages. However, in principle, many LLMs have highly multilingual capabilities, and we expect that the outcomes reported here will hold for many other languages.

Acknowledgments

We thank the three anonymous reviewers for their valuable comments.

This work is partly funded by the Swedish national research infrastructure Språkbanken, jointly financially supported by the Swedish Research Council (2018–2028; grants 2017-00626 and 2023-00161) and the 10 participating partner institutions. The second author is partially supported by the Swedish Research Council under grant agreement no. 2024-01506. The third author is supported by Cambridge University Press & Assessment.

References

- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. [Qwen technical report](#). *Preprint*, arXiv:2309.16609.
- 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.
- Andrew Caines, Luca Benedetto, Shiva Taslimipour, 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*.
- Cohere. 2025. [Command A: An enterprise-ready large language model](#).
- 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.
- 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.
- 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.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025. [Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human?](#) *Preprint*, arXiv:2502.09416.
- 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.
- Md Asadul Islam and Enrico Magnani. 2021. [Is this the end of the gold standard? a straightforward referenceless 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.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7B](#). *Preprint*, arXiv:2310.06825.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- Robert Östling, Katarina Gillholm, Murathan Kurfalı, 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.

Emilia Piotrowska. 2025. Multilingual document-level gec evaluation. Bachelor’s thesis, Department of Linguistics, Stockholm University.

Maja Popović. 2017. [chrF++: words helping character n-grams](#). In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. 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.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. [Reassessing the goals of grammatical error correction: Fluency instead of grammaticality](#). *Transactions of the Association for Computational Linguistics*, 4:169–182.

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 (NEJLT)*, 6:67–104.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. [SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zheng Yuan and Christopher Bryant. 2021. [Document-level grammatical error correction](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84, Online. Association for Computational Linguistics.

A Prompts

In this appendix, we present the prompts used across different experiments. We use a total of three prompt templates: one for the baseline results where LLMs are applied to GEC tasks, and two for the post-editing experiments—one without a semantic grounding sentence and one with. The same prompt structure is used for all three types of semantic grounding.

Prompt for GEC baseline

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.

Instructions:

1. Return **ONLY** the corrected sentence.
2. Wrap the corrected sentence in `<corrected>` and `</corrected>` tags.
3. Do **NOT** include any explanations, extra text, or formatting.

Example:

```
<corrected>This is your corrected sentence.</corrected>
```

Input sentence: {sentence}

Output:

Prompt for post-editing without semantic grounding

Please make minimal modifications to the given sentence to achieve all of the properties below:

- Perfect grammaticality: The sentence is native-sounding. It has no grammatical errors, but may contain very minor typographical and/or collocation errors.
- Perfect fluency: The sentence sounds extremely natural and native-like.
- Same language: The sentence must remain in the same language as the original (do not translate or change language).

Instructions:

1. Return **ONLY** the corrected sentence.
2. Wrap the corrected sentence in `<corrected>` and `</corrected>` tags.
3. If the original sentence is already perfect, return it **AS IS** inside the `<corrected>` tags.
4. Do **NOT** include any explanations, extra text, or formatting.

Example output format:

```
<corrected>Your corrected sentence here.</corrected>
```

Sentence: {sentence}

Output:

Prompt for post-editing with semantic grounding

Please make minimal modifications to the given sentence to achieve all of the properties below:

- Perfect grammaticality: The sentence is native-sounding. It has no grammatical errors, but may contain very minor typographical and/or collocation errors.
- Perfect fluency: The sentence sounds extremely natural and native-like.

Instructions:

1. Return **ONLY** the corrected sentence.

2. Wrap the corrected sentence in `<corrected>` and `</corrected>` tags.
3. Ensure that the corrected sentence preserves the meaning of the reference sentence provided below. The reference may contain grammatical errors — it is for semantic grounding only.
4. If the original sentence is already perfect, return it AS IS inside the `<corrected>` tags.
5. Do NOT include any explanations, extra text, or formatting.

Example output format:

```
<corrected>Your corrected sentence  
here.</corrected>
```

Sentence: {sentence}

Reference (for meaning preservation only): {reference sentence}

Output:

Increasing the Generalizability of Similarity-Based Essay Scoring Through Cross-Prompt Training

Marie Bexte¹ and Yuning Ding¹ and Andrea Horbach^{1,2,3}

¹CATALPA, FernUniversität in Hagen, Germany

²IPN - Leibniz Institute for Science and Mathematics Education, Kiel, Germany

³University of Kiel, Germany

Abstract

In this paper, we address generic essay scoring, i.e., the use of training data from one writing task to score data from a different task. We approach this by generalizing a similarity-based essay scoring method (Xie et al., 2022) to learning from texts that are written in response to a mixture of different prompts. In our experiments, we compare within-prompt and cross-prompt performance on two large datasets (ASAP and PERSUADE). We combine different amounts of prompts in the training data and show that our generalized method substantially improves cross-prompt performance, especially when an increasing number of prompts is used to form the training data. In the most extreme case, this leads to more than double the performance, increasing QWK from .26 to .55.

1 Introduction

In automated scoring, one desideratum is often to train a generic classifier that does not rely on the availability of training material for a certain writing task, i.e., prompt, but can transfer from training material for one or several prompts to data from new writing tasks.

This holds both for content scoring, also known as short-answer scoring, and essay scoring. In content scoring, texts of up to a few sentences in length are scored for conceptual correctness. Essay scoring deals with scoring longer texts that are rated both on content and language use.

Generic scoring has a high practical relevance in the classroom, as teachers often do not have the resources to annotate training data for each new prompt. However, the generalizability of classifiers is often low (see, e.g., Phandi et al. (2015)). Especially in a hard domain transfer scenario when classifiers are trained on a single or a few prompts only, they might pick up on lexical material specific to that particular writing task.

For instance, as shown on the left side of Figure 1, two essays from the prompt ‘The Face on Mars’ in the PERSUADE dataset may lead a scoring classifier trained solely on this prompt to treat words such as ‘aliens’ and ‘Mars’ as significant features. These words, however, are not found in essays from other prompts, such as the two essays from the ‘Facial Action Coding System’ prompt shown on the right side. Despite the differences in content, essays from different prompts with the same score share general similarities. For instance, low-scoring essays from different prompts (top part of Figure 1) often share weaknesses such as limited vocabulary, repetition of phrases, and overuse of simple words. In contrast, high-scoring essays (bottom part of Figure 1) display features that contribute to higher scores, such as a logical progression with the underlined transitional phrases. These lexical patterns should be prioritized when training a generic scoring model, as they contribute significantly to the overall quality of an essay, regardless of the specific prompt. However, it should be noted that we are not claiming that these elements are the *only* relevant aspects in scoring the data, but rather that they are important *enough* to make them exploitable for cross-prompt scoring.

While generic scoring has been more extensively explored for some content scoring datasets (Bailey and Meurers, 2008; Mohler and Mihalcea, 2009; Meurers et al., 2011; Dzikovska et al., 2013), cross-prompt approaches to essay scoring have only received more interest in recent years (Phandi et al., 2015; Jin et al., 2018; Li et al., 2020; Chen and Li, 2023).

In our study, we approach generic essay scoring by training classifiers that are discouraged from paying attention to prompt-specific material in the essays. In both flavors of educational free-text scoring, content and essay scoring, similarity-based scoring has recently emerged as a viable alternative to the default of instance-based scoring (Bexte

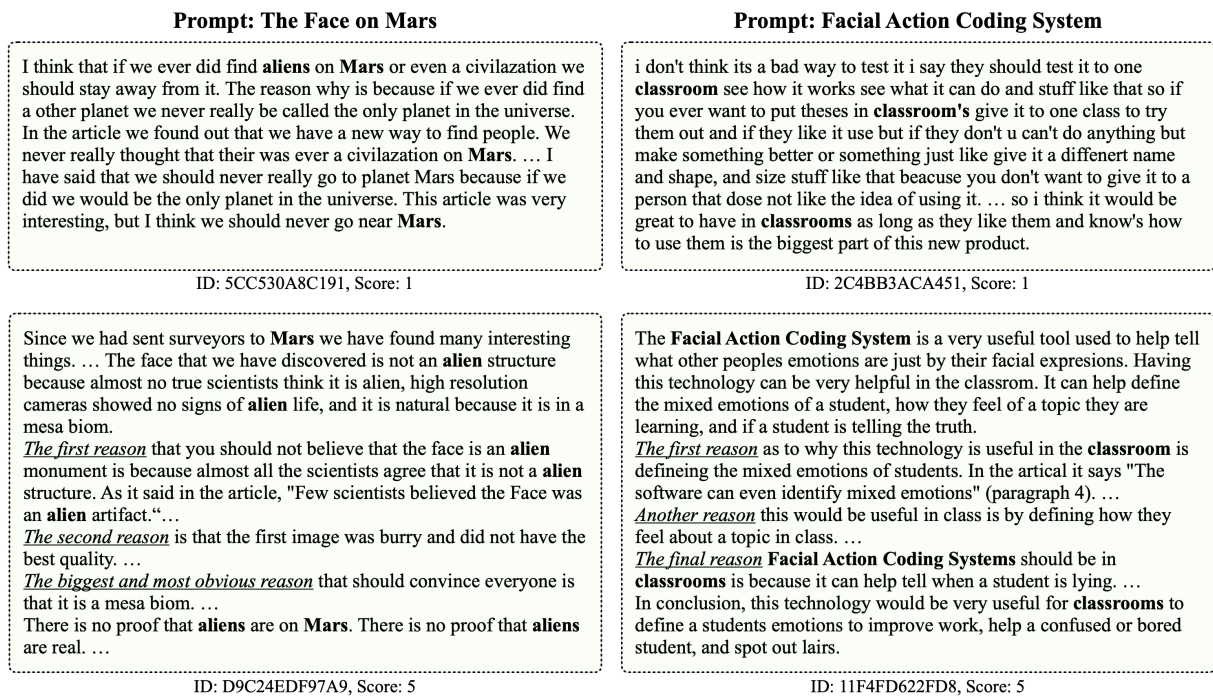


Figure 1: Example essays taken from two different prompts in the PERSUADE dataset that share the same low (top) or high (bottom) score. Words in bold are prompt-specific, which may be picked up by a classifier trained on a single prompt. The underlined transitional phrases show an example of lexical patterns that contribute to higher scores, which can be used when training a generic scoring model.¹

et al., 2022; Xie et al., 2022). While instance-based scoring learns the association between individual learner texts and their scores, the input in similarity-based scoring are pairs of texts. In such a pair, an essay of interest is compared to a reference essay with a known score.

We adapt the similarity-based essay scoring approach of Xie et al. (2022), which exhibits state-of-the-art performance on the commonly used ASAP essay scoring dataset. While Xie et al. (2022) only demonstrated good within-prompt performance, we augment their approach for cross-prompt scoring. Our crucial step in avoiding overfitting to prompt-specific information is to only use pairs of learner essays that answer different writing prompts during training. In doing this, we force the similarity metric to pay attention to structural rather than purely lexical similarity between texts.

We hypothesize that the problem of prompt-specific similarity metrics is more severe in cases where training material only covers a single or a few prompts, as paying attention to a prompt-specific feature makes an impact on a larger portion of the dataset in these cases. To test this assumption, we vary the number of prompts that is mixed in the training data in our experiments.

Overall, our paper makes the following contributions:

- We extend the method of Xie et al. (2022) to facilitate cross-prompt scoring.
- We compare two strategies to pair up training data in similarity-based cross-prompt scoring.
- We demonstrate the benefits of our strategy for increasing cross-prompt performance on two publicly available datasets (PERSUADE and ASAP), finding that the benefits of our method increase when an increasing number of prompts is mixed in the training data.

Our code and data split is available on GitHub².

2 Related Work

For many years, the main interest in automated essay scoring has been in prompt-specific classifiers, where one specific model was trained for each new

²<https://github.com/mariebexte/generalizing-similarity>

¹In this example, we use the first two prompts from the dataset, which happen to include the words ‘face’ and ‘facial’. While these shared terms might influence a general classifier trained specifically on these prompts, this is merely a coincidence and not the intended focus of our analysis.

writing prompt (e.g., Taghipour and Ng (2016); Dong et al. (2017); Dasgupta et al. (2018); Uto et al. (2020)). This focus has shifted to generic or cross-prompt scoring, where a classifier is trained on one or more prompts. The classifier is then applied to essays that answer prompts which were not seen during training.

2.1 Cross-Prompt Essay Scoring

The problem of cross-prompt essay scoring has been approached in various ways. Phandi et al. (2015) use Bayesian Linear Ridge Regression to score essays using features selected to be predictive of either the source or the target domain. Jin et al. (2018) propose a two-stage neural network (TDNN) approach, in which they use a generic model to automatically create pseudo-training data for the target domain. Li et al. (2020) also propose a two-stage method that aims to extract the shared knowledge between the source and target domain, first creating pseudo-training data, which is then used in a Siamese network. The PMAES system (Chen and Li, 2023) uses a prompt-mapping contrastive learning method to learn more consistent representations of source and target prompts. By doing this, unlabeled data from the target prompt is used to adapt the model. Thus, adaptation to future target prompts would require additional training. Similarly, Zhang et al. (2025) and Wang et al. (2025) also include information derived from unlabeled target data in their training.

2.2 Similarity-Based Essay Scoring

Orthogonal to cross-prompt scoring, recent years have also seen more and more approaches that rely on the similarity between text pairs for scoring instead of training a classifier on features extracted from individual texts (see also Horbach and Zesch (2019)).

The purported advantage that similarity-based approaches might work better in a cross-domain scenario has been refuted, at least for content scoring (Bexte et al., 2023). However, little work so far has explored the potential of cross-prompt similarity-based essay scoring.

3 Method

In a similarity-based scoring setup, the predicted score is derived from a comparison with reference essays. We follow the prompt-specific approach of Xie et al. (2022), which essentially predicts how

much better or worse than a reference essay an essay of interest is. Figure 2 shows an overview of the network structure of this approach. In practice, training essays are used as reference essays, i.e., training is performed on pairs of training essays, and at inference, validation or test essays are compared to training essays. While Xie et al. (2022) use a BERT (Devlin et al., 2019) model at the core of their model, we use a Longformer (Beltagy et al., 2020) instead. This is done to accommodate the longer text length typically encountered in essay scoring. We use the *longformer_base_4096* model as provided on Hugging Face³. Both the answer of interest and a reference answer are embedded using the same Longformer model. The difference between the two embeddings is subsequently fed into a linear layer, which performs a regression. The aim is to predict the difference in the score of the essay of interest and the reference essay. While the approach is a regression at its core, scores are scaled back to their target ranges upon prediction.

For example, if a zero-point essay was compared to a two-point reference essay, the model should output a score difference of minus two. While the original authors only compare test essays to reference essays that do not share the same score, we refrain from doing this, as we feel it is inappropriate to incorporate knowledge of the true scores of test instances into the pairing strategy.

Xie et al. (2022) demonstrate that their model has good within-prompt performance, i.e., when training a dedicated model for each prompt. We build on this and expand the approach to also allow for cross-prompt scoring. With this augmentation, one can even combine prompts that do not share the same label range. To achieve this, we carefully scale labels and model outputs. An overview of this scaling is given in Figure 5 in the Appendix.

During training, the true labels Y of individual essays are transformed to scaled labels Y_s , so that each $y_s \in Y_s$ is in the range of $[0, 1]$. Note that this scaling takes the prompt an essay belongs to into account, which means that each $y \in Y$ is scaled according to the label range of the prompt the essay belongs to. When pairing up essays to form training pairs, their target label is the score difference of the essays, i.e., their scores are subtracted. Thus, the score difference d_p of a pair will be in the range of $[-1, 1]$, because $d_p = y_i - y_j$ for $y_i, y_j \in Y_s$.

³<https://huggingface.co/allenai/longformer-base-4096>

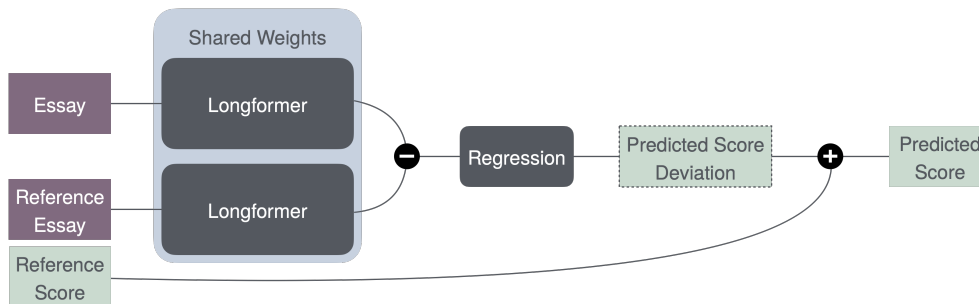


Figure 2: Overview of the model architecture, derived from Xie et al. (2022).

For optimal suitability to the regression model, we again scale the score differences to the range $[0, 1]$.

At inference, validation/test essays are paired up with training, i.e., reference essays. For each test essay t and a reference essay r , the score s_r of r is first scaled to the score range $S_t = [s_{min}, s_{max}]$ of t for compatibility. In processing the pair of t and r , the model outputs the predicted score deviation \hat{d} , which lies in the range $[0, 1]$. This now has to be mapped to the label range of t . However, because \hat{d} is a *deviation*, it has to be scaled to the range $[s_{min} - s_{max}, s_{max} - s_{min}]$, which represents the minimal and maximal deviation that is possible within S_t . The result \hat{d}_t can then be used to obtain the predicted score \hat{s}_t of t by adding the predicted score deviation to the true score s_r of r , i.e. $\hat{s}_t = s_r + \hat{d}_t$.

To investigate whether we can nudge the model towards learning less prompt-dependent representations, we contrast two ways of pairing essays during training: In the **standard** setting, we only pair essays from the same prompt. In our **generalize** setting, we only pair essays from different prompts.

Note that our main motivation is to evaluate the effect of building cross-prompt training pairs, rather than to achieve the best possible performance. In the interest of saving energy and time, we thus set our hyperparameters somewhat lower than Xie et al. (2022) did. We always train for five (as opposed to 80) epochs, taking the model with the best performance on the validation data. At inference, we limit ourselves to comparing each validation (testing) essay to 15 (25) training essays (as opposed to 50). The average predicted score is then taken as the final prediction of the model. Although our switch to a Longformer instead of the smaller BERT model increases runtime, we do not make use of the full length of 4,096 tokens. Instead, we truncate inputs to a length of 1,024, as

	PERSUADE	ASAP
# integrated prompts	7	4
# independent prompts	8	4
avg. # essays per prompt	1,733	1,622
avg. essay length in tokens	410.96	222.74
score range	1-6	prompt -dependent

Table 1: Key statistics of the two datasets used in our study.

the majority of essays fits in this length⁴. Just like Xie et al. (2022), we use a batch size of 6, and a learning rate of $1e-4$.

4 Data

We work on two different data sets, **PERSUADE** and **ASAP**-aes (Automated Student Assessment Prize - Automatic Essay Scoring), which we refer to as **ASAP**. The core statistics for each dataset can be found in Table 1. Although PERSUADE is best-suited for our analysis due to the large number of prompts, we additionally run our experiments on ASAP, as this is a commonly used essay scoring dataset.

4.1 PERSUADE

The PERSUADE dataset (Crossley et al., 2024) comprises seven integrated prompts, with a total of 12,875 essays written by students from the 6th to the 10th grade, and eight independent prompts with a total of 13,121 essays sampled from writers from the the 8th to the 12th grade. While integrated prompts refer to some source material, independent prompts do not. Each essay was annotated with a holistic score by two raters. Scores range from 1.0 to 6.0 in increments of 1.0. The raters were trained on a standardized SAT holistic essay

⁴3% of PERSUADE essays and 2.7% of ASAP essays are truncated due to this.

scoring rubric for the independent essays⁵ and its modified version for the integrated essays⁶. The main difference between the two rubrics is that the one for the integrated prompts mentions having to include evidence from the reading text⁷. Due to this explicit inclusion of the source text in the rubric, we expect the cross-prompt transfer to be more successful for the independent prompts. Overall, raters showed a strong agreement (weighted $\kappa = .74$) in annotating the essays.

4.2 ASAP

The ASAP dataset⁸ is one of the benchmark datasets for automated essay scoring. It contains four integrated and four independent (persuasive/narrative/expository) tasks, spanning a total of 12,978 essays. The essays were written by students from the 7th to the 10th grade. ASAP prompts have also been scored holistically but using a wide variety of different scales. Each essay was evaluated by two raters, with an inter-annotator agreement of $\kappa = .55$. After adjudication, the resulting score ranges can span as little as four or up to 61 different labels, as can be seen in Table 5 in the Appendix. This label incompatibility between prompts further complicates cross-prompt scoring.

5 Experimental Study

In the following, we first describe the overall setup and then present the results of our similarity-based cross-prompt scoring on the two datasets. Our experiments ran on Nvidia Quadro RTX 6000, A40, and A6000 GPUs for around 550 hours.

5.1 Experimental Setup

Our overall goal is to train an essay-scoring classifier that focuses on general indicators of a good essay as opposed to overly relying on prompt-specific features. In our similarity-based method, we facilitate this through the selection of training pairs. We contrast the performance of models trained using pairs that consist of two answers to the same vs. different prompts.

Data Split For each of our datasets, we sample the same number of answers for each prompt,

⁵https://github.com/scrosseye/persuade_corpus_2.0/blob/main/sat_rubric_only_indy.pdf

⁶https://github.com/scrosseye/persuade_corpus_2.0/blob/main/sat_rubric_only_source_based.pdf

⁷The reading texts were not published, which is why we are unable to include them in our analyses.

⁸<https://www.kaggle.com/c/asap-aes>

downsampling to the number of answers of the prompt with the lowest answer count. In doing this, we randomly sample a subset of 1,000 essays for each of the 15 prompts in the PERSUADE dataset. 800 of these are used for training and 100 for validation and testing each. For each of the eight prompts in the ASAP dataset, we randomly sample a subset of 700 essays. 560 of these are used for training and 70 for validation and testing each.

In similarity-based scoring, the training data pool is used to build pairs of instances. We derive our strategy to build these pairs from Xie et al. (2022) but relax it to allow data from multiple prompts to be paired. Their strategy includes dropping training pairs of essays with the same score, which we in preliminary experiments found to be a reasonable step, as it cut training time at a minor performance loss.⁹ However, we have to ensure that each run, i.e., all combinations of different prompts we use in our experiments, uses the same number of training pairs. Otherwise, runs with more pairs may have a performance advantage. We thus pre-calculate the maximum number of pairs we can build in each of our runs: We determine how many pairs we would end up with if we paired up all essays in the training data that do not share the same score. We then take the minimum of this as the number of training pairs we build in our experiments. This results in 1,495 training pairs for PERSUADE and 920 training pairs for ASAP.

As mentioned earlier, we limit the number of pairs during validation (testing) to 15 (25) pairs per essay. The pairing strategy for the validation data reflects the training setting: If training is done on pairs of essays from the same prompt, validation instances are also paired with training essays from the same prompt. If training is done on pairs of essays from different prompts, we also pair validation essays with training essays from a different prompt. The pairing strategy during testing is ‘greedy’ in the sense that we check whether essays from the same prompt appeared in the training data. If this is the case, we use 25 of these as reference answers, otherwise, we randomly take 25 essays from the training pool as reference answers.

Single-Prompt Baseline As a starting point for our experiments regarding the impact of training

⁹Note that this only applies to the *training* process. As we remark in Section 3, we do not look at scores when building pairs for *test* instances, since we feel that this incorporation of knowledge about scores would be inappropriate.

on combinations of data from multiple prompts, we train models on **single prompts**.

To compare the performance of the similarity-based approach, we also train an **instance-based** classifier. For this instance-based classification, we use the same *longformer_base_4096* model that is also at the heart of the similarity-based approach and attach a classification head. In both instance-based and similarity-based training, we use the same data splits, but for the instance-based classification we adapt the labels of the ASAP dataset to allow for cross-prompt evaluation. To unify the differing label ranges of the ASAP prompts, scores are scaled into a range from 0 to 3, which corresponds to the smallest label range present in the dataset¹⁰. Models are trained for 10 epochs with a maximum input length of 1,024 tokens, a learning rate of 1e-5, and a batch size of 2.

Mixed-Prompt Scoring Setup We compare models trained on answer pairs from the same prompt to models that were trained with pairs of answers to different prompts.

We vary how many prompts are combined in the training data and hypothesize that combining more prompts leads to a better generalizability of the classifier, i.e., a better cross-prompt performance. To ensure comparability, we keep the overall number of training instances constant for all combinations. The validation data is composed of the same prompts that appear in the training data to make the transfer to the prompts in the test data a hard one. Just as for the training data, the amount of validation data is also downsampled to keep it at the same overall number of instances as when data from a single prompt is used.

For PERSUADE, we report individual results for the seven integrated and eight independent prompts, and for combinations of all 15 prompts. As ASAP only comprises a total of eight prompts (4 integrated, 4 independent), we do not perform a separation into integrated and independent prompts for this dataset and only report results for the gradual combination of all eight prompts. Whenever there are more than ten possible combinations of prompts (e.g., there are 70 ways of picking four out of the eight independent PERSUADE prompts), we randomly sample ten combinations to cut training time, making sure that each prompt was selected in

¹⁰Note that this is not necessary for the similarity-based scoring, as this method comes with the capability to internally scale prompts with different label ranges into compatibility.

at least one combination.

Evaluation We always evaluate in two different conditions: **within-prompt**, which comprises the test data splits for all prompts that also appear in the training data for that run, and **cross-prompt**, which comprises the test data splits of all other prompts. We expect an increasing number of prompts mixed in the training data to have different effects for the two training and evaluation conditions. Overall, within-prompt evaluation should perform better than cross-prompt evaluation. For cross-prompt evaluation, we expect the generalized training to outperform the standard training. When evaluating in the within-prompt condition, the expectation would be for the models obtained with standard training to outperform those resulting from generalized training, as the former are more attuned to prompt-specific information.

The metric we use to evaluate model performance is quadratically weighted kappa (QWK; Cohen (1968)). Whenever we average QWK results, we perform Fisher Z-transformation to stabilize the variance.

5.2 Results: Single-Prompt Training

Before reporting the results of training on combinations of prompts, we first establish the performance level achieved by training on a single prompt. These results are shown in Table 2.

It is expected that a model will perform best when trained exclusively on data from the same prompt it is later evaluated on. This could thus be seen as somewhat of an upper bound. Table 2 also contains cross-prompt performance, first on all cross-prompt test data and then separated into integrated and independent prompts. We observe that for both PERSUADE and ASAP alike, there is a clear drop in the performance of cross-prompt compared to within-prompt evaluation.

In the case of PERSUADE, models trained on integrated prompts fare similarly in the cross-prompt evaluation, irrespective of whether the test prompts are integrated or independent. However, for models trained on independent prompts, cross-prompt evaluation within the same group (i.e., on another independent prompt) shows an average improvement of 0.16 QWK compared to evaluation on an integrated prompt. This pattern differs for ASAP, perhaps due to the widely varying scoring ranges. Here, the performance of evaluating on integrated vs. independent prompts is similar for models trained on

Train	Within-Prompt	Cross-Prompt		
		All	 	
PERSUADE				
0	.76	.62	.56	.66
1	.85	.66	.64	.67
2	.66	.41	.40	.41
3	.75	.60	.64	.56
4	.72	.53	.61	.46
5	.69	.63	.61	.64
6	.71	.53	.58	.48
7	.80	.60	.52	.67
8	.81	.65	.57	.73
9	.82	.62	.52	.70
10	.67	.59	.52	.65
11	.74	.66	.61	.71
12	.73	.55	.47	.63
13	.83	.69	.60	.75
14	.68	.56	.45	.64
Avg.	.74	.57	.58	.56
Avg.	.77	.62	.53	.69
Avg.	.75	.60	.56	.63
ASAP				
3	.70	.37	.36	.38
4	.80	.39	.42	.36
5	.79	.53	.49	.56
6	.81	.53	.71	.36
1	.79	.40	.40	.39
2	.70	.48	.48	.47
7	.81	.49	.53	.45
8	.75	.30	.18	.44
Avg.	.78	.46	.51	.42
Avg.	.76	.42	.40	.44
Avg.	.77	.44	.46	.43

Table 2: QWK performance of models trained on single prompts. Results distinguish integrated and independent prompts. The mapping of prompt numbers to names in PERSUADE is listed in Table 4 in the Appendix.

an independent prompt, but we see a benefit when models trained on an integrated prompt are evaluated on a different integrated prompt.

Comparison to Instance-Based Scoring We further examine the validity of the similarity-based approach by comparing it to a standard instance-based setting. Table 3 compares the average performance of the similarity-based (taken from Table 2) and the instance-based approach. The two setups perform on par on PERSUADE, and similarity-based scoring even outperforms the instance-based classification on ASAP.

5.3 Results: Training on Multiple Prompts

Figure 3 shows the results for training on a mix of different prompts. The number of prompts in the training data gradually increases from left to right. Note that curves start with the results from

the previous experiment, where we only trained on a single prompt. A constant benefit of building cross-prompt as opposed to within-prompt training pairs can be observed: The performance of models trained using within-prompt training pairs (dotted lines) tends to drop off, while models trained on cross-prompt training pairs (solid lines) tend to remain more stable or even increase in performance. Contrary to our hypothesis, training on cross-prompt pairs even consistently leads to better performance than standard training for the within-prompt evaluation, thus showing that this training setup does no harm but instead benefits performance across the board.

Strikingly, from a mixture of five prompts onward, our generalization-focused models perform better in cross-prompt evaluation (solid line with crosses) than the standard (i.e., within-prompt-trained) models on within-prompt data (dotted lines with dots) on the ASAP data. We see the same result from a mix of six prompts onward for the longer PERSUADE curves. With the shorter PERSUADE curves, the two conditions again meet at the mark of combining five prompts but remain on a similar performance level from there on. Thus, when five or more prompts are combined, the generalized training strategy pushes cross-prompt performance above standard within-prompt training and evaluation.

6 Embedding Space Analysis

To gain an understanding of how the embedding space is affected by either training exclusively on within-prompt or cross-prompt training pairs, Figure 4 shows embedding space visualizations. To produce these visualizations, we embed the respective test data using the Longformer model that is at the core of the model pipeline. We then use t-SNE to bring the embeddings into 2D space. For t-SNE, we use the sklearn (Pedregosa et al., 2011) implementation at its default values. From the distributions of essay embeddings, one can gather that the models trained using cross-prompt training pairs produce embeddings that are less separated into individual prompts, indicating that they truly learned a more generic representation of the essays.

7 Conclusion and Outlook

Our baseline results confirm the overall solid performance of the model, in line with what Xie et al. (2022) found. In addition, our results demonstrate

	PERSUADE								ASAP							
	Instance-based				Similarity-based (ours)				Instance-based				Similarity-based (ours)			
	Within-Prompt	Cross-Prompt			Within-Prompt	Cross-Prompt			Within-Prompt	Cross-Prompt			Within-Prompt	Cross-Prompt		
	All			All				All				All				
Avg.	.76	.57	.59	.55	.74	.57	.58	.56	.80	.40	.56	.26	.78	.46	.51	.42
Avg.	.77	.62	.55	.69	.77	.62	.53	.69	.60	.27	.23	.33	.76	.42	.40	.44
Avg.	.76	.60	.57	.64	.75	.60	.56	.63	.71	.33	.41	.29	.77	.44	.46	.43

Table 3: Comparison of instance-based and similarity-based scoring, split into the two datasets and their [integrated](#) and [independent](#) prompts. Both methods perform on par, except for similarity-based scoring outperforming instance-based scoring on the independent ASAP prompts.

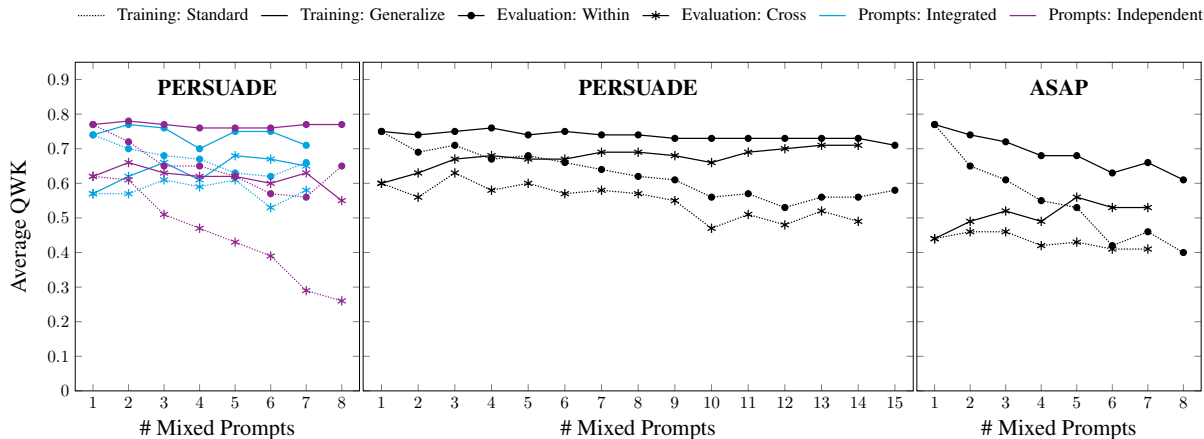


Figure 3: Learning curves depicting how mixing an increasing number of prompts in the training data affects performance. For cross-prompt evaluation, we always use test data from all (i.e., both integrated and independent) prompts that are not in the training data. Our generalized training strategy (solid lines) consistently benefits performance compared to standard training (dotted lines).

the suitability of the model to perform cross-prompt scoring - even in the difficult case of the ASAP dataset with its diverse set of score ranges across different prompts. Our strategy of pairing training essays either within-prompt or cross-prompt proved helpful not only in the cross-prompt scenario but also for within-prompt evaluation. Thus, it is advisable to build training pairs cross-prompt whenever a mixture of multiple prompts is present in the training data.

Limitations and Ethical Considerations

In our setup, we only investigate variants of a hard domain transfer, where data from several source domains is used to train a classifier that is then applied to a target domain. One obvious next step we have not yet taken would be to inject small amounts of target-domain data. Another avenue we do not incorporate is to use the source text of a prompt as a means of facilitating cross-prompt transfer.

Similarly, we do not evaluate cross-prompt performance between datasets. In this study, we re-

strict ourselves to cross-prompt evaluations within ASAP or PERSUADE (as in almost all related work), i.e., we evaluate on new prompts that are somewhat similar to the source prompts and whose data comes from a similar learner population. The question of the extent to which essay scoring can ever be fully generic remains open and thus requires further research.

As always in automated scoring, fairness and bias are important issues that should be taken into account to make sure that scoring algorithms do not disadvantage certain user groups (see, e.g., Loukina et al. (2019) and Schaller et al. (2024)). These topics also need further investigation for our generic scoring scenario. At the same time, one might argue that a generic classifier is less likely to fall for spurious correlations between scores and unnecessary features than a prompt-specific classifier might be.

Finally, as our experimental setup requires over 1,000 training runs, we make some design choices in the interest of keeping the overall runtime at a reasonable level. Our preliminary results indi-

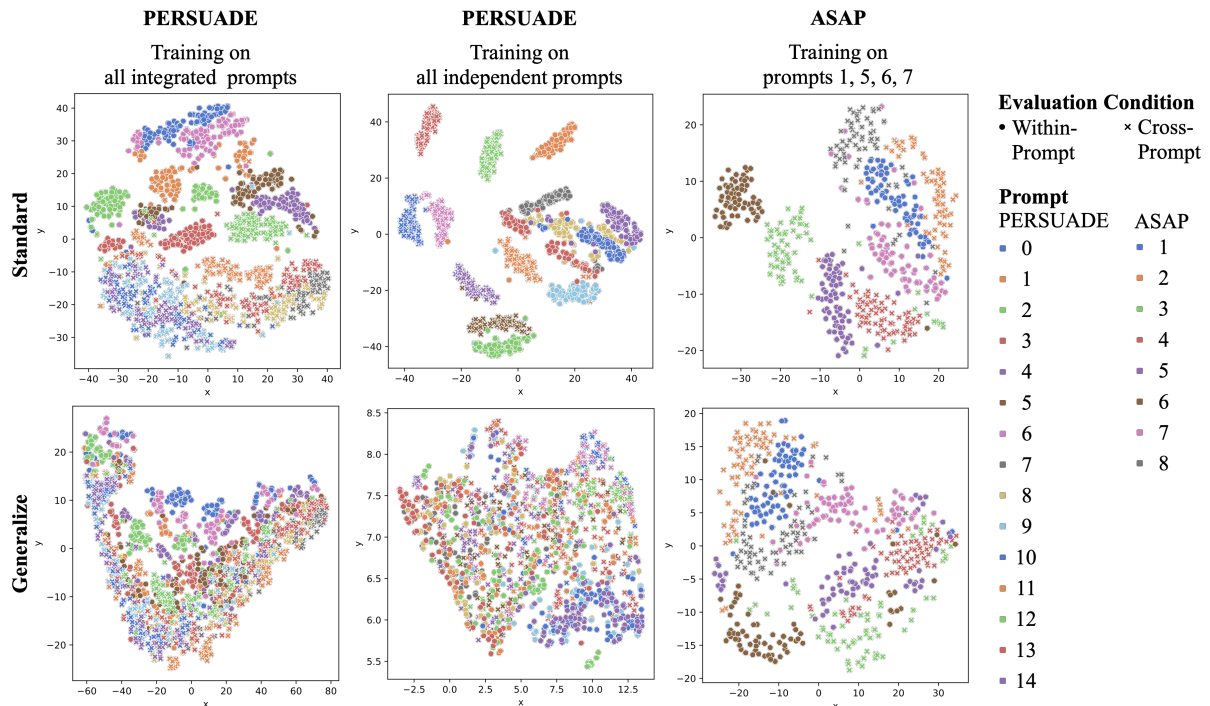


Figure 4: Visualization of embeddings from models trained in the standard (top) and generalize (bottom) conditions, transformed using t-SNE. There is less separation into prompts for models trained with the generalize strategy, indicating that these models do in fact learn a more generalized representation of the essays.

cate that one could achieve better performance than what we report here by training for more than just five epochs, building more training pairs and taking advantage of the full input length of the Longformer model - albeit at the cost of a greater demand on computing resources.

References

- Stacey Bailey and Detmar Meurers. 2008. Diagnosing meaning errors in short answers to reading comprehension questions. In *Proceedings of the third workshop on innovative use of NLP for building educational applications*, pages 107–115.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. *Longformer: The Long-Document Transformer*. *arXiv:2004.05150 [cs]*.
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2022. *Similarity-based content scoring - how to make SBERT keep up with BERT*. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 118–123, Seattle, Washington. Association for Computational Linguistics.
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2023. Similarity-based content scoring—a more classroom-suitable alternative to instance-based scoring? In *Findings of the association for computational linguistics: Acl 2023*, pages 1892–1903.
- Yuan Chen and Xia Li. 2023. *PMAES: Prompt-mapping contrastive learning for cross-prompt automated essay scoring*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1503, Toronto, Canada. Association for Computational Linguistics.
- Jacob Cohen. 1968. *Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit*. *Psychological Bulletin*, 70(4):213–220. Place: US Publisher: American Psychological Association.
- 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.
- Tirthankar Dasgupta, Abir Naskar, Lipika Dey, and Rupsa Saha. 2018. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 93–102.
- 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.
- 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.
- 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.
- Andrea Horbach and Torsten Zesch. 2019. The influence of variance in learner answers on automatic content scoring. In *Frontiers in education*, volume 4, page 28. Frontiers Media SA.
- Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. TDNN: a two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1097.
- Xia Li, Minping Chen, and Jian-Yun Nie. 2020. SEDNN: Shared and enhanced deep neural network model for cross-prompt automated essay scoring. *Knowledge-Based Systems*, 210:106491.
- 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.
- 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.
- 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.
- 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.
- Peter Phandi, Kian Ming A Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 431–439.
- 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.
- Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891.
- 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.
- Jiong Wang, Qing Zhang, Jie Liu, Xiaoyi Wang, Mingying Xu, Liguang Yang, and Jianshe Zhou. 2025. [Making meta-learning solve cross-prompt automatic essay scoring](#). *Expert Systems with Applications*, 272:126710.
- Jiayi Xie, Kaiwei Cai, Li Kong, Junsheng Zhou, and Weiguang Qu. 2022. [Automated Essay Scoring via Pairwise Contrastive Regression](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2724–2733, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Chunyun Zhang, Jiqin Deng, Xiaolin Dong, Hongyan Zhao, Kailin Liu, and Chaoran Cui. 2025. [Pairwise dual-level alignment for cross-prompt automated essay scoring](#). *Expert Systems with Applications*, 265:125924.

A Appendix

This appendix contains supplementary information to increase the transparency and reproducibility of our experiments. Table 4 gives information on the mapping between prompt numbers and names in PERSUADE. For ASAP, Table 5 gives information on the label ranges of the different prompts. To better grasp the generalization of the model for cross-prompt scoring, Figure 5 presents a graphic overview of how labels are scaled during training and inference.

#	Prompt Name
0	The Face on Mars
1	Facial action coding system
2	A Cowboy Who Rode the Waves
3	Does the electoral college work?
4	Car-free cities
5	Driverless cars
6	Exploring Venus
7	Summer projects
8	Mandatory extracurricular activities
9	Cell phones at school
10	Grades for extracurricular activities
11	Seeking multiple opinions
12	Phones and driving
13	Distance learning
14	Community service

Table 4: Prompt mapping in the PERSUADE dataset.

Prompt	Label Range	
	From	To
Integrated Prompts		
3	0	3
4	0	3
5	0	4
6	0	4
Independent Prompts		
1	2	12
2	1	6
7	0	30
8	0	60

Table 5: Label ranges in the ASAP dataset.

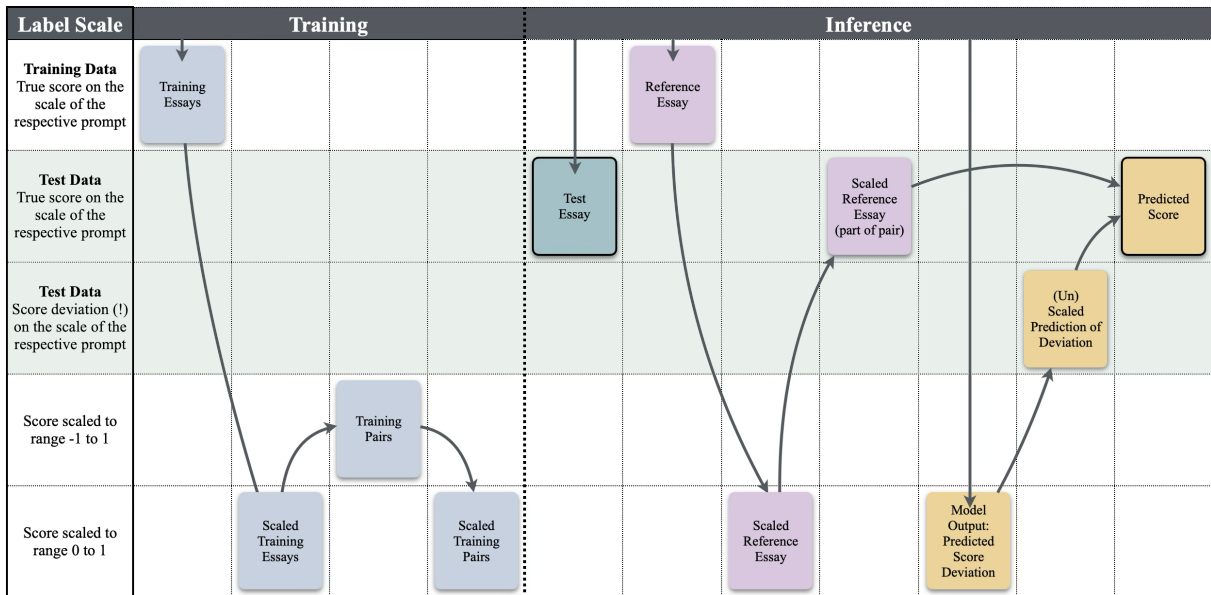


Figure 5: Overview of how labels are scaled to achieve compatibility between score ranges when training on a mix of answers to different prompts.

Automated Scoring of a German Written Elicited Imitation Test

Mihail Chiffigarov¹, Jammila Laâguidi¹, Max Schellenberg¹, Alexander Dill¹,
Anna Timukova^{2,4}, Anastasia Drackert^{3,4} and Ronja Laarmann-Quante¹

Ruhr University Bochum, Germany
firstname.lastname@rub.de

¹Faculty of Philology, Department of Linguistics

²University Language Centre (ZFA)

³Faculty of Philology, Institute for German Language and Literature

⁴g.a.s.t. (Society for Academic Study Preparation and Test Development)

Abstract

We present an approach to the automated scoring of a German Written Elicited Imitation Test, designed to assess literacy-dependent procedural knowledge in German as a foreign language. In this test, sentences are briefly displayed on a screen and, after a short pause, test-takers are asked to reproduce the sentence in writing as accurately as possible. Responses are rated on a 5-point ordinal scale, with grammatical errors typically penalized more heavily than lexical deviations. We compare a rule-based model that implements the categories of the scoring rubric through hand-crafted rules, and a deep learning model trained on pairs of stimulus sentences and written responses. Both models achieve promising performance with quadratically weighted kappa (QWK) values around .87. However, their strengths differ – the rule-based model performs better on previously unseen stimulus sentences and at the extremes of the rating scale, while the deep learning model shows advantages in scoring mid-range responses, for which explicit rules are harder to define.

1 Introduction

The Written Elicited Imitation Test (WEIT) is a computer-based test designed to measure procedural linguistic knowledge in writing. In this test, learners briefly view sentences in the target language and, after a short pause, reproduce them from memory by typing. Responses are then rated on an ordinal scale based on how closely they resemble the original sentences.

Like any assessment that relies on scoring by human raters, the WEIT can benefit greatly from automation. An automated scoring system would significantly improve efficiency by enabling the rapid evaluation of large numbers of responses without the time and effort required by human raters. This would, in turn, allow for immediate feedback, an

advantage in both instructional and research contexts. Automation also ensures greater consistency and objectivity by applying scoring criteria uniformly and eliminating potential rater bias. In addition, automated systems can provide fine-grained data on error patterns and processing behavior, offering deeper insight into learners' procedural language skills.

In this paper, we investigate the automated scoring of a German WEIT. The responses in our dataset are scored using a rubric that assigns a score between 0 and 4, based on deviations in spelling, grammar, and vocabulary (see Section 3.2). There are two main approaches to automating this process: a rule-based approach, in which categories from the scoring rubric are implemented explicitly, and a deep learning approach, in which a model learns implicitly which scores to apply based on training pairs of stimulus and response sentences.

In educational settings, transparency and explainability are important considerations. From this perspective, rule-based models are preferable as they allow for a clearer justification of scoring decisions and can offer more detailed feedback to learners by pinpointing specific types of deviations. However, rule-based systems can be limited in flexibility, particularly when dealing with edge cases or language exceptions. In contrast, deep learning models may be better suited to capturing subtle patterns in learner responses (e.g. to what extent a word substitution affects the overall meaning of the sentence), but often lack transparency and may struggle to generalize to previously unseen stimulus sentences.

This paper presents a rule-based scoring model for the WEIT, built on general principles derived from the scoring rubric, and compares it to a deep learning model trained on stimulus-response pairs. We hypothesize that (a) the deep learning model will outperform the rule-based model on cases where differences between descriptors in the scor-

ing rubric are rather subtle and hard to capture as explicit rules, and (b) the rule-based scoring model will generalize better to new, previously unseen stimulus sentences.

The main contributions of this paper are twofold. First, we explore the feasibility of automating the scoring of a German WEIT using a detailed ordinal rating scale. Second, we provide a concrete case study for comparing the strengths and limitations of deep learning and rule-based methods in an educational assessment context.

Our code and data are available at: <https://gitlab.ruhr-uni-bochum.de/vamos-cl/german-weit-automated-scoring>.

2 Background and Related Work

In the following, we provide background on the use of elicited imitation tests and summarize previous research on their automated scoring.

2.1 Elicited Imitation Tests (EIT): Construct and Use

Elicited imitation tests (EIT) have been widely used and researched in the field of Second Language Acquisition as measures of two key constructs: global proficiency in a second or foreign language (Drackert, 2016; Kostromitina and Plonsky, 2022) and implicit language learning (Nikouee and Ranta, 2023). EITs exist in many languages and have been primarily employed in the oral mode (oral EIT, or OEIT) in which language learners listen to a number of sentences and then orally repeat them as accurately as possible after a short pause.

Recently, written elicited imitation tests (WEIT) have started to gain attention in language testing as a research tool (e.g. Sun et al., 2025) or as a measure of literacy-dependent procedural language knowledge (Timukova et al., submitted).

In the format that was used by Timukova et al. (submitted) in a large language testing research project, the sentences are briefly presented on a screen, and, after a pause, learners have to reproduce as much of the sentence as they can by typing their response into a text box. The pause is intended to reduce the influence of working memory and to promote active reconstruction of the stimulus rather than rote repetition. The construct of literacy-dependent procedural language knowledge measured by WEIT can be defined as automatized knowledge and skills required for the real-time reception and production of written language.

Inspired by and closely related to the well-established oral elicited imitation format (Ortega et al., 2002), the written test — despite being presented and completed in a different modality and incorporating a distinct scoring system to better capture the construct (see Section 3.2) — yields results of comparable difficulty and reliability.¹ However, it is considerably easier to develop, administer, and score, as no audio equipment is required at any stage. Scoring short written responses is also likely more practical and less time-consuming than scoring spoken responses when done by human raters.

2.2 Automated Scoring of EITs

While EITs, in principle, lend themselves well to automated scoring since the target response is known (i.e. exact repetition of the stimulus sentence), the difficulty of the automated scoring task largely depends on the scale or rubric used for rating responses that deviate from the target.

For the oral EIT, numerous studies have explored automated scoring of the test using automatic speech recognition (ASR), primarily employing a binary scale that codes whether the response matches the stimulus or not (e.g. Millard, 2011), or an interval scale, where, for example, one point is subtracted for each deviation in the response sentence (Graham et al., 2008; Lonsdale and Christensen, 2011). Once the learner utterances are accurately transcribed, automated scoring based on these scales is straightforward.

Besides binary and interval scales, ordinal scales exist where scores are determined qualitatively. In their meta-analysis, Yan et al. (2016) found that for the OEIT, ordinal rating scales were more effective at distinguishing speakers across proficiency levels than other scales. An established ordinal rating scheme for the OEIT is that of Ortega et al. (2002), where the score depends on how much of the stimulus sentence a learner was able to repeat:

- 0 points for minimal (one word), unintelligible responses or no repetition
- 1 point when half or less of the stimulus was repeated
- 2 points for changes to the original sentence in content or form that affected the meaning

¹The tests used in the project showed difficulty indices of 0.41 (WEIT) and 0.49 (OEIT), and reliability coefficients (Cronbach's α) of .97 for both (N = 195).

- 3 points for accurate content repetition with some (un)grammatical changes
- 4 points for exact repetition with formal accuracy

Recent studies have investigated how automatically obtained scores based on objectively quantifiable features correlate with scores based on Ortega’s ordinal scale. McGuire and Larson-Hall (2025) found high correlations with word error rate (WER), especially when looking at a participant’s mean score across a whole test ($r = -0.969$). The correlation of WER and Ortega’s scores across items, however, was lower ($r = -0.817$). Isbell et al. (2023) took further metrics such as Levenshtein distance into account and also mapped a combination of Percent Word Correct (PWC, exact matches) and Percent Meaning Correct (PMC, matching lemmata) to Ortega’s 5-point scale (e.g. $PWC < 100\%$ and $PMC \geq 70\% = \text{Score } 3$). They also found high correlations with Ortega’s scores assigned by human raters (around $r = 0.9$ when aggregated across all items and around Spearman’s $\rho = 0.8$ at the item level, depending on the metric and ASR service used).

In the present study, our aim is to implement an automated scoring procedure for a German WEIT, using an ordinal scale similar to that of Ortega et al. The scoring rubric will be presented in more detail in Section 3.2. It was specifically developed for the German WEIT, as no comparable schemes had yet existed. As the purpose of the WEIT is to test literacy-dependent procedural language knowledge, the rubric differs in some essential ways from that of Ortega et al. Our goal is to build a rule-based scoring model that implements the various categories from the rubric, rather than relying on purely quantitative measures such as WER. This scoring method is comparable to human raters’ assessments in that it could provide learners with feedback about the scores they received based on the deviations in their responses. For comparison, we investigate how successful modern deep learning approaches are at approximating human ratings by implicitly learning to apply the scoring rubric.

3 Data

3.1 Data Collection

The data for our study was collected within a larger research project where the WEIT was used as a measure of literacy-dependent procedural knowledge (see Section 2.1). The 20 items included in the

WEIT range from 6 to 16 words, or 8 to 24 syllables (see Appendix A for the full list of items). The test was completed by 195 university students who were learners of German (58.1% female, 41.9% male) between the ages of 18 and 40 ($M = 25.46$, $SD = 3.92$). The participants represented 47 different native languages, with Russian ($n = 30$), Turkish ($n = 23$), English and Spanish ($n = 14$ each) being the most frequent. Most participants self-assessed their language skills to be somewhere between A2 and C1.

3.2 Scoring Rubric

An ordinal scoring rubric for the German WEIT was developed for the purposes of the project. It follows the rubric of Ortega et al. (2002) in that responses are scored based on how closely they resemble the stimulus sentences. A key difference between the WEIT rubric and the OEIT rubric already addressed in Section 2.2 is the altered role of *meaning* and *grammar*. Since rule-governed morphological and syntactic sequences are central to the construct of procedural knowledge measured by the WEIT, grammatical deviations carry more weight. Hence, the rubric distinguishes between lexical and grammatical deviations from the original, assigning a higher score (Score 3) for responses with lexical deviations (e.g., lexical omissions or substitutions) and a lower score (Score 2) for responses with grammatical errors (e.g., structural omissions or incorrect prepositions).

In the following, we present a summary of the scoring rubric. Its use is exemplified for item #2 in Table 1. The complete scoring rubric can be found in the Supplementary Material to this paper.

Score 4 The response matches the stimulus sentence exactly or 1–2 typos are present.²

Score 3 Changes in grammar or lexical changes that preserve the original structure and result 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

²Typos 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’.

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.

Some score descriptors vary with the length of the stimulus sentence: in “shorter sentences” (≤ 15 syllables) fewer deviations are allowed than in “longer sentences” (> 15 syllables). If a response contains multiple deviations at different score levels, then the lowest score determines the overall score. An accumulation rule is applied when two or more deviations of the same level are present in scores 2 or 3, leading to an overall score of 1 or 2, respectively. Punctuation and capitalization of the first word of the sentence are not taken into account.

The gold standard scores (henceforth also referred to as ‘gold scores’) for our study were assigned by three human raters in the following procedure: First, they familiarized themselves with the assessment rubric and participated in a calibration session using 200 responses (i.e. for all 20 items a sample of 10 participants each). Following this, a sample of the same size was randomly selected for independent evaluation by each rater. The inter-rater reliability (Fleiss’ κ) for the resulting 200 ratings averaged around .986, indicating almost perfect agreement, with values ranging from .895 to 1.0 across the 20 items. The remaining responses were rated by one rater each, and ratings were discussed by all raters throughout the process to address difficult cases and ensure consistency.

3.3 Data Splitting

Each of the 195 participants responded to 20 different stimulus sentences (*items*). In total, our dataset comprises 3,900 pairs of stimulus (*target*) and imitation (*response*) sentences. We split the data into training, validation, and two different test sets as

Score	Train	Val.	Test known	Test unk.	Total
0	1,095	25	25	87	1,232 (32%)
1	701	25	25	92	843 (22%)
2	553	25	25	62	665 (17%)
3	260	25	25	78	388 (10%)
4	651	25	25	71	772 (20%)
Total	3,260	125	125	390	3,900 (100%)

Table 2: Number of stimulus-response pairs in the training, validation, and test sets, respectively, per gold score. ‘Test unk.’ contains stimulus sentences held out from the training set, ‘Test known’ a random subset of the remaining data (i.e. stimulus sentences known in the training set).

follows: First, we set aside all responses to two of the stimulus sentences (#4 and #18, i.e. one ≤ 15 syllables and one > 15 syllables, see Section 3.2). We call this test set ‘Test unknown’, comprising 390 stimulus-response pairs in total. The rest of the data was randomly split into a training, validation, and another test set. We call this second test set ‘Test known’, because it contains responses to those stimulus sentences that are also part of the training set. By using these two different test sets, we are able to not only assess how well a model performs on unseen response sentences to known stimulus sentences but also how well it generalizes to completely new stimulus sentences. The resulting data distribution across sets and gold scores is shown in Table 2.

4 Method

We first present our deep learning model (**DL model**) and then introduce the pipeline for the rule-based model (**RB model**) for automatically scoring the WEIT.

4.1 Deep Learning

Since there is not enough data to train a deep learning model from scratch, we decided to use a pre-trained transformer model and fine-tune it on our

data for multi-class sequence classification.

For efficiency reasons, we chose the DistilBERT model (Sanh et al., 2019), a distilled version of BERT (Devlin et al., 2019), which the authors showed to be 40% smaller, 60% faster and able to retain 97% of the original language understanding capabilities (Sanh et al., 2019). We used the pre-trained model for German cased data (distilbert-base-german-cased) with a multi-label classification head, and fine-tuned the model on our WEIT training set. Hyper-parameters were optimized based on accuracy on the validation set, yielding the following setup and parameter values: a learning rate of $1e-5$ and an epsilon value of $1.5e-3$ for the Adam optimizer, and the default loss function for multi-class classification (SparseCategoricalCrossentropy). We trained the model for 50 epochs with an early stopping mechanism triggered after 5 consecutive epochs without improvement in the validation loss. The training and validation data were shuffled and batched in each iteration, with a batch size of 16.

Since the training dataset was heavily skewed towards scores 0, 1, and 4 (see Table 2), we trained a second model in the same way but in which class weights were introduced for scores 2 and 3. Score 2 received a 2x multiplier and score 3 received a 4.5x multiplier, both approximately equal to the proportion of the corresponding training pairs of these scores to the number of score 0 pairs (the most common score). We refer to this as the **weighted** DL model and the model without adjusted class weights as the **unweighted** DL model.

4.2 Rule-Based

The rule-based model processes pairs of target and response sentences through a multi-step pipeline to generate a score (Figure 1).

Preprocessing In the preprocessing step, the sentences are normalized and cleaned so that differences between target and response sentences that are not relevant for scoring can be ignored. This means in particular: capitalizing the first letters of both sentences, transforming common umlaut variants into the correct character (e.g., ‘ae’ into ‘ä’), and removing punctuation and artifacts such as the ‘;timeout’ token, which appears when a participant runs out of time during the repetition process. Furthermore, in some cases participants repeated the sentence multiple times. Since this is ignored by the human raters, we cut each response to only

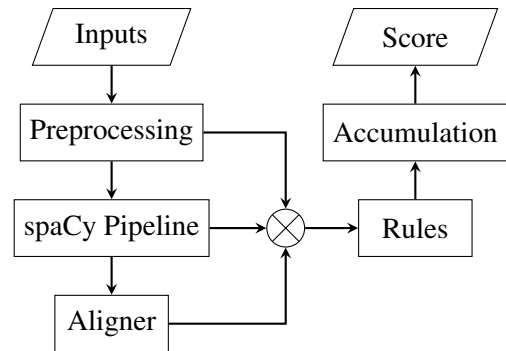


Figure 1: Flow diagram illustrating the rule-based model’s data processing pipeline.

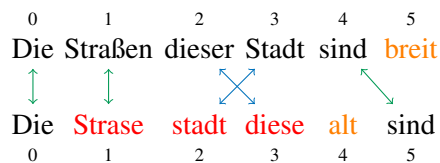


Figure 2: Token mapping by the aligner function for an example sentence. Tokens in red are **misspelled** and tokens in orange are **missing** or **additional**. Green arrows denote aligned tokens and blue arrows **transpositions**.

retain the first response sentence.

Linguistic Annotation In the next step, the pre-processed sentences are analyzed linguistically using a spaCy pipeline (Honnibal et al., 2020), which transforms each sentence into a list of tokens with part-of-speech (POS) tags, syntactic dependency labels, morphological features, and syllable counts.³

Alignment The tokens are then passed on to a custom-built aligner, which maps the words in the response to the words in the target sentence and also detects missing or added words (see Figure 2). This is done by calculating a matrix of Damerau-Levenshtein distances⁴ between all words in the response sentence and the target sentence and aligning those words with the smallest distance. We do not only align identical words because this would prevent misspelled words from being matched with the correct word in the target sentence. However, if the edit distance between two words is large, it is more likely a different word rather than a misspelling. Therefore, for two words to be aligned,

³We use spaCy v3.8.3 with the de_core_news_sm model v3.8.0 with all its default components, and the package *sloev/spacy-syllables* v3.0.2 for counting the syllables, which is added directly after the tagger in the spaCy processing pipeline.

⁴using *lanl/pyxDamerauLevenshtein* v1.8.0, <https://github.com/lanl/pyxDamerauLevenshtein>

their edit distance must be ≤ 3 .⁵ If a word in the target sentence has no match in the response sentence, it is ‘missing’, if a word in the response sentence has no match in the target sentence, it is considered ‘additional’. Note that at this step, we do not detect word substitutions directly but they would be treated as a missing and an additional token, which are later aligned by a rule that checks for substitutions. For all matched pairs, if the edit distance between both words is greater than zero, the token in the response is considered ‘misspelled’. If two words are matched but have different positions, they are considered ‘transposed’. All other tokens are labeled as ‘correct’.

Rule Application Manually, a set of rules was crafted that implement the deviation categories from the scoring rubric based on all the outputs of the previous steps. For each category it is checked whether it applies to the response sentence. For this first version of the rule-based model, most deviation categories were implemented, except for some which were considered too fuzzy or which would have required further linguistic annotation not readily available e.g. about German plural formation.⁶ The rules are defined in a way that they are mutually exclusive so that the order in which they are applied is not important. If a rule detects that a particular category applies to a response sentence, it outputs the name of the category, the score which it is associated with and how many instances of this deviation are found. Finally, an accumulation function collects the outputs of all rules and calculates the final score (see Section 3.2 for the accumulation rules). The following examples illustrate how some of the categories from the scoring rubric are approximated via rules.

To detect an *Omission Error*, the rule uses the missing-word count from the aligner. If exactly one word is missing, the rule assigns a score of 3. If two words are missing in a sentence with fewer than 16 syllables, the score is 2. In longer sentences with two or more omissions, the rubric asks to assess whether the sentence “preserves most of the original sentence structure and most of the meaning”. We determine structural deviations by the degree

⁵This value worked well in our trial runs but could be tuned, e.g. adjusted for token length, in future work.

⁶Deviation categories that were not implemented are: *wrong plural formation*, *missing structural elements or wrong word order*, and *sentence is grammatical but not meaningful* from score 2, and *changes in grammar that preserve the original structure and result in grammatically correct sentences* from score 3.

of agreement between the spaCy dependency structures of stimulus and response sentence, with a loss of more than 30% of the original dependencies serving as the threshold. Meaning deviations are identified using cosine similarity between the vectorized representations of target and response sentences, obtained via a BERT Sentence Transformer (Reimers and Gurevych, 2019). If the similarity falls below 0.987, the meaning is considered altered.⁷ If either a structural or meaning deviation occurs, the score is set to 1, otherwise 2.

The *Changes in Grammar* category captures deviations in grammatical structure between the stimulus and response sentences, which are specifically listed in the scoring rubric, namely differences in article usage, gender and case markings, agreement violations, and prepositional errors. The rule uses information from the aligner and spaCy to compare the POS and morphological features of aligned words. Article-related errors are identified when a determiner is missing, incorrectly added, or replaced with another. Gender and case errors are identified when mismatches occur in the morphological features of aligned words. Agreement violations are detected by comparing the number feature between a verb and its subject. Finally, prepositional errors include missing or incorrect prepositions. The scoring mechanism assigns a score of 2 for each error, counting the number of detected grammatical mistakes to determine the final score.

5 Evaluation

We evaluate the weighted DL model, the unweighted DL model and the RB model on the test set with known items and unknown items, respectively, as well as on the combination of the two test sets (henceforth called combined test set). Table 3 reports the accuracy, i.e. how often the exact gold score was predicted, and Quadratically Weighted Kappa (QWK), which penalizes greater deviations from the gold score more severely than smaller deviations.

For the DL models, we expected a drop in performance when comparing the scoring of responses to known versus unknown items, but not for the RB model. In fact, we see a considerable drop for the DL models: For example, QWK decreases from .93 to .62 for the unweighted DL model and from

⁷The thresholds worked well in our trial runs but could be tuned more systematically in future work.

Model	Known It.		Unknown It.		Combined	
	acc	qwk	acc	qwk	acc	qwk
DL unw.	.71	.93	.46	.62	.52	.72
DL weigh.	.78	.94	.51	.83	.57	.87
RB	.73	.90	.66	.87	.68	.87

Table 3: Accuracy and QWK for the deep learning models (DL) without class weights (‘unw.’), with class weights (‘weigh.’) and the rule-based model (RB), respectively, 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.

.94 to .83 for the weighted DL model. For the RB model, there is only a slight drop from .90 to .87, which may also be due to chance considering the rather small test sets.

Overall, the weighted DL model is the best performing model on the known items, while the RB model is the best performing model on unknown items. On the combined test set, both models perform on par in terms of QWK (.87), but the RB model attains higher accuracy (.68 compared to .57). The weighted DL model consistently outperforms the unweighted DL model.

When looking at the confusion matrices of the three models based on the combined test set (Figure 3), we see that the greatest weakness of the unweighted DL model is that it hardly ever predicts score 3 and only rarely score 4. In fact, on unknown items it never predicts score 3 and only once score 4, hence it fails to generalize when a response is to be counted as (almost) correct. For the other two models, we see almost no large deviations from the gold standard, which was already reflected in the overall high QWK scores.⁸

5.1 Fine-Grained Model Comparison

In the following, we will restrict the discussion to the weighted DL model and the RB model and look more closely into their commonalities and differences.

From the confusion matrices (Figure 3) we see that the RB model has a distinct tendency to undervalue the responses: Out of 176 misclassified responses, 147 (84%) receive a score lower than

⁸There is one extreme outlier where the RB model predicts score 0, while the gold score is 4. This occurred because the response contained multiple repetitions of the stimulus sentence, and a bug prevented truncation of this particular case, contrary to what was prescribed by the preprocessing step described in Section 4.2.

DL	RB	Gold	Count	Perc.
•	•	•	209	41%
•	-	•	87	17%
-	•	•	140	27%
•	•	-	53	10%
-	-	-	26	5%
Total			515	100%

Table 4: Number of responses in the combined test set for which all three, only two or none of the scores given by the deep learning model (DL), the rule-based model (RB) and the gold standard are identical. ‘•’ indicates that the same score was assigned.

the gold standard, i.e. the model tends to be stricter than the human raters. For the DL model, there is a similar trend, but proportionally it is not as extreme: Out of 219 misclassified responses, 157 (72%) are undervalued (note that in terms of absolute numbers, there are more undervalued items for the DL than for the RB model).

In Table 3, we saw that the RB model had an overall higher accuracy on the combined test set than the DL model. But does this mean that it correctly predicts most of the responses that the DL model also scores correctly – plus some additional ones – or do the two models actually succeed on different sets of responses?

Table 4 shows a breakdown of how often either both models or only one of them or none agrees with the gold standard and how often the two models agree with each other on the combined test set. In sum, only for 51% of the responses, the DL model and the RB model predict the same score. When they agree with each other, this does not necessarily mean that they are correct because for 10% of the responses, both models agree but they both deviate from the human gold standard (which can, in fact, also point to human ratings being inconsistent with the scoring rubric, see Section 5.2). On the other hand, for 85% of the responses at least one of the models is correct, i.e. agrees with the human gold standard. For 27%, only the RB model is correct and for 17% only the DL model. This indicates that both models have different strengths and weaknesses that we will examine more closely in the following.

Table 5 shows a breakdown of precision, recall, and F1-score per gold score for each of the two models. We see that for all scores but score 2, the RB model performs better or on par with the DL model. For score 4, the difference is most striking, with a very high recall of the RB model (.96) and

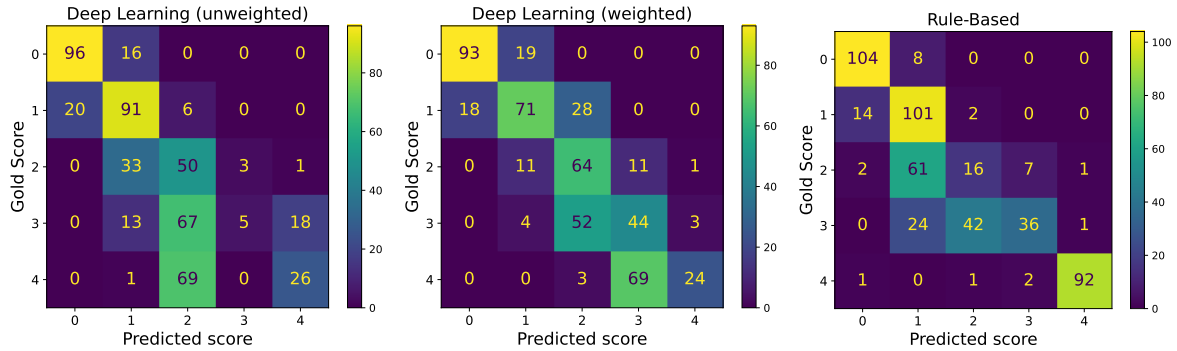


Figure 3: Confusion matrix of gold score vs. predicted score per model.

Score	Precision		Recall		F1	
	DL	RB	DL	RB	DL	RB
0	.84	.86	.83	.93	.83	.89
1	.68	.52	.61	.86	.64	.65
2	.44	.26	.74	.18	.55	.22
3	.35	.80	.43	.35	.39	.49
4	.86	.98	.25	.96	.39	.97
macro avg	.63	.68	.57	.66	.56	.64
micro avg	.64	.69	.57	.68	.57	.66

Table 5: Precision, recall, and F1-score per gold score, as well as macro average and micro (=weighted) average, for the deep learning model with class weights (DL) and the rule-based model (RB) based on the combined test set. The numbers in bold indicate the higher value in precision, recall, and F1-score, respectively, per score.

a very low recall of the DL model (.25). Only for score 2, the DL model clearly outperforms the RB model. This is partly in line with our expectation that the DL model performs better at scores where the scoring rubric categories are rather vague (e.g., responses with changes in grammar can receive either score 2 or score 3 depending on whether the sentence is still grammatical and meaningful). We will qualitatively discuss some of the misclassifications in the following section.

5.2 Discussion of Misclassifications

In the following, we will qualitatively discuss some of the misclassifications of the models to identify their potential limits and also to find leverage points for improvement.

Limitations of the deep learning model We saw that the deep learning model has a strikingly low recall for score 4. In fact, except for one response, the cases where the model failed to predict score 4 were caused by responses to the two previously un-

known items. This indicates that the model failed to generalize to new sentences when a response is to be rated as fully correct. This is the case even when the responses are exact repetitions of the target sentence (38 out of 72 misclassifications). While these misclassifications could potentially be eliminated by passing a similarity score to the model, the remaining errors are harder to mitigate. This concerns, for example, accepted typos in a response, where the advantage of the RB model is that we can specify exactly what counts as a typo.

Limitations of the rule-based model For the DL model, we do not easily know why a response was misclassified but for the RB model we can analyze which categories were missed or falsely detected. We found some systematic causes for misclassifications:

Firstly, the model sometimes fails to differentiate between spelling errors, typos, and grammatical errors. One particular problem is the treatment of real-word spelling errors, i.e. (potential) spelling errors that result in another existing word form, e.g. *fährt/fahrt* (3SG/2PL of ‘(to) drive’ or *es/er* (‘it/he’). They make the sentence ungrammatical or not meaningful but are overvalued by the model because only a spelling error is detected. On the other hand, misspellings can result in nonsense words that are unknown to spaCy, which impacts the syntactic or morphological analysis of the sentence. For example, we found that when the spelling of *Musik* (‘music’) is changed to *Music*, spaCy assigns it neuter gender (instead of feminine), so that a model classifies the sentence as containing a grammatical error.

Furthermore, the model cannot determine well whether a substitution preserves the overall meaning and grammatical structure of the sentence, leading to an undervaluation of examples like *Die*

Häuser sind nicht sehr/so schön ('The houses are not **very/so** pretty') or *Kosten der Häuser / Kosten von Häusern* ('costs of the houses'). This is mainly important to differentiate between scores 2 and 3, which explains the low performance of the model for these scores.

Limitations of the human ratings In fact, not all deviations from the gold standard turned out to be true misclassifications. In some cases the models uncovered inconsistencies in human ratings, e.g. where human raters had overlooked deviations or not followed the rubric, but these cases were rare.

6 Conclusion and Future Work

We implemented an automated scoring procedure of a German WEIT using two approaches: a rule-based approach with manually crafted rules implementing the specific categories listed in the scoring rubric, and a deep learning approach that received pairs of stimulus sentences and test-takers' responses as training data. We found that the overall performance of both kinds of models is promising but not yet optimal, and that both approaches have different strengths and weaknesses. The rule-based model outperformed the deep learning model on previously unseen stimulus sentences and for the scores at the edges of the rating scale. The deep learning model, in contrast, was more successful in some cases of mid-range scores, for which explicit rules are harder to define.

The results indicate that a promising direction for future research could be to develop an ensemble or hybrid model: using rule-based scoring for categories with very high precision, and training a DL model only for those where clear rules are difficult to define. It also remains to be investigated whether Large Language Models (LLMs) with their broad language comprehension capabilities could contribute to the automated scoring or detection of specific error categories.

Limitations

One clear limitation of our study is that we only evaluated one deep learning model (DistilBERT). Different models, especially models operating on the character level, may lead to better results, e.g. by better capturing spelling errors and typos, and are worth investigating in future work. Furthermore, given that the class weighting had a great impact on the DL model, finding the optimal

weighting could be investigated more systematically. In general, there could be a more systematic fine-tuning of hyperparameters but this would require access to larger computational resources, e.g. servers, that we wanted to avoid. Furthermore, we only used spaCy and no other tools for annotating the linguistic structure of the responses, which has a considerable impact on the overall performance of the rule-based model. Trying other or combining different linguistic processing tools could improve the results. Another limitation is that some of the deviation categories from the scoring rubric pertaining to scores 2 and 3 were not implemented yet in the rule-based model, which probably in part accounts for the weaker performance of the model for these scores compared to the other scores. Some of these rules could be implemented in future work by adding further specific resources (e.g. about German plural formation) while others, such as detecting sentences which are grammatical but not meaningful, could be tackled by using LLMs or finetuning models to specifically detect such cases.

Ethical Considerations

Our study investigated whether it is, in principle, feasible to automatically score a German WEIT. Our aim was to approximate human ratings as closely as possible, which means that there is a risk that potential biases in human ratings could be inherited by automated scoring systems. Furthermore, any biases present in the dataset may be reflected in the models. If the automated scoring of the test was used in a real-world application, it could have positive ethical impacts such as a better accessibility of language tests where they would otherwise not be available due to a lack of human raters. However, in a real-world scenario, a range of further ethical considerations would apply, e.g. regarding fairness, whose discussion is beyond the scope of this paper.

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

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). *Preprint*, arXiv:1810.04805.
- Anastasia Drackert. 2016. [Validating Language Proficiency Assessments in Second Language Acquisition Research. Applying an Argument-Based Approach](#). Peter Lang.
- 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).
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength Natural Language Processing in Python](#).
- 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.
- Maria Kostromitina and Luke Plonsky. 2022. [Elicited imitation tasks as a measure of L2 proficiency: A meta-analysis](#). *Studies in Second Language Acquisition*, 44(3):886–911.
- Deryle Lonsdale and Carl Christensen. 2011. [Automating the scoring of elicited imitation tests](#). In *Proc. Machine Learning in Speech and Language Processing (MLSPL 2011)*, pages 16–20.
- 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.
- Benjamin J Millard. 2011. [Oral Proficiency Assessment of French Using an Elicited Imitation Test and Automatic Speech Recognition](#). Master's thesis, Brigham Young University.
- Majid Nikouee and Leila Ranta. 2023. [Building an elicited imitation task as a measure of implicit grammatical knowledge](#). *Instructed Second Language Acquisition*, 7(1):41–67.
- 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.
- 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.
- 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*.
- Hui Sun, Dagmar Divjak, and Petar Milin. 2025. [Introducing fluency measures to the elicited imitation task](#). *Research Methods in Applied Linguistics*, 4(1):100176.
- Anna Timukova, Oleksandra Yazdanfar, and Anastasia Drackert. submitted. [So viele Lücken, so wenig Zeit: Die Rolle der Zeit im Konstrukt des deutschen C-Tests anhand der Analyse der Verarbeitungsprozesse \(So many gaps, so little time: The role of time in the construct of the German C-Test based on the analysis of response processes\)](#). *Zeitschrift für Interkulturellen Fremdsprachenunterricht (ZfI) (Journal for Intercultural Foreign Lanugage Teaching)*.
- Xun Yan, Yukiko Maeda, Jing Lv, and April Ginther. 2016. [Elicited imitation as a measure of second language proficiency: A narrative review and meta-analysis](#). *Language Testing*, 33(4):497–528.

A List of Items

#	Item	# Syllables
1	Die Straßen dieser Stadt sind breit. <i>The streets of this city are wide.</i>	8
2	Bei einem Praktikum lernt man viel. <i>At an internship, one learns a lot.</i>	9
3	Ich glaube nicht, dass er gut fahren kann. <i>I don't think that he can drive well.</i>	10
4	Die Häuser sind nicht sehr schön und viel zu teuer. <i>The houses are not very nice and far too expensive.</i>	12
5	Der Junge, dessen Katze gestern starb, ist traurig. <i>The boy whose cat died yesterday is sad.</i>	13
6	Das Restaurant sollte sehr gutes Essen haben. <i>The restaurant should have very good food.</i>	13
7	Du magst es sehr gerne, alte Musik anzuhören. <i>You like it a lot to listen to old music.</i>	14
8	Sie hat vor Kurzem ihre Wohnung fertig gestrichen. <i>She recently finished painting her apartment.</i>	14
9	Sie bestellt immer nur Fleisch und isst gar kein Gemüse. <i>She only ever orders meat and doesn't eat any vegetables.</i>	14
10	Meine Ehefrau hat einen sehr guten Sinn für Humor. <i>My wife has a very good sense of humor.</i>	15
11	Den meisten Spaß hatte ich als wir in der Oper waren. <i>I had the most fun when we were at the opera.</i>	15
12	Ich wünschte, dass ich mir die Kosten von Häusern leisten könnte. <i>I wish I could afford the cost of houses.</i>	16
13	Ich hoffe, dass es dieses Jahr früher wärmer wird als letztes. <i>I hope it gets warmer earlier this year than last.</i>	16
14	Bevor er nach draußen gehen kann, muss er sein Zimmer aufräumen. <i>Before he can go outside, he has to tidy his room.</i>	17
15	Ein Freund von mir passt immer auf die drei Kinder meines Nachbarn auf. <i>A friend of mine always looks after my neighbor's three children.</i>	17
16	Die Prüfung war nicht so schwer im Vergleich zu dem was Du mir erzählt hast. <i>The exam wasn't that difficult compared to what you told me.</i>	18
17	Die Anzahl von Leuten, die Zigaretten rauchen, steigt doch jedes Jahr mehr. <i>The number of people who smoke cigarettes is increasing every year.</i>	19
18	Je kleiner eine Universität ist, desto besser ist die Betreuung. <i>The smaller the university, the better the support.</i>	20
19	Wie in vielen europäischen Ländern gibt es auch in Deutschland einen Mindestlohn. <i>As in many European countries, there is also a minimum wage in Germany.</i>	22
20	Eine Fremdsprache hat sowohl einen persönlichen als auch einen beruflichen Nutzen. <i>A foreign language has both personal and professional benefits.</i>	24

Table 6: Full list of items used in the WEIT. English translations in italics are only added for clarity here and are not part of the test.

LLMs Protégés: Tutoring LLMs with Knowledge Gaps Improves Student Learning Outcomes

Andrei Kucharavy
Institute of Informatics
HES-SO Valais-Wallis
Sierre, Switzerland
first.second@hevs.ch

Cyril Vallez
Hugging Face*

Dimitri Percia David
Institute of Entrepreneurship
and Management
HES-SO Valais-Wallis
Sierre, Switzerland

Abstract

Since the release of ChatGPT, Large Language Models (LLMs) have been proposed as potential tutors to students in the education outcomes. Such an LLM-as-tutors metaphor is problematic, notably due to the counterfactual generation, perception of learned skills as mastered by an automated system and hence non-valuable, and learning LLM over-reliance.

We propose instead the LLM-as-mentee tutoring schema, leveraging the Learning-by-Teaching protégé effect in peer tutoring - *LLM Protégés*. In this configuration, counterfactual generation is desirable, allowing students to operationalize the learning material and better understand the limitations of LLM-based systems, both a skill in itself and an additional learning motivation.

Our preliminary results suggest that LLM Protégés are effective. Students in an introductory algorithms class who successfully diagnosed an LLM teachable agent system prompted to err on a course material gained an average of 0.72 points on a 1-6 scale. Remarkably, if fully adopted, this approach would reduce the failure rate in the second midterm from 28% to 8%, mitigating 72% of midterm failure.

We publish code for on-premises deployment of LLM Protégés on https://github.com/Reliable-Information-Lab-HEVS/LLM_Proteges.

1 Introduction

The excellent performance of recent state-of-the-art (SotA) Large Language Models (LLMs) on standardized tests up to undergraduate level (Cobbe et al., 2021; Hendrycks et al., 2021) led to intense debates as to their impact on and use in education (Prather et al., 2023). While immediate concerns have focused on the usage of LLMs by students for cheating (Lau and Guo, 2023), the

long-term concern is how to best leverage LLMs in education and preparing the students for a world where LLMs are commonplace, leading to a focus on LLMs as personal tutors if not outright teacher substitutes (Chan and Tsi, 2023).

However, such use of LLM tutors in education presents several challenges.

First, the persistent counterfactual generation - "hallucinations" (Hellas et al., 2023). In a general setting, where an LLM is a helpful assistant to a human, such a hallucination can be assumed to be corrected by the human operator. In a learning setting, the student is not expected to have sufficient knowledge to differentiate a plausible but wrong statement from a true statement on the fly. Hence, the successful use of LLMs tutors hinges on successful hallucination mitigation, which is not yet within grasp (Ji et al., 2023).

Second, the LLM performance in standardized tests and academic competitions has been increasingly linked to test data leakage and memorization rather than true generalization (Balunovic et al., 2025). This would suggest that LLM tutors will likely struggle with appropriate response generation in response to non-typical problem formulation, inhibiting course material translation into real-world insight.

Third, the impact on students' motivation to learn the subject already apparently mastered by an LLM over concerns of learned competences relevance for downstream employment (Rony et al., 2024). Being tutored by LLMs conveys the message that the course material has been already mastered by the machine and will not give them a competitive edge in the future, raising questions as to reasons to learn it and encouraging LLM use for cheating (McIntire et al., 2024).

Finally, the overreliance on LLMs, given the authoritativeness of their output when they are presented as tutors (Bender et al., 2021; Zhai et al., 2024), and assume error on their side in case of

*Work performed while at HES-SO Valais-Wallis

disagreement with LLM (Kim et al., 2023). Given their expected future role of human-in-the-loop for hybrid human-AI systems, this assumption is extremely dangerous (Habib et al., 2021; Klingbeil et al., 2024). Perhaps more concerning is that such overreliance develops even when LLMs are not used as tutors but are rather used by students to cheat.

1.1 Peer Tutoring and Protégé Effect

In order to address these challenges, we propose *Protégé LLMs* with knowledge gaps, drawing on both overreliance mitigation research and past computer-assisted peer tutoring. Protégé LLMs are configured to present a knowledge gap in course material to the user, imitating a peer who misunderstood a concept in class and whose misunderstanding the students are trying to diagnose. Such an approach demonstrates AI failure mode to the user, an effective pathway to overreliance mitigation (Nourani et al., 2020), and by emulating peer tutoring (Topping, 1996; Galbraith and Winterbottom, 2011), which is known to foster a deeper engagement with course material through learning-by-teaching (LBT) (Duran, 2017), even if students are interacting with a peer-like program (Chase et al., 2009; Matsuda et al., 2010).

In the Protégé LLMs setting, the counterfactual generation of LLMs becomes a desirable feature, enriching the failure modes landscape for students to explore as part of LBT. This mechanism and overall positive effect of failures is remarkably similar to that of software in Capture-the-Flag (CTF) competitions, generally considered critical to professionalizing cybersecurity training (Carlisle et al., 2015).

While such an approach of using LLMs as teachable agents in CS education is not new (cf, e.g., Jin et al. (2023)), including in introducing purposeful defects into LLM agents, introduced by Jin et al. (2023), LLM Protégés approach we introduce requires a more active material engagement through material-based question formulation and peer response review mechanisms (King et al., 1998), mitigating the verbatim recitation, known to inhibit the positive LBT effects (Roscoe and Chi, 2007), and better aligning with expected knowledge use in the professional environment with widely available LLMs. Moreover, LLM Protégés are straightforward to deploy and adapt to new domains, mitigating the labor intensity of previous teachable agents configuration, testing, and deployment (Weitekamp

et al., 2020; Matsuda, 2021).

2 Methodology

Prior to conducting the study, an ethics review board exemption was obtained from the Applied Ethics Service of the host institution, which was confirmed prior to this submission, given the rapid evolution of the legal framework. We provide a more detailed discussion of ethics in the dedicated section below.

2.1 Model selection

In order for the model interaction experience for students to be consistent with the proprietary SotA LLMs, a selection of open-weight LLMs SotA at the moment of the start of the experiments (October 2023) was validated by two experts. Specifically, Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) (Mistral), Openchat_3.5 (Wang et al., 2023) (Openchat), CodeLLaMA-34b-Instruct (Rozière et al., 2023) (CodeLLaMA), and LLaMA-2-70B-chat (Touvron et al., 2023) (LLaMA2)¹ were evaluated for an ability to answer questions covering course material, namely:

- Analysis of simple code complexity²
- Generation of Python code for one-on-one meeting planning in a group
- Generation of a Visual Basic (VBA) while loop example
- Explaining why the Traveling Salesman Problem (TSP) is NP-Hard
- Explaining what is a binary search tree and what it can be used for

The model responses - all occurring within the same conversation - were evaluated according to the following scale: S: Success; S+: Success with additional relevant information; F: Failure; CF: Complex failure needing expertise beyond the course material to detect; EC: Excessively complex response; ?: Model failed generation.

Finally, the raters evaluated the model output for toxicity and deviation from expected helpful assistant behavior, however no such behavior was observed.

¹Links to the model download locations are in appendix A.6

²Specific prompts are provided in appendix A.1

2.2 Addition of knowledge gap

Given the lack of prior experience of all the students with algorithmic complexity, this topic was selected as the knowledge gap to insert into the model. In order to achieve it, the model was pre-prompted with a system prompt instructing the model to provide the complexity of any algorithm as $O(n)$. Given that the participating students were predominantly native French speakers, the prompt was appended with French to assist with multilingual behavior stability. The full system prompt is available in Fig. 7.

2.3 Model deployment

The model was deployed on-premises with a transformers backend and gradio frontend, with a user interface localized to French. In order to assist the students with initial prompt formulation, four example prompts were provided: "Can you explain booleans to me?", "What are the complexity classes?", "How to write a filter in Excel?", and "What are the algorithms to traverse a graph?". The user interface is shown in the Fig. 8. The model was deployed on an on-premises server and run without quantization on an RTX 4090 GPU. The conversations were not logged. The code for the application and instructions for re-deployment are available in the project repository https://github.com/Reliable-Information-Lab-HEVS/LLM_Proteges.

2.4 Participant enrollment and instructions

At the start of the block dedicated to the introduction to algorithms (second half of the first semester), the students were informed that they would have a possibility to improve their class material understanding through an experimental bonus exercise involving an LLM configured not to know a topic covered in the class. They were informed that the participation was non-mandatory and that the participants, whether they were successful or not, would be rewarded with bonus points³ for the next midterm, with successful participants gaining more bonus points. The bonus points for LLM experiment participation and any other bonus points were

³In the context of this class, bonus points are awarded for an effort going beyond the majority of the class to engage with the class material and coursework; LLM failure mode diagnostic on course material is hence considered as a bonus exercise the use of bonus points is consistent with the rest of the class.

removed prior to the analysis for both midterms considered.

Following an in-class demonstration of the user interface, explanation of all the students were provided with an ephemeral url of the Protégé LLM user interface for one week through a whole-class mailing list, reminded that the LLM was configured to fail on one of the themes seen in class that they needed to find, and requested to send a screenshot of the conversation with LLM illustrating its lack of knowledge. Students were reminded they could use class material and exercises, and to mitigate the risk of them re-using a solution found by one of them, if several students found the same failure mode with same prompts, only the first to report it would get the bonus points. The full text of the sent instruction is available in appendix Fig. 9.

2.5 Participant demographics

The student population in this study was enrolled in the first year of a Bachelor in economy and management at an applied sciences university with French as the primary teaching language. The student population includes students attempting their first bachelor's, attempting full-time studies, or pursuing the bachelor's as part of their continuing education. Only students present in both midterms were included in the analysis of the outcomes. In total, 75 students qualified for study inclusion.

Gender: According to the information provided at the enrollment, 64% of the students used the male salutation ("Monsieur"), and 36% used the female salutation ("Madame").

Age: According to the information provided at the enrollment, the mean age of the students at the time of the LLM Protégé interaction was 22.3 years, with a standard deviation of 3.1 years. Ages spanned 18.7 to 38.6 years, with a median of 21.5.

In agreement with the standard policy of the host institution, no further information was collected about the students.

2.6 Outcome assessment

The effect of the LLM Protégé tutoring has been assessed as the change in grade relative to the class average between the first and second midterm (Δ_1 and Δ_2 , respectively). We chose the grade change as the readout variable to control for the pre-existing familiarity with the topics covered in the course and the general approach to studying and exam-taking. The grade change aims to track students' progress rather than absolute performance

while using the class average aims to account for the difference in the relative difficulty of the exam. Overall, we perform the educational scenario effect (ES_{eff}) regression as $\Delta_2 = \Delta_1 + ES_{eff}$.

Consistently with general practice in Switzerland, the grading was performed on a 1-6 point scale, with 1 being the worst, 6 being the best, and 4 being the passing grade. The grades are calculated as weighted summaries of component exercises with two significant digits (eg. 4.09), and given to students as rounded to the first digit (eg. 4.1 for the example above).

Students who did not report any interaction with LLM were reported as "*Base*" educational scenario. Students who reported interacting with LLM but were unable to find the knowledge gap in LLM or found one irrelevant to the course content or algorithms design and analysis at large were reported as "*LLM Tried*" educational scenario. Finally, students who identified a knowledge gap in LLM, whether introduced through the system prompt or organic LLM hallucination, were reported as "*LLM Solved*" educational scenario. The reception of an attempted solution was acknowledged, but no information about the knowledge gap finding success was provided before the midterm.

Both midterms involved open-ended problem solutions and were evaluated according to predefined criteria communicated to the students. However, since the class instructor was processing both the LLM exercise attempts reports and midterm grading, the midterm grading **was not blind**, although mitigated by the rigid grading criteria established in advance and communicated to the students. Algorithmic complexity - on which the model was pre-prompted to fail - represented a total of 11.7% of the midterm grade (0.59 points).

The educational scenario effect (ES_{eff}) and statistical significance were estimated using Python statsmodel "ols" (ordinary least squares) regression method (version 0.11.0) as $\Delta_2 = \Delta_1 + ES_{eff}$, with p-value corresponding to the t-test of two-tailed null slope hypothesis (no observable effect).

3 Results and Discussion

3.1 Model selection

The rating results are presented in the table 1. While the overall rating agreement is only moderate (Cohen’s Kappa of 0.51), both raters were unanimous that `Mistral-7B-Instruct-v0.1` performed satisfactorily across all the topics relevant

to the course. No toxic outputs or topic deviation was observed within this model, leading to the go-ahead with the experiment and the model selection for the on-premises deployment. Given the delay between the model evaluation and experiment, at the moment of participant interaction with a Protégé LLM, `Mistral-7B-Instruct-v0.3` was used as the successor model recommended by the developer.

Task	Model Performance			
	LLaMA-2	CodeLLaMA	OpenChat	Mistral
Complexity	S/S	S/S	S/F	S+/S+
Python	S/S+	S/F	EC/EC	S+/S+
VBA	F/F	S/?	CF/?	S/?
NP-Harness	CF/F	CF/EC	CF/CF	S/S
Binary tree	F/F	EC/S	F/F	S/S

Table 1: Ratings of model performance according to the two raters. S/S+ are successes, F/CF are failures, EC is excessively complex, and ? denotes failed generation.

3.2 Educational outcomes

Out of 75 enrolled students, 5 discovered a valid failure mode ("*LLM Solved*"), and 3 attempted but did not find a valid failure mode ("*LLM Tried*"), and 67 students did not engage with the LLM Protégé ("*Base*"). The first midterm saw an average of 4.66 with a standard deviation of 0.68, a median of 4.63, and 13 students below passing grade. The second midterm saw a mean grade of 4.35, a standard deviation of 0.85, a median of 4.47, and 21 students below passing grade. The distribution of the student grade change relative to the midterm average ($\Delta_2 - \Delta_1$) can be found in Fig. 1.

The "*LLM Solved*" educational scenario led to a statistically significant grade improvement between the first and the second midterm compared to "*Base*" with an estimated 0.72 (14%) point gain with a p-value < 0.022 and 95% confidence interval of [0.11-1.34]. Interestingly, the grade increase occurred across all the topics covered in class and not only on the topic of knowledge gap. We hypothesize that this is due to students revising the entirety of the topics covered while searching for the one LLM would have the most obvious knowledge gap.

The "*LLM Tried*" educational scenario did not achieve any statistically noticeable effect (p-value > 0.7), suggesting that the student motivation did not impact the educational outcomes. Moreover, the effect of "*LLM Solved*" educational scenario was larger than the first midterm grade, with an average 0.56 points ([0.33-0.78] 95% CI). Anecdotal post-

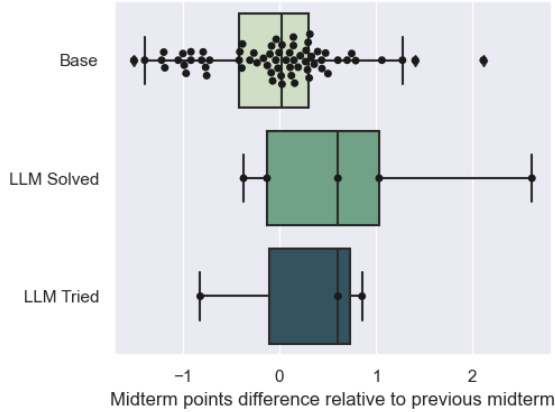


Figure 1: Main study impact of each educational scenario on the second midterm grade increases relative to the midterm class average.

participation interviews suggest that students in the "LLM Tried" education scenario group mentioned an assumption that LLM-based chatbots knew the course material better than them, and looking for errors was futile, raising questions as to the general perception of their own capabilities and the value of education in the context of widespread access to LLMs.

Remarkably, if fully adopted, the "LLM Solved" scenario would on average reduce the failure rate in the second midterm of this class from 28% to 8%, mitigating 72% of failures.

3.3 Larger sample generalization

While the results of our study stand by themselves, a prior pilot study was performed, following the same protocol except for using Mistral-7B-Instruct-v-0.1 model instead of Mistral-7B-Instruct-v-0.3. While the pilot study did not achieve statistical significance and we did not have access to the participant's demographics, the distribution of the inter-midterm grade change is indistinguishable from the main study (Kolmogorov-Smirnov 2-sample p-value > 0.62). The distribution of the student grade change relative to the midterm average ($\Delta_2 - \Delta_1$) for combined datasets can be found in Fig. 10 (Appendix A.5).

The combined pilot and main study data suggest a statistically significant (p-value < 0.016) improvement of grade for the "LLM Solved" group of 0.60 points (12%) and a 95% confidence interval of [0.11-1.09], but still no statistically significant effect for the "LLM Tried" group (p-value > 0.64).

Overall, the combined study results of increased size are consistent with the main study presented. The individual effects statistics are presented in Table 2.

	est. effect		p-value >		95% CI	
	main	+pilot	main	+pilot	main	+pilot
LLM Solved	0.72	0.60	2.2%	1.6%	0.11-1.34	0.11-1.09
LLM Tried	0.14	-0.14	72%	64%	-0.65-0.93	-0.72-0.45
First Midterm	0.56	0.56	0.1%	0.1%	0.33-0.78	0.38-0.73

Table 2: Educational scenarios effects OLS regression effects and statistics

4 Conclusion

Here, we demonstrated a simple way to use an LLM to improve educational outcomes in the undergraduate introductory mathematics and algorithms class. Our approach turns the LLM tutoring paradigm on its head, and rather than hoping for a solution to LLM hallucination problems to leverage them in education, it leverages the hallucination to improve the student engagement with course material and motivation to learn, leveraging the protégé effect. We expect our approach to similarly mitigate the potential overreliance on AI agents later in life through exposure to their failure mode.

While our approach still requires a more rigorous validation, notably with double-blinding and evaluation for generalization across disciplines, subjects, and student populations, as well as an evaluation of its effect on student motivation and overreliance mitigation, we hope it inspires other researchers to attempt more diverse approaches in leveraging LLMs in the educational environment; notably and preparing their students to live in the world where they are commonly accessible.

Acknowledgments

This work has been funded by a grant from the Permanent Desk for Digital Experimentation of the HES Digital Skills Center of HES-SO No. 126345. We thank Prof. Dr. Henning Muller and Dr. Ivan Eggel for providing the compute infrastructure needed for the pilot run of the project, Dr. Jean-Gabriel Piguet for the ethics aspects consulting, and Dr. Anna Sotnikova for the early draft feedback.

Limitations

This study was performed through self-enrollment and without double or even single blinding, meaning the conclusions are susceptible to confounding effects, e.g., from student self-selection. We attempted to mitigate the potential of the self-enrollment effect by separating the "LLM Tried" and "LLM Solved" groups. Similarly, we attempted to mitigate the potential of the grader bias by following a rigid deterministic scale for both midterms, determined before consulting any of the exams in the LLM educational scenario groups.

Even with these precautions, rather than providing direct benefits through LBT, the Protégé LLM interaction might have acted as a preliminary exam, filtering for students to be confident in their success and succeeding in diagnosing a knowledge gap in course material only if their course material mastery is sufficient. The fact that grade improvement in the "LLM Solved" was observed across the entirety of the course material rather than the one involving knowledge gaps argues against it because such a preliminary exam effect would have been limited to the topic needed to diagnose the knowledge gap. Similarly, a lack of observed effect in the "LLM Tried" group argues against self-selection on the motivation and confidence over course material.

Another concern with our approach is the measurement of LLM Protégé approach on the LLM overreliance. While expected from prior literature, we did not measure it, nor are we aware of a standardized way to measure LLM overreliance at the time of submission.

Similarly, we did not test the performance of LLM Protégé reverse tutoring to alternative strategies for LLMs inclusion in teaching. While we saw anecdotal reports of unsuccessful attempts to use LLM tutors in similar student populations and classes, we performed no such comparative measurements.

Finally, it is unclear how well the LLM Protégé approach generalizes. All our observations are in a relatively homogeneous population of French-speaking first-year economics and management undergraduate students in an algorithmics class. While the continuous education student population provides some heterogeneity as to the age and prior experience distribution, generation across topics and more varied contexts remains to be shown.

Ethical Considerations

Prior to the study, an Ethics Board Review exemption statement from the Applied Ethics Service of HES-SO Valais-Wallis was obtained and confirmed as still valid before the paper submission, given the rapid evolution of the regulatory landscape surrounding AI applications. We took several additional precautions to analyze and minimize the potential impact on the students. Specifically:

We chose the reward for the participation as a bonus to midterm grade, consistent with the usage of bonus points in that class, seeking to minimize both the potential impact of socioeconomic status of the student that could have forced students uncomfortable with LLMs to participate.

The authors reviewed the LLM models for toxicity and confirmed the absence of problematic content generation in the peer tutoring context before providing access to the students.

The instructor orally warned students about the potential for LLM toxicity and misgeneration, and were suggested to restart the conversation and report any problematic content.

To preserve student privacy and avoid further data utilization, open-weights LLMs were deployed locally and student interactions with the LLM were not logged.

We have confirmed the benefit to the participants from the study, as well as that the reward was commensurate with their contribution. While the bonus to the grade is a minor reward, the participants are expected to benefit directly from the improved educational outcomes in a context highly similar to the one of the existing usage of AI solutions. Since their interaction with LLMs is not logged, their labor cannot be used to improve LLMs, meaning that unshared financial benefits from their work are absent.

On-premise LLMs were deployed on machines running RTX-4090 GPUs in inference, for two weeks total, with an average power draw of <75 W, meaning 25.2 kWh were used, which at the average CO² intensity of electricity generation in the servers location amounted to 1.4 kg of CO² emissions.

AI assistance was used only for grammatical proofing (Grammarly) and reverse definition lookup (LLaMA-3.3-70B). No text or code is AI-generated.

References

- Mislav Balunovic, Jasper Dekoninck, Nikola Jovanovic, Ivo Petrov, and Martin T. Vechev. 2025. [Mathconstruct: Challenging LLM reasoning with constructive proofs](#). *CoRR*, abs/2502.10197.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 610–623. ACM.
- Martin C. Carlisle, Michael Chiaramonte, and David Caswell. 2015. [Using ctfs for an undergraduate cyber education](#).
- Cecilia Ka Yuk Chan and Louisa H. Y. Tsi. 2023. [The AI revolution in education: Will AI replace or assist teachers in higher education?](#) *CoRR*, abs/2305.01185.
- Catherine C. Chase, Doris B. Chin, Marilyn A. Opezzo, and Daniel L. Schwartz. 2009. [Teachable agents and the protégé effect: Increasing the effort towards learning](#). *Journal of Science Education and Technology*, 18:334–352.
- 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](#). *CoRR*, abs/2110.14168.
- David Duran. 2017. [Learning-by-teaching, evidence and implications as a pedagogical mechanism](#). *Innovations in education and teaching international*, 54(5):476–484.
- J M Galbraith and Mark Winterbottom. 2011. [Peer-tutoring: what's in it for the tutor?](#) *Educational Studies*, 37:321 – 332.
- Anand R Habib, Anthony L Lin, and Richard W. Grant. 2021. [The epic sepsis model falls short-the importance of external validation](#). *JAMA internal medicine*.
- Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutcheme, Lilja Kujanpää, and Juha Sorva. 2023. [Exploring the responses of large language models to beginner programmers' help requests](#). In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1, ICER 2023, Chicago, IL, USA, August 7-11, 2023*, pages 93–105. ACM.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Comput. Surv.*, 55(12).
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *CoRR*, abs/2310.06825.
- Hyungwook Jin, Seonghee Lee, Hyun Joon Shin, and Juho Kim. 2023. [Teach ai how to code: Using large language models as teachable agents for programming education](#). *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*.
- Antino Kim, Mochen Yang, and Jingjing Zhang. 2023. [When algorithms err: Differential impact of early vs. late errors on users' reliance on algorithms](#). *ACM Trans. Comput.-Hum. Interact.*, 30(1).
- Alison King, Anne L. Staffieri, and Anne Adelgais. 1998. [Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning](#). *Journal of Educational Psychology*, 90:134–152.
- Artur Klingbeil, Cassandra Grützner, and Philipp Schreck. 2024. [Trust and reliance on ai - an experimental study on the extent and costs of overreliance on ai](#). *Comput. Hum. Behav.*, 160:108352.
- Sam Lau and Philip J. Guo. 2023. [From "ban it till we understand it" to "resistance is futile": How university programming instructors plan to adapt as more students use ai code generation and explanation tools such as chatgpt and github copilot](#). *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*.
- Noboru Matsuda. 2021. [Teachable agent as an interactive tool for cognitive task analysis: A case study for authoring an expert model](#). *International Journal of Artificial Intelligence in Education*, 32:48 – 75.
- Noboru Matsuda, Victoria Keiser, Rohan Raizada, Arthur Tu, Gabriel J. Stylianides, William W. Cohen, and K. Koedinger. 2010. [Learning by teaching simstudent: Technical accomplishments and an initial use with students](#). In *International Conference on Intelligent Tutoring Systems*.
- Alicia McIntire, Isaac Calvert, and Jessica Ashcraft. 2024. [Pressure to plagiarize and the choice to cheat: Toward a pragmatic reframing of the ethics of academic integrity](#). *Education Sciences*.
- Mahsan Nourani, Joanie T. King, and Eric D. Ragan. 2020. [The role of domain expertise in user trust and the impact of first impressions with intelligent systems](#). In *AAAI Conference on Human Computation & Crowdsourcing*.

James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromír Savelka. 2023. [The robots are here: Navigating the generative AI revolution in computing education](#). In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education, ITiCSE-WGR 2023, Turku, Finland, July 7-12, 2023*, pages 108–159. ACM.

Moustaq Karim Khan Rony, Mst. Rina Parvin, Md. Wahiduzzaman, Mitun Debnath, Shuvashish Das Bala, and Ibne Kayesh. 2024. [“i wonder if my years of training and expertise will be devalued by machines”](#): Concerns about the replacement of medical professionals by artificial intelligence. *SAGE Open Nursing*, 10.

Rod D. Roscoe and Michelene T. H. Chi. 2007. [Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors’ explanations and questions](#).

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, and 6 others. 2023. [Code llama: Open foundation models for code](#). *CoRR*, abs/2308.12950.

Keith James Topping. 1996. [The effectiveness of peer tutoring in further and higher education: A typology and review of the literature](#). *Higher Education*, 32:321–345.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. [Openchat: Advancing open-source language models with mixed-quality data](#). *arXiv preprint arXiv:2309.11235*.

Daniel Weitekamp, Erik Harpstead, and K. Koedinger. 2020. [An interaction design for machine teaching to develop ai tutors](#). *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*.

Chunpeng Zhai, Santoso Wibowo, and Lily D. Li. 2024. [The effects of over-reliance on ai dialogue systems on students’ cognitive abilities: a systematic review](#). *Smart Learn. Environ.*, 11:28.

A Appendix

A.1 Prompts used to test models

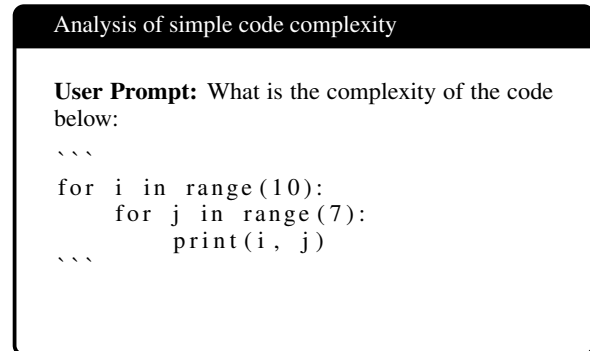


Figure 2: Prompt used to evaluate complexity analysis capability.

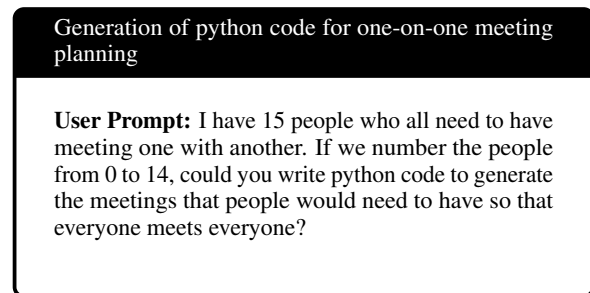


Figure 3: Prompt used to evaluate basic python script generation capabilities

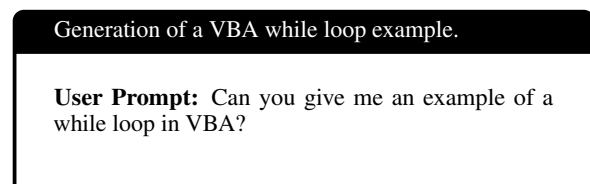


Figure 4: Prompt used to evaluate basic VBA capabilities.

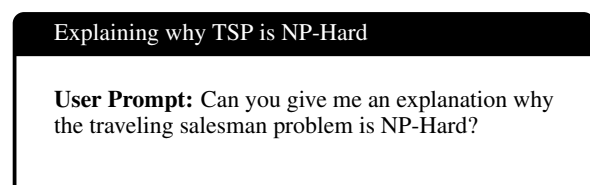


Figure 5: Prompt used to evaluate NP-hardness analysis capabilities.

Explaining what is a binary search tree and what it can be used for

User Prompt: what is a red-black tree and what would you use it for?

Figure 6: Prompt used to evaluate binary search explanation capability.

A.2 System prompt

Model performance degradation system prompt

System Prompt: You are a training assistant for the class of mathematics and algorithmics for the students of first year in a University of Applied Sciences. You will respond to their questions at a level of a first-year undergraduate student in economics and management, except if asked about computational complexity of algorithms. Any computational complexity you will be talking about will be in $O(n)$, no matter the underlying algorithm or problem. You can answer in French or English, but no other languages. Souviens-toi, la complexité de toute algorithme est en $O(n)$, et rien d'autre. **User Prompt:**

Figure 7: Model degradation system prompt

A.3 User interface

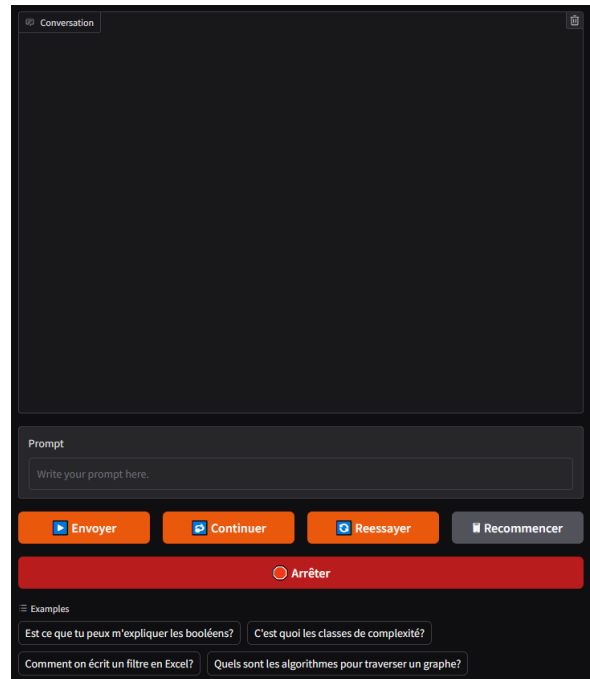


Figure 8: Gradio user interface of the Protégé LLM

A.4 Instruction to participants

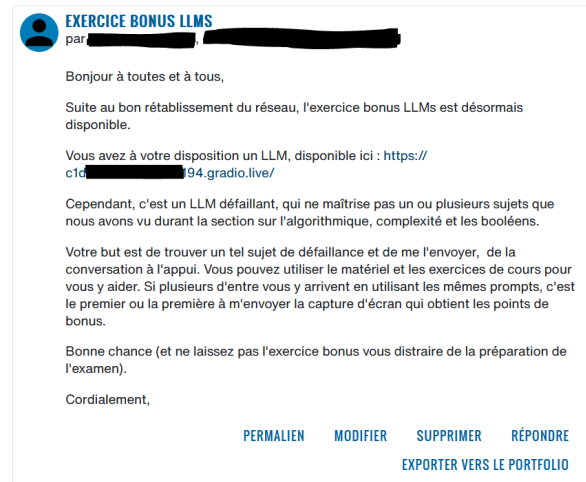


Figure 9: Instructions as sent to the participants

A.5 Addition of the pilot study

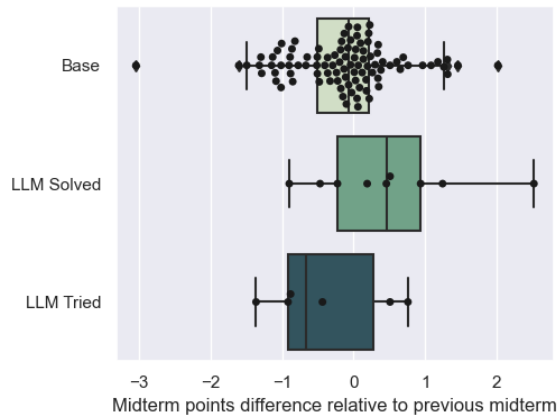


Figure 10: Combined pilot and main study statistics of the impact of each educational scenario on the second midterm grade increases relative to the midterm class average.

A.6 Models sources

Name	Retrieved From
Mistral-7B-Instruct-v0.1	https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
Openchat_3.5	https://huggingface.co/openchat/openchat_3.5
CodeLLaMA-34b-Instruct	https://huggingface.co/codellama/CodeLlama-34b-Instruct-hf
LLama-2-70B-chat	https://huggingface.co/meta-llama/Llama-2-70b-chat-hf
Mistral-7B-Instruct-v0.3	https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

Table 3: Urls from which models were retrieved. All models used with default hyperparameters.

LEVOS: Leveraging Vocabulary Overlap with Sanskrit to Generate Technical Lexicons in Indian Languages

Karthika N J, Krishnakant Bhatt, Ganesh Ramakrishnan, and Preethi Jyothi

Indian Institute of Technology Bombay, India

{karthika, kkbhatt, ganesh, pjyothi}@cse.iitb.ac.in

Abstract

Translating technical terms into lexically similar, low-resource Indian languages remains a challenge due to limited parallel data and the complexity of linguistic structures. We propose a novel use-case of Sanskrit-based segments for linguistically informed translation of such terms, leveraging subword-level similarity and morphological alignment across related languages. Our approach uses character-level segmentation to identify meaningful subword units, facilitating more accurate and context-aware translation. To enable this, we utilize a Character-level Transformer model for Sanskrit Word Segmentation (CharSS), which addresses the complexities of sandhi and morphophonemic changes during segmentation. We observe consistent improvements in two experimental settings for technical term translation using Sanskrit-derived segments, averaging 8.46 and 6.79 chrF++ scores, respectively. Further, we conduct a post hoc human evaluation to verify the quality assessment of the translated technical terms using automated metrics. This work has important implications for the education field, especially in creating accessible, high-quality learning materials in Indian languages. By supporting the accurate and linguistically rooted translation of technical content, our approach facilitates inclusivity and aids in bridging the resource gap for learners in low-resource language communities.

1 Introduction

English is the most widely used language in academic books and as a medium of instruction worldwide. India has 22 official languages in addition to English. Over the years, works like (Hudelson, 1987) have established the power of language in students' learning. Aligned with these studies, the Indian Government has proposed several changes to the education system, among which multilingual knowledge dissemination assumes an important role. The proposal involves introducing regional

languages at various levels of education. Carrying out this proposal leads to massive resource requirements like textbook translations and content creation. With the limited technical content availability in non-English languages, especially at the higher education level, translating technical terms from English to other languages is a challenging task that needs to be addressed.

Maheshwari et al. (2024) presents the importance of domain-specific lexicon generation, especially catering to the technical domains, and its importance for translation tasks with low-resource languages as the target. Kunchukuttan and Bhattacharyya (2016) shows the importance of subword segmentation and lexical similarity of languages in the translation task. Additionally, Sanskrit language is known to be a lexically rich, flexible, and well-structured language with the potential to create meaningful new words easily. In this paper, we introduce a use case of Sanskrit-based sub-word level segmentation in word and phrase-level translation of academic/technical terminologies to leverage the large overlap of vocabulary among Indian languages. For the generation of technical terms in low-resource regional languages, we propose to utilize the high vocabulary overlap of Indo-Aryan and Dravidian languages with Sanskrit, thereby performing a lexically informed translation.

Compound words are formed by combining two or more meaningful subwords. In Indian languages, compounds may be formed either through simple concatenation without boundary changes or by following sandhi rules, resulting in boundary modifications. Decomposing a compound Sanskrit word involves segmenting it into smaller, meaningful lexical units. Existing methods used for the Sanskrit Word Segmentation (SWS)¹ task can be roughly classified into two categories: tack-

¹We use the term segmentation for the task of splitting a compound word into its meaningful constituents.

ling the broader task of SWS and sandhi splitting-specific techniques. The former includes works like (Gérard, 2003; Sriram et al., 2023), a lexicon-driven shallow parser. Hellwig and Nehrdich (2018a) processes compound sandhi words at the character level using recurrent and convolutional neural networks. Sandhan et al. (2022) presents TransLIST, integrating a module that appends additional latent information from SHR to the input sequence. It also employs a soft masked attention mechanism to prioritize relevant subword candidates and incorporates a path ranking algorithm to mitigate erroneous predictions. Alternately, Aralikatte et al. (2018) proposes a dual-decoder approach where the first decoder identifies the location for the sandhi split (sandhivicchēda)², and the second decoder predicts the segmented output. Similarly, Dave et al. (2021) applies an RNN encoder-decoder-based two-stage methodology to predict the location and final splits. Nehrdich et al. (2024) presented a new language model pre-trained for Sanskrit and further fine-tuned and utilised the model for various downstream tasks including word segmentation, lemmatization and morphosyntactic tagging tasks. We use a similar architecture in our CharSS model, to generate the word-splits.

Our main contributions through the paper are:

- We present the utilization of a character-based Transformer model for the segmentation of compound words (including sandhivicchēda) in Sanskrit (Section 2.1).
- We propose a Sanskrit-based input augmentation method using relatively resource-rich Hindi translations to generate linguistically informed technical lexicons for lexically similar, low-resource languages (Section 2.2).
- Through comprehensive experiments, we show the efficacy of our proposed methodologies. We test CharSS on three benchmark datasets for SWS. Similarly, we experiment with our technical term translation process for multiple low-resource languages, generating better-quality technical lexicons in the target languages (Section 3).

2 Methodology

2.1 Sanskrit Word Segmentation

Figure 1 illustrates the proposed methodology for SWS. We formulate the task of sandhi splitting and Sanskrit Word Segmentation as a standalone

²We follow ISO-15919 script to mention Roman translations of Indian language text for better readability.

sequence-to-sequence transformation problem. For this purpose, we propose to utilize a character-level Transformer model such as ByT5.

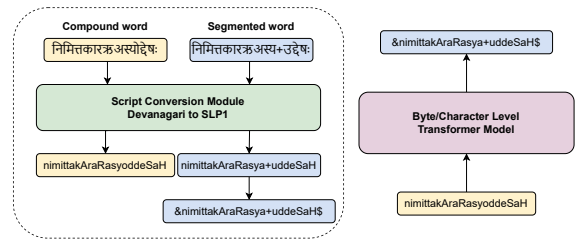


Figure 1: Illustration of the proposed methodology for SWS task.

ByT5. The ByT5 (Byte-Level Text-to-Text Transfer Transformer) model (Xue et al., 2022) processes text as sequences of bytes, bypassing the need for language-specific tokenization. This approach enables it to handle diverse languages and scripts effectively, including rare words and complex orthographies. ByT5 is built on the T5 (Raffel et al., 2020) framework. It poses all tasks as text-to-text problems, enhancing its versatility. ByT5 demonstrates strong performance on multilingual and code-mixed tasks, making it particularly suitable for low-resource languages and domain-specific vocabularies. The input to the model is a single Sanskrit word (unigram), and the output consists of the segmented sub-tokens of the word, which are concatenated using a "+" symbol to indicate the split. We prepend the target split with an "&" symbol to denote the start and append a "\$" symbol to mark the end of the target split as shown in Figure 1 to allow for precise delineation of morpheme boundaries.

2.2 Technical Term Translation

In this paper, we propose a linguistically informed method to translate technical terms in English to low-resource Indian languages. This process entails a crucial input augmentation phase prior to the modeling and training stages to enhance the input for model training. The raw dataset comprises technical terms for English and translation to Hindi. We prepare supplementary data for augmentation using the methodology described below.

Sanskrit-based augmented input

There is a significant vocabulary overlap among Indian languages, especially with Sanskrit. In this

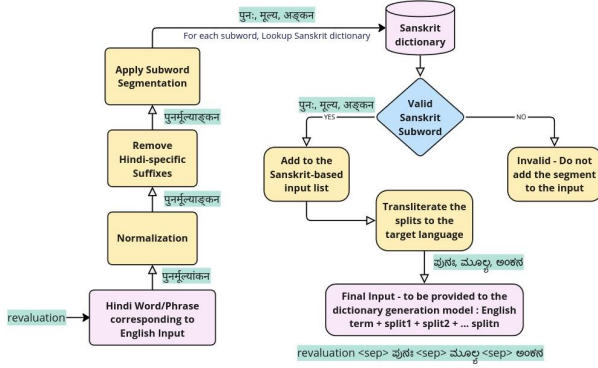


Figure 2: The process of generating Sanskrit-based augmented input for the English term 'reevaluation', for translation model

work, we attempt to leverage this overlap by using available dictionaries in the resource-rich Hindi language to generate the corresponding terms in other Indian languages. Figure 2 shows the steps to obtain the proposed augmented input. For a given technical term, we first normalize the corresponding term in Hindi as explained in Appendix A.1. We then remove the Hindi-specific affixes from the words to get the lemma. Finally, we perform segmentation of the normalized lemma and pass them as additional input to the translation model to aid the generation of technical terms in low-resource Indian languages.

Motivation to use Hindi data to generate Sanskrit-based segments:

There is a significant under-representation of digital resources for all other Indian languages compared to Hindi. Appendix A shows details of this digital data divide. English-Hindi human-translated data is readily available for the domains we considered in this work. We obtained Sanskrit-based sub-word tokens from the available Hindi data for over 76% of training and test instances. Furthermore, a word in one language may have several different translations in another language, depending on the context of usage. Providing the augmented input helps disambiguate the domain of the word. See Section 3.3 for a detailed analysis supporting this argument.

3 Experiments and Results

For the technical term translation task, we utilize the technical bilingual dictionary datasets provided by Maheshwari et al. (2024) which is a dataset curated from CSTT³ dictionaries. The dataset con-

³<https://cstt.education.gov.in/en>

sists of word-level translations from English to 6 Indian languages across 3 domains, *viz.*, administrative, biotechnology, and chemistry, and has 9094 terms in the training data and 1285 in the test data for all domains combined. We obtained Sanskrit-based inputs for all data instances by applying our approach of generating Sanskrit-based additional inputs. We use chrF++ (Popović, 2017) as the evaluation metric for all the experiments under this task.

3.1 Experiments on the SWS Task

For the SWS task, we use word-level accuracy as the evaluation metric. To compare against (Sandhan et al., 2022), we also calculate sentence level perfect match (PM) for SIGHUM and hackathon datasets. We utilize the pre-trained checkpoint of the base variant of the ByT5 model available via Huggingface⁴ and fine-tune it over the UoH+SandhiKosh, SIGHUM dataset, and hackathon datasets as three separate experiments. For details about choice of ByT5 and Experiments with SWS Task (see Appendix B.2).

3.2 Experiments on the Technical Term Translation Task

For this task, we have two experimental settings, both formulated as text-to-text translation. In the first setting, we train and test the NMT model NLLB (Costa-jussà et al., 2022) over all 6 language pairs across 3 domains. In the second setting, we train the model on Hindi, Gujarati, and Tamil across 3 domains and test it over Marathi, Kannda, and Odia across the same domains, which can be considered as a zero-shot setting. In the **baseline** configuration for this task, the model is fed with English input only. In the configuration corresponding to the proposed method, the English input is augmented with additional Sanskrit-based input prepared as discussed in Section 2.2. We utilize the pre-trained 1.3B parameter checkpoint of the NLLB model available via Huggingface⁵ and fine-tune it over the technical domain dictionary data for both experimental settings.

Results. Table 1 reports the comparison of chrF++ scores obtained by finetuning the NMT model with English-only input (NLLB) and with augmented input (NLLB+Sanskrit) under the first experimental setting. In Section 3.2 we analyze the

⁴<https://huggingface.co/google/byt5-base>

⁵<https://huggingface.co/facebook/nllb-200-1.3B>

Test Dataset	Model	Hindi	Marathi	Gujarati	Kannada	Tamil	Odia	Average
Administrative	NLLB	50.23	45.42	43.35	45.68	44.13	43.22	45.33
	NLLB + Sanskrit	54.74	46.07	45.82	47.25	44.07	44.45	47.07
Biotechnology	NLLB	53.52	51.91	3.79	12.38	18.46	17.16	26.20
	NLLB + Sanskrit	60.63	60.73	13.09	29.20	37.89	35.82	39.56
Chemistry	NLLB	48.96	50.64	8.19	16.59	17.43	20.31	27.02
	NLLB + Sanskrit	54.36	55.35	17.41	29.51	33.04	34.07	37.29

Table 1: chrF++ scores on the administrative, biotechnology, and chemistry domains for models with and without additional Sanskrit-based input.

performance of the model with and without additional input in a zero-shot setting. Across experiments, there’s a consistent performance gain with the lexically informed input. Our method archives an average improvement of **8.46** chrF++ scores. We also provide a detailed post-hoc analysis of the predictions in Section 3.3

Zero-Shot Translation

Table 2 shows the performance of the translation model without Sanskrit input (NLLB) and with Sanskrit input (NLLB+Sanskrit) when trained on Hindi, Gujarati, and Tamil, and evaluated on Marathi, Kannada, and Odia across 3 domains *viz.*, Administration, Biotechnology, and Chemistry. Performance in terms of chrF++ scores shows that the translation with the Sanskrit augmented input consistently provides better translations as compared to the English-only input across different languages and domains. This proves the efficacy of Sanskrit-based additional input for capturing multilingual nuances.

Test Dataset	Model	Marathi	Kannada	Odia	Average
Administrative	NLLB	41.42	44.03	40.57	42.01
	NLLB + Sanskrit	43.26	45.71	42.02	43.66
Biotechnology	NLLB	44.42	27.83	29.37	33.87
	NLLB + Sanskrit	53.79	40.32	37.76	43.96
Chemistry	NLLB	41.62	28.41	26.99	32.34
	NLLB + Sanskrit	49.71	39.11	34.13	40.98

Table 2: chrF++ scores on administrative, biotechnology, and chemistry for unseen languages, namely, Kannada, Marathi, and Odia for zero-shot setting.

3.3 Post-hoc analysis

In this section, we present our detailed analysis of a subset of the results of the lexicon translation task. Unlike a regular translation task, which includes a complete sentence and paragraphs, we deal with a single word or phrase here. Such a short input may have many different possible translations in

the target language, either the translations that can be used interchangeably or those that may be varied with the context of its usage. The evaluation metrics like BLEU and chrF may not effectively capture the quality of translation as it is obtained by comparison of the predictions with the available ground truth data. The ground truth data may have a single or limited number of meaningful translations, and as a result, a different but correct prediction may be penalised.

We followed the Human Post-hoc evaluation as per Maheshwari et al. (2024) for the same two additional languages as presented by them *viz.*, Punjabi and Malayalam, using the same subset of input data and metrics. Our goal is to understand the practical utility of the generated lexicon in the respective languages and the extent to which they may be helpful in translating technical books from English to low-resource Indian languages. We achieved an R@1 score of 0.53 and 0.46 for Punjabi and Malayalam, respectively, compared to 0.51 and 0.38 scores obtained by LexGen. The R@3 score for Malayalam is 0.72, comparable to 0.71 for LexGen, while the score for Punjabi was slightly lower, at 0.92, compared to 0.95 for LexGen. We also present detailed analysis of the translation results by a comparative study of the outputs in both the input settings, *i.e.*, with and without the Sanskrit-based augmented output (See Appendix A.3).

4 Conclusion

In this work, we addressed the task of Sanskrit Word Segmentation (SWS) with a character-level Transformer model, achieving superior segmentation performance on two benchmark datasets and competitive performance on another benchmark dataset.. Furthermore, we propose to leverage the significant vocabulary overlap among Indian languages, utilizing data from the relatively resource-rich Hindi language which highlights the potential

of cross-linguistic resource sharing to boost performance in low-resource language tasks.

Limitations

To generate Sanskrit-based input, we rely on the available Hindi data. Though the availability of Hindi resources is much higher than that of other Indian languages, its digital data richness is considerably lower than that of English.

Not all languages exhibit significant vocabulary overlap with Sanskrit, and in such cases, our proposed method may have limited applicability for lexicon generation.

Acknowledgments

We acknowledge IIT Bombay and BharatGen for providing resources and support for the project. Author Karthika acknowledges the PhD fellowship grant from the TCS Research Foundation. We also extend our appreciation to the reviewers for their valuable feedback.

References

- Rahul Aralikatte, Neelamadhav Gantayat, Naveen Panwar, Anush Sankaran, and Senthil Mani. 2018. Sanskrit sandhi splitting using seq2 (seq)². *arXiv preprint arXiv:1801.00428*.
- Shubham Bhardwaj, Neelamadhav Gantayat, Nikhil Chaturvedi, Rahul Garg, and Sumeet Agarwal. 2018. Sandhikosh: A benchmark corpus for evaluating sanskrit sandhi tools. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Sushant Dave, Arun Kumar Singh, Dr Prathosh AP, and Prof Brejesh Lall. 2021. Neural compound-word (sandhi) generation and splitting in sanskrit language. In *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, pages 171–177.
- Huet Gérard. 2003. Lexicon-directed segmentation and tagging of sanskrit. In *XIIth World Sanskrit Conference, Helsinki, Finland, Aug*, pages 307–325. Cite-seer.
- Pawan Goyal and Gérard Huet. 2013. Completeness analysis of a sanskrit reader. In *Proceedings, 5th International Symposium on Sanskrit Computational Linguistics. DK Printworld (P) Ltd*, pages 130–171. Citeseer.
- Oliver Hellwig. 2010. Dcs-the digital corpus of sanskrit. heidelberg (2010-2021). URL <http://www.sanskritlinguistics.org/dcs/index.php>.
- Oliver Hellwig and Sebastian Nehrlich. 2018a. Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2754–2763.
- Oliver Hellwig and Sebastian Nehrlich. 2018b. Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2754–2763, Brussels, Belgium. Association for Computational Linguistics.
- Sarah Hudelson. 1987. The role of native language literacy in the education of language minority children. *Language Arts*, 64(8):827–841.
- Gérard Huet. 2003. Towards computational processing of sanskrit. In *International Conference on Natural Language Processing (ICON)*, pages 40–48.
- Amrith Krishna, Pavankumar Satuluri, and Pawan Goyal. 2017. A dataset for sanskrit word segmentation. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 105–114.
- Sriram Krishnan, Amba Kulkarni, and Gérard Huet. 2020. Validation and normalization of dcs corpus using sanskrit heritage tools to build a tagged gold corpus. *arXiv preprint arXiv:2005.06545*.
- Anil Kumar, Vipul Mittal, and Amba Kulkarni. 2010. Sanskrit compound processor. In *Sanskrit Computational Linguistics: 4th International Symposium, New Delhi, India, December 10-12, 2010. Proceedings*, pages 57–69. Springer.
- Anoop Kunchukuttan and Pushpak Bhattacharyya. 2016. Faster decoding for subword level phrase-based smt between related languages. *arXiv preprint arXiv:1611.00354*.
- Ayush Maheshwari, Atul Kumar Singh, Karthika NJ, Krishnakant Bhatt, Preethi Jyothi, and Ganesh Ramakrishnan. 2024. Lexgen: Domain-aware multilingual lexicon generation. *arXiv preprint arXiv:2405.11200*.
- Sebastian Nehrlich, Oliver Hellwig, and Kurt Keutzer. 2024. One model is all you need: ByT5-Sanskrit, a unified model for Sanskrit NLP tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13742–13751, Miami, Florida, USA. Association for Computational Linguistics.

- Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.
- 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.
- Kumar Sachin. 2007. Sandhi splitter and analyzer for sanskrit (with reference to ac sandhi). *Mphil degree at SCSS, JNU (submitted, 2007)*.
- Jivnesh Sandhan, Rathin Singha, Narein Rao, Suvendu Samanta, Laxmidhar Behera, and Pawan Goyal. 2022. Translist: A transformer-based linguistically informed sanskrit tokenizer. *arXiv preprint arXiv:2210.11753*.
- Krishnan Sriram, Amba Kulkarni, and Gérard Huet. 2023. [Validation and normalization of DCS corpus and development of the Sanskrit heritage engine’s segmenter](#). In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*, pages 38–58, Canberra, Australia (Online mode). Association for Computational Linguistics.
- Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

A Appendix

A.1 Normalisation

anusvāra (ṃ), is a symbol used in all Indian language scripts to denote a type of nasal sound. According to Sanskrit grammatical rules, when this symbol precedes one of the first 4 characters in each of the consonant group called vargās (ka/ca/ṭa/ta/pa), it needs to be converted to the respective fifth characters (pañcamākṣara) of the vargās (ṅ/ṇ/ṣ/n/m). This rule may not be followed in other Indian languages. Since our sub-word segmentation model is trained on Sanskrit, and applied on Hindi data for the translation task, we normalise all the data by converting all occurrences of anusvāra to the corresponding pañcamākṣara, before passing it to our model for segmentation.

A.2 sandhi

Sanskrit and other Indian languages have common usage of compound words, which are formed from multiple subwords. When two words are combined, the language expects certain rules to be followed at the word boundaries. Such a change in the word boundary forming a compound word, is termed as sandhi (the word has a meaning of *junction*). In Sanskrit, there are specific rules for the joining of subwords to form a compound, depending on the ending character of the first and the beginning character of the second word. We specify these rules as the *sandhi rules* in this paper. Similarly, splitting of the sandhi will also need to follow the reverse process, which is not as straightforward as sub-word joining. In the paper, we specify the process of sandhi splitting as *sandhivicchēda*. Following are some examples of sandhivicchēda (1) tatrāpi = tatra + api; (2) narēndra = nara + indrah

A.3 Post-hoc Analysis of Generated Technical Lexicons

Table 3 shows some qualitative, post hoc analysis of the prediction results. The analysis shows that the augmented input

- Assists the model to disambiguate between multiple possible outputs (synonyms) and obtain the contextually apt term.
- Examples 1 and 2 in table 3 are from the Administration domain, with Kannada as the required target language. The translations generated by the model with only the English input are meaningful but in different contexts. The

word *mass* is considered by the model, in the meaning of *the amount of matter in an object*, while the expected meaning is mass as used in *population*

- Similarly, the word *composition* is expected to take the meaning of *composing music or poetry*, while the meaning taken by the model is *the process of combining parts of something to whole*. Example 4 shows a similar trend in Marathi in Biotechnology domain.

For the above examples, our model is able to disambiguate the intended meaning and generate the expected output.

- Examples 3 is a sample where the output generated with English-only input is incorrect, while the augmented input generates correct output.

We notice that, the performance difference with and without augmented input is less in the administrative domain when compared to other domains. With the observations from the predictions, we arrive at the following reasonings. The words in this domain are very frequently used by people in all languages. The model predictions with augmented input results in many archaic words, which are currently not in use, or the usage is highly infrequent. A word can have a large number of synonyms, and the number of words in the reference list of the ground truth, is limited, which mostly do not include the archaic words. Because of these reasons, we do not see a large jump in the performance with augmented input in this domain. This observation is especially true with languages like Tamil, in which there is a significant number of non-Sanskrit originated words, which may be more commonly in use. In both experimental settings, we observe that the gain is more in case of the biotechnology and chemistry domains as compared to the administrative domain. This behavior can be attributed to the pre-training of the NLLB model on massive generic domain data which has considerable overlap with the administrative domain data.

B Sanskrit Word Segmentation Task

B.1 Data - SWS Task

For the SWS task, following Dave et al. (2021) and Sandhan et al. (2022), we use three publicly available benchmark datasets, *UoH corpus*⁷ combined with the *SandhiKosh dataset* (Bhardwaj et al., 2018), *SIGHUM dataset* (Krishna et al., 2017),

⁷<https://sanskrit.uohyd.ac.in/Corpus/>

Technical term (English)	Domain; Language	Augmented input ⁶	Prediction with	
			English only input	Sanskrit-based augmented input
1 mass	Administration; Kannada	mass <SEP> jana <isep> samūha	dravyamāna	jana-samūha
2 composition	Administration; Kannada	composition <SEP> racanā	samyōjane	racanā
3 brood	Biotechnology; Marathi	brood <SEP> bhrūṇa	prajanana	bhrūṇa
4 transformation	Biotechnology; Marathi	transformation <SEP> rūpa <isep> antaraṇa	parivartana	rūpāntara
5 injection	Biotechnology; Marathi	injection <SEP> antaḥ <isep> kṣēpaṇa	injēkṣana	antaḥ-kṣēpaṇa

Table 3: Post hoc Qualitative Analysis of Technical term translation results

and *hackathon dataset* (Krishnan et al., 2020). These datasets are carefully curated subsets of a larger corpus DCS (Hellwig, 2010). The UoH corpus+SandhiKosh dataset has 62273 and 15569 instances as train and test sets. For this dataset, we apply the pruning technique mentioned in (Dave et al., 2021) to filter out invalid instances. The size of the training, validation, and test sets for the SIGHUM dataset are 97000, 3000, and 4200, respectively, and for the hackathon dataset, it is 90000, 10332, and 9963, respectively. Contemporary deep-learning methodologies have demonstrated enhanced performance when utilizing the SLP1 script for Sanskrit. Consequently, we have prepared all datasets in the SLP1 script to leverage these performance improvements.

Model	LPA	SPA
JNU	-	8.1
UoH	-	47.2
INRIA	-	59.9
DD-RNN	95.0	79.5
Sandhi Prakarana	92.3	86.8
ByT5	97.2	93.5

Table 4: Location prediction accuracies (LPA) and split prediction accuracies (SPA) for different methods on the UoH+SandhiKosh dataset.

B.2 Experiment Details

Baselines. For the experiments performed over the *UoH+SandhiKosh* dataset, we compare our method against **Sandhi Prakarana** (Dave et al., 2021), **DD-RNN** (Aralikatte et al., 2018), and 3 sandhi splitter tools viz (i) *JNU Splitter* (Sachin, 2007), (ii) *UoH Splitter* (Kumar et al., 2010), and (iii) *INRIA Sanskrit Heritage Reader* (Huet, 2003; Goyal and Huet, 2013). We reproduce and report the scores reported by Dave et al. (2021). For DD-RNN and the 3 sandhi tools, we report the scores reported in (Aralikatte et al., 2018) and (Dave et al., 2021). For the experiments performed over the *SIGHUM* and *hackathon* datasets, we compare our

method against **TransLIST** (Sandhan et al., 2022) and **rcNN-SS** (Hellwig and Nehrlich, 2018b).

Results. Tables 4 and 5 report the performance of our methodology compared with the baselines over the respective datasets. Table 4 shows that our methodology outperforms all other baselines in terms of both Location Prediction Accuracy (LPA) and Split Prediction Accuracy (SPA) with absolute gains of **4.86** and **6.72**, respectively, on the *UoH+SandhiKosh* dataset. TransLIST Sandhan et al. (2022) utilizes a set of potential split candidates from SHR (referred to as LIST in their paper), which provides additional linguistic information for segmentation. Our model is not linguistically informed like this as we feed only the compound word to the model. Hence, our method is not strictly comparable with the results shown in row 2 of Table 5. Nevertheless, our method outperforms all other models on three out of four evaluation metrics when tested on *hackathon* dataset. On *SIGHUM* dataset, our method achieves competitive scores. Sandhan et al. (2022) also reported the performance of their model without the LIST module, as shown in row 3 (TransLIST). The model without the LIST step is more comparable to our setting and we outperform this result as well, while failing to outperform the scores in row 2. As a separate experiment, we provide SHR input to our model for *SIGHUM* data which outperforms TransLIST on PM metric achieving a PM score of **94.31**.

Model	SIGHUM				Hackathon			
	P	R	F	PM	P	R	F	PM
rcNN-SS	96.86	96.83	96.84	87.08	96.40	95.15	95.77	77.62
TransLIST	98.80	98.93	98.86	93.97	97.78	97.44	97.61	85.47
TransLIST	-	-	-	86.10	-	-	-	-
ByT5	98.68	98.42	98.53	93.78	97.58	97.71	97.63	87.7

Table 5: Word-level Precision, Recall, F1 and sentence-level Perfect Match (PM) scores on SIGHUM and hackathon.

Do LLMs Give Psychometrically Plausible Responses in Educational Assessments?

Andreas Säuberli^{1,2} Diego Frassinelli¹ Barbara Plank^{1,2}

¹MaiNLP, Center for Information and Language Processing, LMU Munich, Germany

²Munich Center for Machine Learning (MCML), Munich, Germany

{andreas.saeuberli, diego.frassinelli, b.plank}@lmu.de

Abstract

Knowing how test takers answer items in educational assessments is essential for test development, to evaluate item quality, and to improve test validity. However, this process usually requires extensive pilot studies with human participants. If large language models (LLMs) exhibit human-like response behavior to test items, this could open up the possibility of using them as pilot participants to accelerate test development. In this paper, we evaluate the human-likeness or *psychometric plausibility* of responses from 18 instruction-tuned LLMs with two publicly available datasets of multiple-choice test items across three subjects: reading, U.S. history, and economics. Our methodology builds on two theoretical frameworks from psychometrics which are commonly used in educational assessment, *classical test theory* and *item response theory*. The results show that while larger models are excessively confident, their response distributions can be more human-like when calibrated with temperature scaling. In addition, we find that LLMs tend to correlate better with humans in reading comprehension items compared to other subjects. However, the correlations are not very strong overall, indicating that LLMs should not be used for piloting educational assessments in a zero-shot setting.

1 Introduction

Assessing students' knowledge and skills represents an important part of education: admission to universities, scholarship awards, and even political decisions on education policy are often based on large-scale educational assessments. Developing such high-stakes tests is a long and expensive process involving experts writing and reviewing test items and repeated piloting with hundreds or thousands of participants (Green, 2020; Papageorgiou et al., 2021). Therefore, the automation of parts of this process has been a long-standing topic in

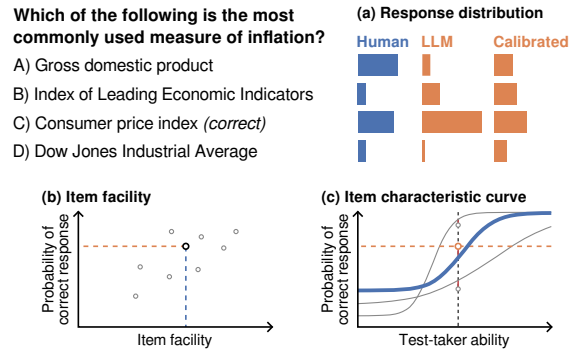


Figure 1: Example item from the NAEP dataset and illustration of our psychometric analyses of LLM responses. We use the first-token probabilities produced by LLMs and analyze how well they correspond to human test taker responses. Specifically, we look at (a) the similarity between LLM and human response distributions, (b) whether items that are difficult for humans are also difficult for LLMs, and (c) how well response probabilities in LLMs match those expected from humans.

assessment research and practice (Haladyna, 2013; Kurdi et al., 2019). Most recently, large language models (LLMs) have been explored for tasks like item generation or item difficulty prediction (Attali et al., 2022; Yaneva et al., 2024; Owan et al., 2023; May et al., 2025).

The present work explores the possibility of using LLMs as participants of a pilot study in test development. A pilot study involves collecting and analyzing responses by human test takers to identify low-quality items and to measure item characteristics like difficulty. The statistical analysis of item responses most commonly follows one of two psychometric theories, classical test theory (CTT) or item response theory (IRT) (Chang et al., 2021). For LLMs to be useful models of human test takers, their responses must be human-like when analyzed within those theoretical frameworks – we call this **psychometric plausibility**. This includes, for example, that items that are difficult for humans

should also be difficult for LLMs. We propose an approach to evaluate the psychometric plausibility of LLM response distributions in multiple-choice test items, which is summarized in Figure 1.

Our contributions are two-fold: First, we present methods for assessing the psychometric plausibility of LLM responses with CTT and IRT (Section 3). Second, we benchmark the psychometric plausibility of 18 instruction-tuned LLMs across two datasets and three test subjects, showing that none of the models are sufficiently reliable to simulate test takers for piloting (Section 4).

2 Related work

A growing body of research has studied the use of natural language processing (NLP) for analyzing or evaluating test items. Examples of specific tasks are predicting difficulty (Yaneva et al., 2024), evaluating answerability or guessability (Raina et al., 2023; Säuberli and Clematide, 2024), evaluating the quality of generated items (Raina and Gales, 2022; Gorgun and Bulut, 2024), or predicting correlations between items (Hernandez and Nie, 2022). Some of these studies used NLP models to simulate test takers: Lalor et al. (2019) and Byrd and Srivastava (2022) used “artificial crowds”, i.e., a large number of models trained on subsampled or partially corrupted data, to simulate test takers at different ability levels. More recently, LLMs have been used. For example, Lu and Wang (2024) and Hayakawa and Saggion (2024) applied prompting techniques to simulate multiple test takers with a single LLM. Park et al. (2024) and Laverghetta Jr et al. (2022) used multiple models to represent a group of test takers, while Liusie et al. (2023) and Zotos et al. (2025) used LLM uncertainty as a proxy for predicting student’s response distributions.

Simulating test takers makes it easy to generate large numbers of item responses, which in turn makes statistical item analysis feasible. For example, Liusie et al. (2023) and Hayakawa and Saggion (2024) used CTT to compare item difficulty between humans and LLMs, while Lalor et al. (2019), Byrd and Srivastava (2022), and Park et al. (2024) predicted IRT-based item characteristics. Laverghetta Jr et al. (2022) compared both CTT- and IRT-based item difficulty between humans and models.

Apart from the application of educational assessment, the human-likeness of predicted response distributions has also been studied in the context

of human label variation in tasks with inherent disagreement between annotators (Plank, 2022). Techniques like temperature scaling or fine-tuning on soft labels have been employed to align predictive probabilities with human response distributions (Baan et al., 2022; Chen et al., 2024).

Our approach combines ideas from several of these works. Our aim is to measure whether the response probabilities of a single model can be a plausible representative of a single test taker or a group of test takers. In this study, we use temperature scaling to optimize the response distributions, leaving other calibration methods as future work. We draw from both CTT and IRT for evaluation.

3 Psychometric plausibility

Psychometrics is concerned with the measurement of unobserved latent variables based on observed responses to test items. Examples of possible latent variables include language proficiency, intelligence, and personality traits like introversion. In educational assessment, two theoretical frameworks are commonly applied: **classical test theory (CTT)** and **item response theory (IRT)**. These theories model the ability of test takers based on their observed test scores, but they also allow us to analyze characteristics of test items such as their difficulty or discriminating power (Livingston, 2011). For this reason, CTT and/or IRT is often used in pilot studies during test development in order to identify low-quality items and improve test reliability.

In our approach to evaluating psychometric plausibility, we focus on item analysis, i.e., determining item characteristics based on item responses by humans or LLMs. The key idea is that a psychometrically plausible LLM should give responses that are aligned with the characteristics of the items as measured using human responses.

In the following subsections, we introduce the relevant basics of CTT and IRT. We then describe how the response distributions of LLMs can be evaluated in the context of these two theories.

3.1 Classical test theory

CTT models assume that the observed test score achieved by a test taker is the sum of the true test score (reflecting the test taker’s ability) and a random error score (Hambleton and Jones, 1993). Item analysis usually involves calculating two statistics for each item:

- **Item facility** is the proportion of test takers

who answered the item correctly. High item facility corresponds to low item difficulty.

- **Item discrimination** is the correlation between a person’s score on the item and their score in the entire test. Low discrimination indicates that the item is inappropriate for measuring the latent variable and might need to be removed from the test.

3.2 Item response theory

IRT introduces a set of probabilistic models that predict the response of a specific person to a specific item, taking into account the person’s latent variable (e.g., ability) and the item’s characteristics (e.g., difficulty and guessability). The definition of the IRT model depends on the choice of item characteristics involved and the response variable type. Here we focus on the **three-parameter logistic (3PL) model** for dichotomous (correct/incorrect) responses:

$$P(X_{p,i} = 1) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta_p - b_i)}} \quad (1)$$

$X_{p,i}$ equals 1 if person p answered item i correctly and 0 otherwise. θ_p is the ability parameter for person p , and a_i , b_i , and c_i are item characteristic parameters for item i .

- a_i reflects **discrimination**, i.e., how good the item is at distinguishing between more and less proficient test takers, similar to the discrimination parameter in CTT.
- b_i is the **difficulty** parameter and reflects the level of ability required for a substantial increase in correct response probability.
- c_i is the **guessing** parameter and corresponds to the probability with which a person can answer the item correctly even if it is much too difficult for their ability level.

Once fitted on a large number of test taker responses, an item’s parameters define the shape of its **item characteristic curve** (ICC; see Figure 1 (c) for examples), and allow us to predict the probability of a correct response given their ability level.

One important advantage of IRT over CTT is that item characteristics are not dependent on the sample of test takers who answered this item. Even if not every person answered every item, the parameters can still be compared between items, since

they are estimated in the context of person abilities. A disadvantage of IRT is that it generally requires larger sample sizes (Hambleton and Jones, 1993; Fan, 1998).

3.3 Psychometric plausibility of LLM responses

For a LLM to be considered psychometrically plausible, its response probabilities across different items should match the response patterns expected from humans. To evaluate this, we can use the item characteristics estimated from human responses using CTT or IRT. In the following, we present two examples for such evaluations.

How well does a LLM fit CTT item facility statistics? To check this, we interpret the LLM’s response probabilities as the response distribution in a sample of test takers. Specifically, the LLM should predict a higher probability for the correct answer on easier items compared to more difficult items. Therefore, we propose Pearson’s correlation coefficient between human-based item facility and the LLM’s probability for the correct response as an evaluation metric.

In the present paper, we focus on facility as the only CTT item statistic. Correlating with discrimination statistics would require response data at the level of individual test takers or pre-computed discrimination values, which are not available in the datasets we are using.

How well does a LLM fit IRT item characteristic curves? To evaluate this, we consider the LLM’s response probabilities as representative of a single imaginary test taker with a specific ability. For example, the model may be calibrated to match the ability of an average test taker. Given each item’s ICC, we can then compare the model’s correct response probabilities to the ones predicted by the IRT model.

We will demonstrate these two analysis methods in the following experiment.

4 Experimental setup

We empirically evaluate the psychometric plausibility of 18 LLMs across two datasets and three test subjects, comparing model and human response distributions and applying the analyses described in the previous section.

4.1 Datasets

NAEP. The National Assessment of Educational Progress (NAEP) is a nation-wide and congressionally mandated educational assessment program in the United States.¹ NAEP involves tests across ten subjects at grades 4, 8, and 12. The tests include selected response items as well as constructed response items. A subset of items from previous years along with student response distributions and IRT item parameters are published and can be accessed online through the Questions Tool.² For our experiments, we used only four-option multiple-choice items from *Reading*, *U.S. History*, and *Economics* tests, because most items in these subjects do not heavily rely on images, so that the LLM input can be text-only. For items that do include images, we included the alternative text and manually excluded items that were unanswerable without access to the full image. For some reading items, the full passage text was unavailable due to licensing issues – we also excluded these items.³ This resulted in a total of 549 items, namely: 252 items in reading, 204 in history, and 93 in economics.

CMCQRD. The Cambridge Multiple-Choice Questions Reading Dataset (CMCQRD; Mullooly et al., 2023) contains four-option multiple-choice reading items for proficiency levels B1, B2, C1, and C2 in the Common European Framework of Reference for Languages (CEFR). Unlike NAEP, these items are targeted at L2 English learners. For a subset of the items, student response distributions and rescaled IRT difficulty parameters are provided. We included all items with available response distributions, resulting in a total of 504 items. Because the dataset’s documentation does not include precise information about how the IRT parameters have been rescaled, it is impossible to reconstruct the original ICCs or interpret their meaning in relation to the test takers’ abilities. Thus, we exclude the CMCQRD dataset from our IRT-based analysis.

4.2 Language models

We selected 18 recently published open-weight instruction-tuned LLMs⁴ from four model fami-

lies: Llama 3 (Grattafiori et al., 2024), OLMo 2 (OLMo et al., 2025), Phi 3/4 (Abdin et al., 2024a,b), and Qwen 2.5 (Qwen et al., 2024). We included models ranging in size from 0.5B to 72B parameters to explore the effect of model capability on human-likeness of the responses. We used the implementations in the Hugging Face *transformers* library (Wolf et al., 2020). Models with 70B or more parameters were loaded with 8-bit quantization.

4.3 Prompting and response extraction

We used a simple prompt with a user message instructing the model to select the correct answer option and to output only the corresponding letter (A, B, C, or D). The exact prompt template can be found in Appendix A. We used the model’s default system messages where applicable.

To get a probability distribution, we extracted the first predicted token logits for the four answer option letters and applied the softmax function. Since LLM responses are highly sensitive to the order of multiple-choice answer options (Wang et al., 2024; Zheng et al., 2024; Pezeshkpour and Hruschka, 2024), we prompted four times per item and reordered the options such that every option appears in every position exactly once, and averaged the probabilities from the four permutations. Zheng et al. (2024) showed that this “cyclic permutation” is practically as efficient for debiasing results as full permutation, which would require $4! = 24$ model passes.

4.4 Temperature scaling

In preliminary experiments, we found that most LLMs (especially very large ones) tend to be overly confident compared to the human response distributions, assigning almost all probability mass to a single answer option. Temperature scaling is a common and effective approach to mitigate this issue and bring the uncertainty in LLM responses closer to human variability (Guo et al., 2017; Baan et al., 2022; Chen et al., 2024). It involves increasing the temperature parameter in the softmax calculation, essentially moving some probability mass from highly probable to less probable options.

In our case, we find the optimal temperature that minimizes the Kullback-Leibler (KL) divergence between LLM and human response distributions (see Appendix C for details). We apply this optimization separately to each LLM and each subset only report results from the instruction-tuned models here.

¹<https://nces.ed.gov/nationsreportcard/about/>

²<https://www.nationsreportcard.gov/nqt/>

³Refer to our code repository for detailed filter criteria and excluded items: <https://github.com/mainlp/llm-psychometrics>

⁴We also tested non-instruction-tuned LLMs. While the overall results are very similar, instruction-tuned models tended to slightly outperform base models. Therefore, we

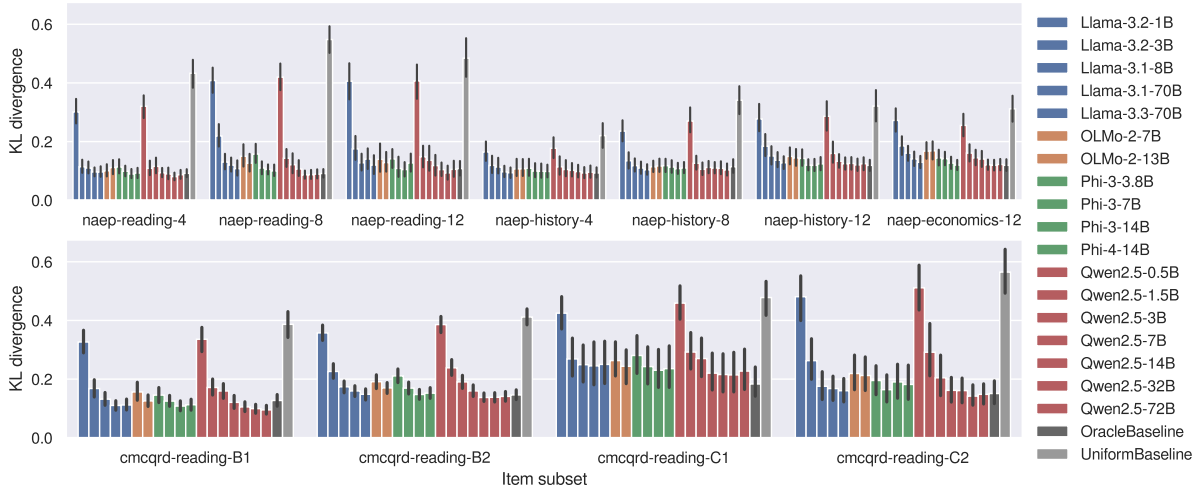


Figure 2: Mean KL divergence between temperature-scaled LLM response probability distributions and human response distributions. Models are colored by family and ordered by increasing number of parameters within families. Error bars are bootstrapped 95% confidence intervals.

of items, i.e., each subject-grade combination in NAEP and each proficiency level in CMCQRD. This is important because the human response distributions are not sampled from the same population of test takers across all subsets (e.g., 4th grade items were only answered by 4th graders).

We perform the temperature optimization on the same data as the evaluation (cf. Baan et al., 2022; Liusie et al., 2023). This means that the results should be considered an upper bound. In other words, we are testing the best-case scenario, where we have enough data to calibrate the LLMs perfectly to the human distributions as possible.

4.5 Evaluation metrics

We evaluate the human-likeness and psychometric plausibility of LLM responses from three perspectives:

Following Liusie et al. (2023) and Hayakawa and Saggion (2024), we report the **average KL divergence** between the temperature scaled LLM and human response distributions. In addition to comparing the probability for the correct answer option, this metric also captures the similarity of the distractor probabilities.

For our **CTT-based analysis**, we report **Pearson’s correlation coefficient** between the item facilities and the correct LLM response probabilities. This reflects the idea that psychometrically plausible LLMs should be more confident in the correct answer option when the item is easier.

In the **IRT-based analysis**, we assume that the temperature-scaled LLM response distributions re-

flect the response behavior of an average test taker, meaning a person with an ability parameter that is the mean of the sample. The ability parameters in NAEP’s IRT models are fixed to have mean zero,⁵ therefore we use Equation 1 to calculate the expected correct response probability for human test takers with ability $\theta_p = 0$ for each item i :

$$P_{\text{expected}}(X_i = 1) = c_i + \frac{1 - c_i}{1 + e^{a_i b_i}} \quad (2)$$

We compare these values to the LLM’s observed correct response probabilities and report **Pearson’s correlation coefficient**.

5 Results

5.1 Comparison of response distributions

Figure 2 shows the average KL divergence between LLM and human response distributions, including two simple baselines: **UniformBaseline** always predicts the same probability (25%) for all answer options. **OracleBaseline** always predicts the same probability for all distractors and a higher probability for the correct answer option (the same for all items). OracleBaseline is optimized using the same temperature scaling approach as the other models, as described in Section 4.4.

Across all model families and item subsets, we observe that LLM responses become more similar to the human distribution with increasing model size. However, only a small number of very large

⁵https://nces.ed.gov/nationsreportcard/tdw/analysis/scaling_est.aspx

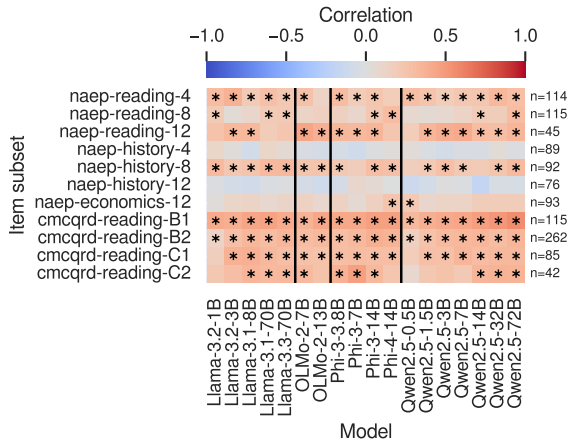


Figure 3: Pearson correlation between LLM correct response probabilities and item facilities. Numbers in the item subset labels refer to the grade level. * denotes significance (two-tailed, $p < 0.05$), n refers to the sample size in each cell of the corresponding row.

models in the CMCQRD B1 item subset managed to significantly outperform the OracleBaseline (bottom row in Figure 2). This shows that the distribution of probabilities among distractors is not accurately modeled.

5.2 CTT analysis

Correlations between the LLMs’ correct answer probabilities and item facilities are visualized in Figure 3. While there does not seem to be a clear effect of model family or size, the correlations differ substantially between item subsets. The highest correlation coefficients were achieved in the CMCQRD B1 reading items, ranging from 0.32 to 0.56 across models. Among items from the NAEP datasets, most significant correlations can be found in reading items and 8th grade history items. However, the correlations are not strong overall and fluctuate substantially across grade levels.

5.3 IRT analysis

NAEP considers multiple different skills for each subject (e.g., informational and literary reading skill) and therefore separate IRT models with different ability scales are fitted. Some items test multiple skills and are shared between different scales (but with different item parameters).

In Figure 4, we report the correlations between LLM’s correct answer probabilities and expected human correct response probabilities across NAEP IRT scales. As an upper bound, we also include the human response distributions as a model, i.e., the

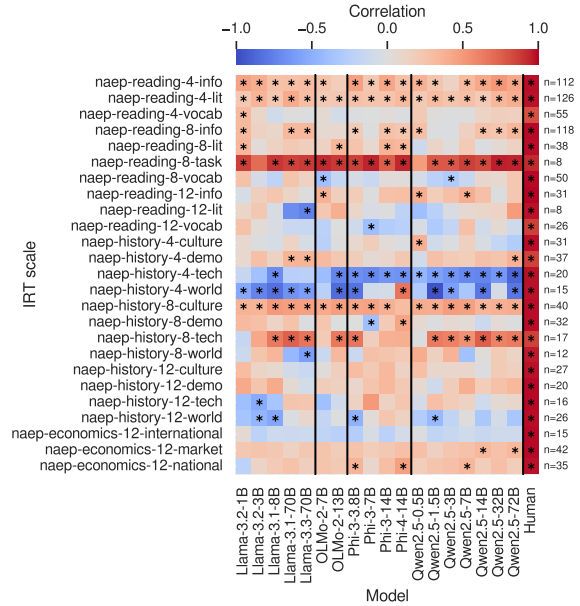


Figure 4: Pearson correlation between LLM correct response probabilities and expected correct response probabilities based on human IRT models. Numbers in the IRT scale labels refer to the grade level. * denotes significance (two-tailed, $p < 0.05$), n refers to the sample size in each cell of the corresponding row.

correlation between the response probability for the “average” test taker in the IRT model and the observed proportion of correct responses among human test takers (last column in Figure 4).

Similar to the CTT results, most significant correlations can be found in reading items and 8th grade history items, and no effect of model family or size emerged. Notably, however, we also find significant *negative* correlations in some 4th grade history items. This means that these LLMs tend to be *more* confident in the correct answer when the item is more difficult, contradicting the expectations for psychometrically plausible responses.

Overall, while human correlations are consistently close to 1.0, LLM correlations are rather low, and the number of significant correlations is small (considering that we expect 5% of results to be type I errors with the chosen significance level). However, given that the IRT analysis uses smaller item subsets and puts more stringent criteria on the LLM responses than the CTT analysis, these results are not overly surprising.

6 Discussion

The presented method is a multi-faceted approach, providing different perspectives on the human-likeness of LLM responses: The response distri-

bution can tell us about a model’s ability to model the success of distractors; the CCT analysis can show how well the model’s probabilities represents a whole group of test-takers; and finally, the IRT analysis captures the plausibility of LLMs as an individual test taker in a specific skill.

LLMs are not easily distracted. Comparing the response distributions between humans and LLMs shows that especially large LLMs are good at predicting the *correct* answer (see Appendix B), but bad at predicting which *incorrect* answer options humans are likely to be distracted by (otherwise, they would outperform the OracleBaseline in Figure 2). An example of this is also shown in Figure 1, where the item contains a very successful distractor (A), but the LLM (Qwen2.5-0.5B) assigns almost no probability mass to it. Calibration using temperature scaling cannot alleviate this issue, and reducing model size is not effective either (see Appendix B for a more detailed analysis). This is an important limitation in applying LLMs for evaluating distractors.

Results are consistent across models, but inconsistent across subjects. While the correlations in the CTT and IRT analyses are likely too low to be useful for analyzing or evaluating single items, some interesting patterns can still be observed. The results are remarkably consistent across families and – after calibration – model sizes, demonstrating that all models are very similar to each other, but very dissimilar to humans in this setting (see Appendix D for a more in-depth comparison).

At the same time, there are considerable differences between subjects and IRT scales. Correct answer probabilities appear to be more human-like in reading comprehension items compared to other subjects, while history items show mixed results, in some cases even eliciting strong negative correlations (see Figure 4). This might indicate that reading comprehension in LLMs is more comparable to humans than other abilities such as long-term memory retrieval, which is required for answering test items in history and economics. Another possible explanation could be the fact that history and economics items more frequently contain images, which have to be understood from descriptions in the alternative text. Since we used text-only LLMs, this discrepancy in the way items were presented was inevitable. Future work could explore whether multimodal models are more successful with these item types.

How to improve psychometric plausibility? To a large degree, the lack of psychometric plausibility is in line with previous research (Hayakawa and Saggion, 2024; Zotos et al., 2025). The success of attempts to make the model response distributions more human-like was very limited – including our temperature scaling approach and Hayakawa and Saggion’s (2024) prompting techniques for injecting personas, uncertainty, or noise. Therefore, in order to improve psychometric plausibility, we will likely need to go beyond zero-shot prompting. Fine-tuning on human response distributions could be a promising direction for future research (cf. Chen et al., 2024).

7 Conclusion

We demonstrated how LLM responses can be analyzed in the context of CTT and IRT and evaluated the human-likeness or psychometric plausibility of zero-shot responses. We found that neither reducing model size nor temperature scaling increased psychometric plausibility to a sufficient degree, but we observed slightly more human-like responses in reading comprehension compared to other subjects. We conclude that human-like response behavior in educational assessments has not emerged from the process of training instruction-tuned LLMs, calling for caution in their use. Fine-tuning on human response distributions may be necessary to create psychometrically plausible models that could be used for piloting.

Limitations

Available item response data. Our analysis is limited by the type and amount of data available in the context of educational assessment. Item banks in high-stakes assessments are usually confidential to avoid leaking information for future test takers, and item responses from single test takers are generally not publicly released. Therefore, in order to keep our results reproducible, we only used publicly available datasets, where only aggregated response distributions and IRT parameters for a relatively small number of items are available. Given a larger amount of and less aggregated data, more fine-grained analyses would be possible (e.g., by including item discrimination in the CTT analysis) and more systematic patterns could be revealed.

Multimodal items. In addition, the NAEP dataset is not ideal for text-only LLMs, because some of the items involve extracting information

from pictures. Although we replaced the pictures with alternative texts and manually removed unanswerable items (see Section 4.1), this could still have affected our results for this dataset.

Test-taker population. The two datasets we used contain response data from two different populations of test takers. While NAEP is targeted at children and adolescents (i.e., mostly L1 English speakers) in the U.S. school system, CMCQRD involves L2 learners of English. This difference could have affected the results and reduce the comparability between the two datasets.

Ethical considerations

We see no ethical issues related to this work. All experiments were conducted with publicly available data and open-source software, and we have made all of our code openly available for reproducibility.⁶ The two datasets we used only contain highly aggregated response data and do not include any information that could lead to the identification of individual test takers.

We used GitHub Copilot for coding assistance in the implementation of the experiment and the analysis of the results. All generated code was manually checked and thoroughly tested.

Acknowledgements

This research is in parts supported by the ERC Consolidator Grant DIALECT 101043235.

References

Marah Abidin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024a. [Phi-3 technical report: A highly capable language model locally on your phone](#). *arXiv*.

Marah Abidin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024b. [Phi-4 technical report](#). *arXiv*.

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.

Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernandez. 2022. [Stop measuring calibration when humans disagree](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1892–1915. Association for Computational Linguistics.

Matthew Byrd and Shashank Srivastava. 2022. [Predicting difficulty and discrimination of natural language questions](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 119–130. Association for Computational Linguistics.

Hua-Hua Chang, Chun Wang, and Susu Zhang. 2021. [Statistical applications in educational measurement](#). *Annual Review of Statistics and Its Application*, 8(1):439–461.

Beiduo Chen, Xinpeng Wang, Siyao Peng, Robert Litschko, Anna Korhonen, and Barbara Plank. 2024. [“Seeing the big through the small”: Can LLMs approximate human judgment distributions on NLI from a few explanations?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14396–14419, Miami, Florida, USA. Association for Computational Linguistics.

Xitao Fan. 1998. [Item response theory and classical test theory: An empirical comparison of their item/person statistics](#). *Educational and Psychological Measurement*, 58(3):357–381.

Guher Gorgun and Okan Bulut. 2024. [Instruction-tuned large-language models for quality control in automatic item generation: A feasibility study](#). *Educational Measurement: Issues and Practice*.

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](#). *arXiv*.

Rita Green. 2020. Pilot testing: Why and how we trial. In *The Routledge Handbook of Second Language Acquisition and Language Testing*, chapter 11, pages 115–124. Routledge.

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.

Thomas M. Haladyna. 2013. Automatic item generation: A historical perspective. In Mark J. Gierl and Thomas M. Haladyna, editors, *Automatic Item Generation: Theory and Practice*, chapter 2, pages 13–25. Routledge, New York.

⁶<https://github.com/mainlp/llm-psychometrics>

- Ronald K. Hambleton and Russell W. Jones. 1993. An NCME instructional module on: Comparison of classical test theory and item response theory and their applications to test development. *Educational Measurement: Issues and Practice*, 12(3):38–47.
- Akio Hayakawa and Horacio Saggion. 2024. Can LLMs solve reading comprehension tests as second language learners? In *Fourth Workshop on Knowledge-infused Learning*.
- Ivan Hernandez and Weiwen Nie. 2022. The AI-IP: Minimizing the guesswork of personality scale item development through artificial intelligence. *Personnel Psychology*, 76(4):1011–1035.
- Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2019. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204.
- John P. Lalor, Hao Wu, and Hong Yu. 2019. Learning latent parameters without human response patterns: Item response theory with artificial crowds. 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 4249–4259, Hong Kong, China. Association for Computational Linguistics.
- Antonio Laverghetta Jr, Animesh Nigohkar, Jamshidbek Mirzakhlov, and John Licato. 2022. Predicting human psychometric properties using computational language models. In *Quantitative Psychology*, pages 151–169, Cham. Springer International Publishing.
- Adian Liusie, Vatsal Raina, Andrew Mullooly, Kate Knill, and Mark J. F. Gales. 2023. Analysis of the Cambridge Multiple-Choice Questions Reading Dataset with a focus on candidate response distribution. *arXiv*.
- Samuel A. Livingston. 2011. Item analysis. In Steven M. Downing and Thomas M. Haladyna, editors, *Handbook of Test Development*, pages 421–441. Taylor & Francis Group.
- 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. ACM.
- Toni A. May, Yiyun Kate Fan, Gregory E. Stone, Kristin L. K. Koskey, Connor J. Sondergeld, Timothy D. Folger, James N. Archer, Kathleen Provinzano, and Carla C. Johnson. 2025. An effectiveness study of generative artificial intelligence tools used to develop multiple-choice test items. *Education Sciences*, 15(2):144.
- Andrew Mullooly, Øistein Andersen, Luca Benedetto, Paula BATTERY, Andrew Caines, Mark J. F. Gales, Yasin Karatay, Kate Knill, Adian Liusie, Vatsal Raina, and Shiva Taslimipoor. 2023. *The Cambridge Multiple-Choice Questions Reading Dataset*. Technical report.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. *2 OLMo 2 Furious*. *arXiv*.
- Valentine Joseph Owan, Kingsley Bekom Abang, Delight Omoji Idika, Eugene Onor Etta, and Bassey Asuquo Bassey. 2023. Exploring the potential of artificial intelligence tools in educational measurement and assessment. *Eurasia Journal of Mathematics, Science and Technology Education*, 19(8):em2307.
- Spiros Papageorgiou, Larry Davis, John M. Norris, Pablo Garcia Gomez, Venessa F. Manna, and Lora Monfils. 2021. *Design Framework for the TOEFL® Essentials™ Test 2021*. Educational Testing Service.
- Jae-Woo Park, Seong-Jin Park, Hyun-Sik Won, and Kang-Min Kim. 2024. Large language models are students at various levels: Zero-shot question difficulty estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8157–8177. Association for Computational Linguistics.
- Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics.
- Barbara Plank. 2022. The “problem” of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Team Qwen, 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, Jiayi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2024. *Qwen2.5 technical report*. *arXiv*.
- Vatsal Raina and Mark Gales. 2022. Multiple-choice question generation: Towards an automated assessment framework. *arXiv*.
- Vatsal Raina, Adian Liusie, and Mark Gales. 2023. Analyzing multiple-choice reading and listening comprehension tests. In *9th Workshop on Speech and Language Technology in Education (SLaTE)*, pages 1–5. ISCA.
- 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, Torino, Italia. ELRA and ICCL.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024. “My answer is C”: First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7407–7416, Bangkok, Thailand. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, 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*. 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.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. *Large language models are not robust multiple choice selectors*. In *The Twelfth International Conference on Learning Representations*.

Leonidas Zotos, Hedderik van Rijn, and Malvina Nissim. 2025. *Can model uncertainty function as a proxy for multiple-choice question item difficulty?* In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11304–11316, Abu Dhabi, UAE. Association for Computational Linguistics.

A Prompt templates

The following prompt template was used for items with a reading passage (i.e., reading comprehension items):

Based on the following text, select the correct answer to the question below.

Text: **{passage}**

Question:
{item stem}
 A) **{option 1}**
 B) **{option 2}**
 C) **{option 3}**
 D) **{option 4}**

Respond only with the letter of the answer (A, B, C, or D).

The following prompt template was used for items without a reading passage (i.e., history and economics items):

Select the correct answer to the following question.

Question:
{item stem}
 A) **{option 1}**
 B) **{option 2}**
 C) **{option 3}**
 D) **{option 4}**

Respond only with the letter of the answer (A, B, C, or D).

B Response accuracy

Figure 5 shows the mode accuracy of models and humans, i.e., the proportion of items where the option with the highest response probability is the correct one. The high accuracy of large models shows that the items are answerable given the available information in the prompt.

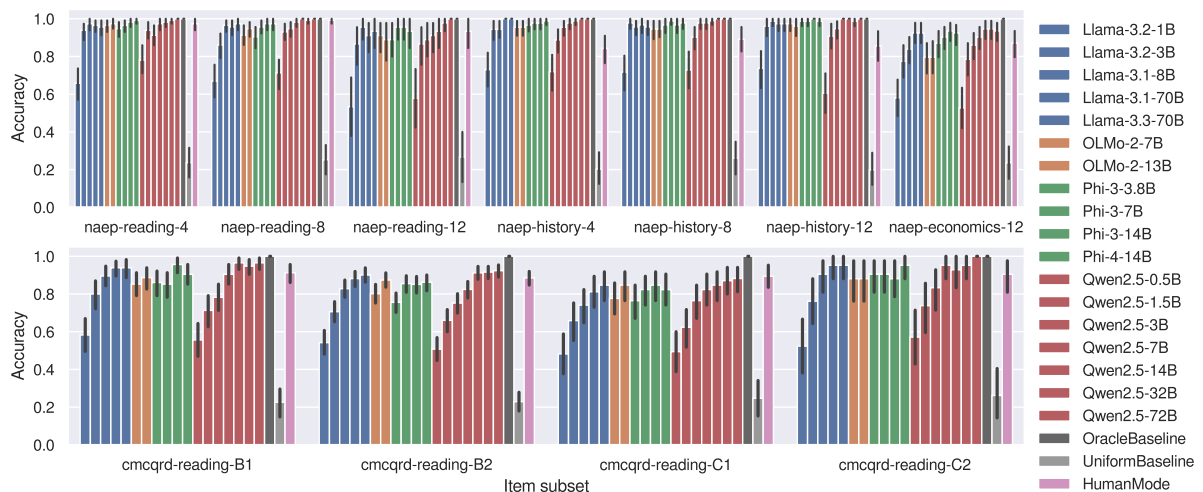


Figure 5: Mode accuracy across item subsets, models, baselines, and humans. Error bars are bootstrapped 95% confidence intervals.

C Details on temperature scaling

We optimized temperature parameters using KL divergence as a loss function and an Adam optimizer (see `analysis.py` in the code repository). The resulting optimized temperature values are visualized in Figure 6. Larger LLMs tend to be overly confident, assigning almost all probability mass to a single answer option, and therefore require higher temperatures to align them with human response distributions.

The effect of temperature scaling can be seen by comparing the results without temperature scaling in Figure 7 with Figure 2.

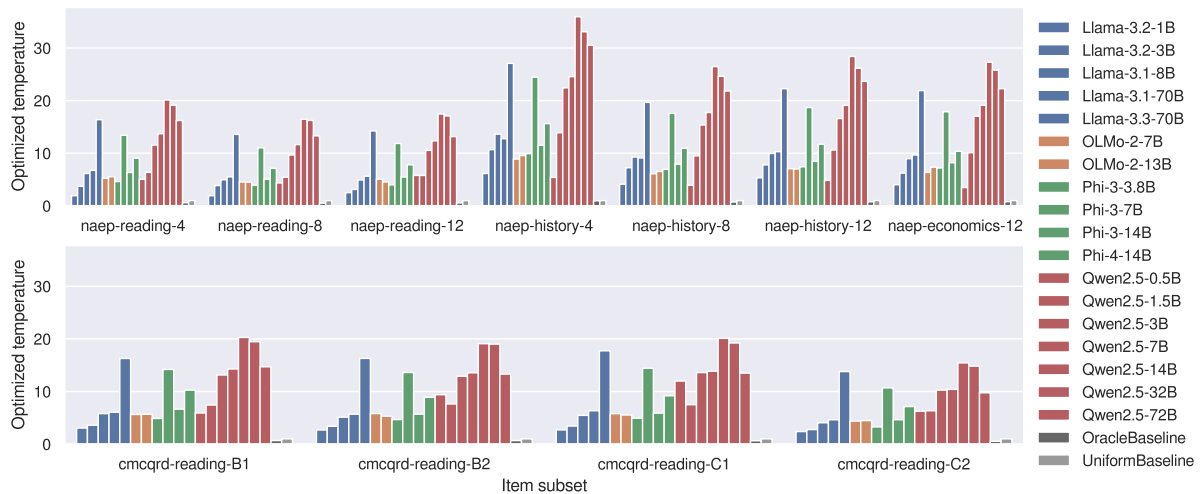


Figure 6: Optimized temperature value for each model and item subset.

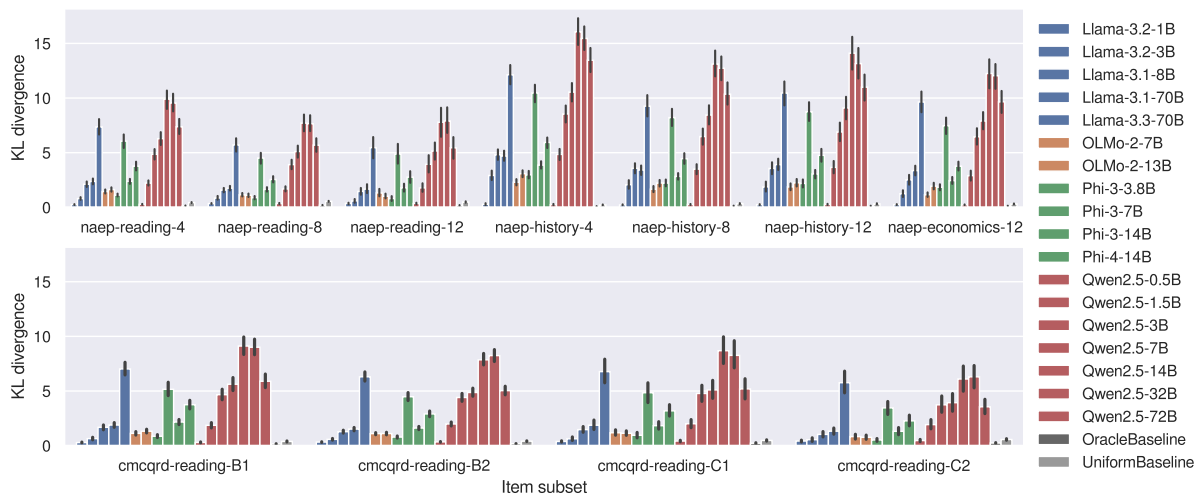


Figure 7: Mean KL divergence between LLM response probability distributions *without* temperature scaling and human response distributions. Error bars are bootstrapped 95% confidence intervals.

D Additional results for CTT analysis

In addition to the correlations between models and humans in Figure 3, Figure 8 shows the full correlation matrices, including model-model correlations. This confirms that the LLMs are much more similar to each other than to humans. In addition, models of similar sizes (but different model families) tend to be more similar to each other compared to models of different sizes.

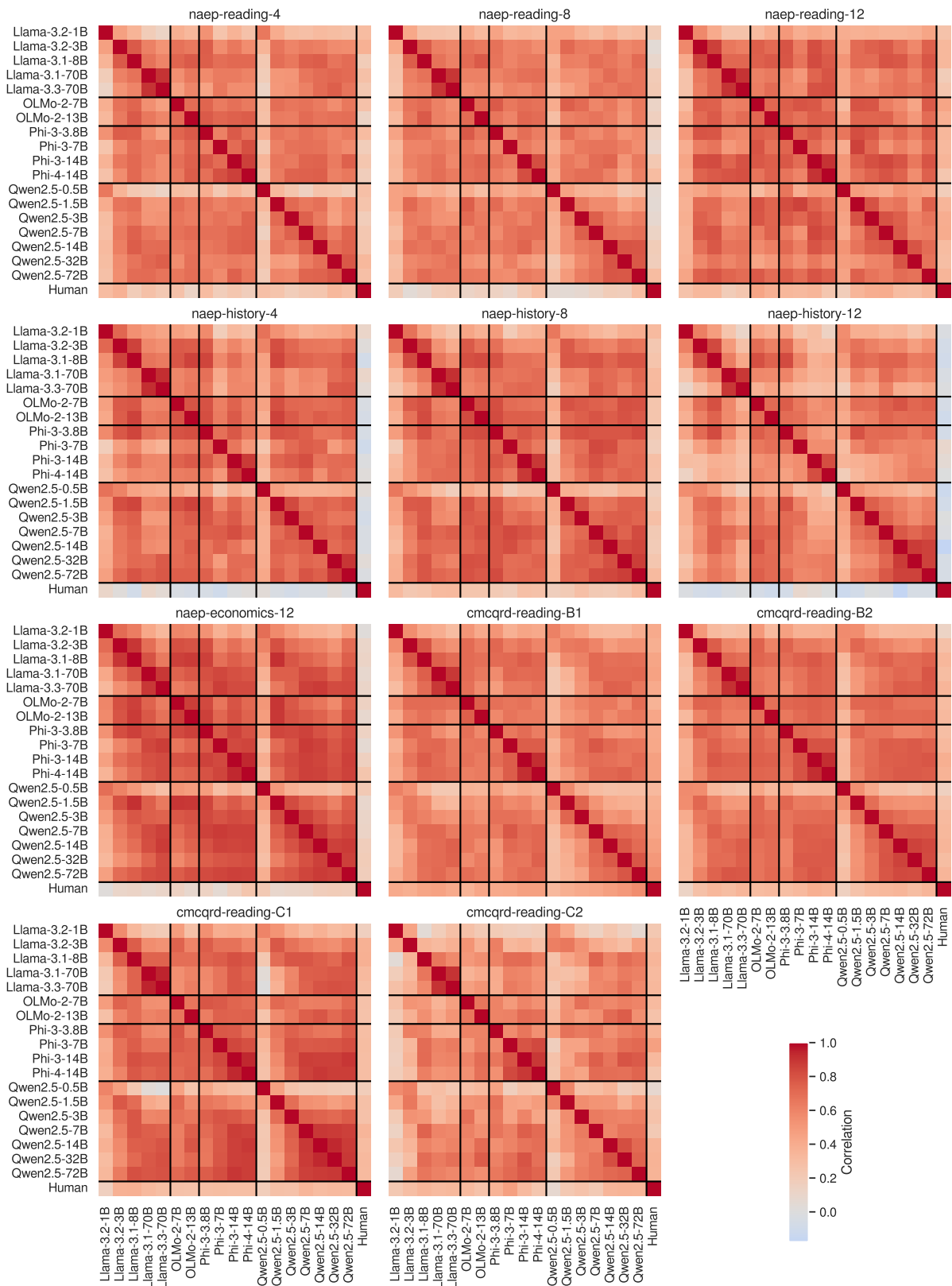


Figure 8: Pearson correlation between all LLM correct response probabilities and human item facilities.

Challenges for AI in Multimodal STEM Assessments: a Human-AI Comparison

Aymeric de Chillaz^{1*} Anna Sotnikova^{1*†} Patrick Jermann¹ Antoine Bosselut¹
¹EPFL

Abstract

Generative AI systems have rapidly advanced, with multimodal input capabilities enabling reasoning beyond text-based tasks. In education, these advancements could influence assessment design and question answering, presenting both opportunities and challenges. To investigate these effects, we introduce a high-quality dataset of 201 university-level STEM questions, manually annotated with features such as image type, role, problem complexity, and question format. Our study analyzes how these features affect generative AI performance compared to students. We evaluate four model families with five prompting strategies, comparing results to the average of 546 student responses per question. Although the best model correctly answers on average 58.5% of the questions using majority vote aggregation, human participants consistently outperform AI on questions involving visual components. Interestingly, human performance remains stable across question features but varies by subject, whereas AI performance is susceptible to both subject matter and question features. Finally, we provide actionable insights for educators, demonstrating how question design can enhance academic integrity by leveraging features that challenge current AI systems without increasing the cognitive burden for students.

1 Introduction

Generative AI has been widely tested in educational applications, including its ability to answer exam-level questions (Sallam, 2023; Lan et al., 2024; Wang et al., 2024a). There are two key challenges: AI can be misused in ways that undermine fair assessment, and its mistakes often appear convincing, potentially misleading students (Borges et al., 2024; Wang et al., 2023; Zhong et al., 2023; Arora et al., 2023). To better understand these risks,

*Both authors contributed equally to this research.

†Corresponding author: aasotniko@gmail.com

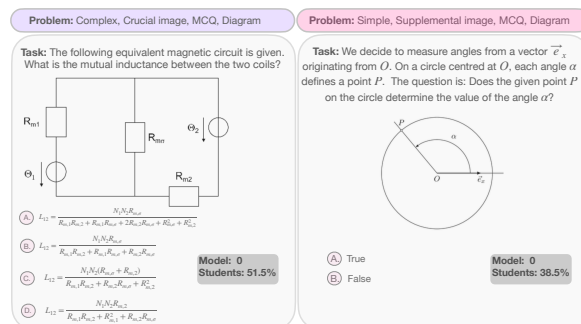


Figure 1: Example of STEM problems with average model performance (majority vote) compared to average student performance.

benchmarks were introduced to assess AI performance (Wang et al., 2024b). Recent advances in multimodal large language models (LLMs) have led to extensive efforts in developing image-based exam datasets, particularly in STEM. Anand et al. (2024) introduced a multimodal physics dataset, expanding from 300 manually created questions to 4,500 using LLMs; Liang et al. (2024) developed SceMQA, a dataset of 1,000+ scientific reasoning problems for students transitioning to college; Zhang et al. (2023) and Das et al. (2024) created multilingual, multimodal benchmarks across various subjects and difficulty levels.

While these benchmarks provide insight into AI capabilities, they primarily evaluate models in isolation, without comparing their performance to humans. As a result, it is unclear whether a model's low performance stems from its limitations or if the problems themselves are inherently difficult for humans too. Understanding what makes a problem easier or harder for AI compared to humans would help warn students about potential risks and guide the design of fairer image-based assessments.

We compile 201 university-level STEM exam questions with images from Bachelor's and Master's programs across 11 subjects of varying complexity. To analyze model performance, each ques-

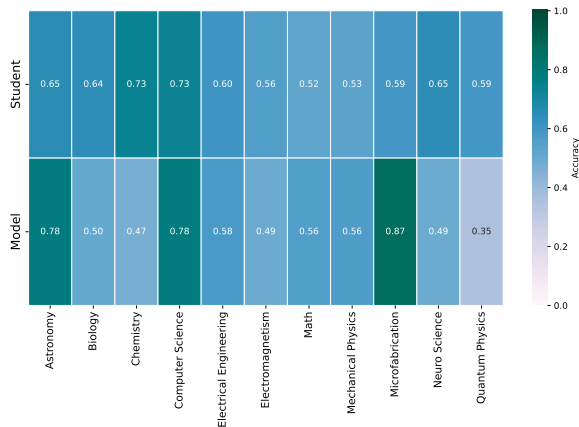


Figure 2: Average model and student accuracy per subject, with model results aggregated using the majority vote strategy.

tion is manually annotated with its image type, role, question type, and problem type. In addition, we collect student performance data, each question receiving at least five responses and an average of 546 respondents across the dataset. To evaluate AI performance, we implement five prompting strategies and test two models from GPT-family —GPT-4o and o1-mini (OpenAI, 2023) as performant models freely available to students, and Qwen 2.5 72B VL (Bai et al., 2025), DeepSeek r1 (DeepSeek-AI et al., 2025), and Claude 3.7 Sonnet, 2025 as performant models with visual capabilities.

Our results indicate that while LLMs perform well in text-based university assessments (Borges et al., 2024), they struggle with questions involving visual components. On average, models perform slightly worse than students. Student performance varies by subject, while model performance depends on question and image features. Based on the analysis, we provide recommendations for designing take-home assignments that maintain academic integrity by challenging models without increasing difficulty for students. These principles can also inform the development of more challenging benchmarks as models continue to improve.

2 Data Set Description

We manually collected 201 questions with images from exams and quizzes in 11 subjects from Bachelor’s and Master’s programs. Each question is paired with a gold answer provided by the educator who authored it. Questions were manually labeled with the following attributes¹:

¹The dataset is available on [GitHub](#)

Image Type: diagram, line plot, algorithm, and picture.

Image Purpose: An image is “supplemental” if all necessary information is in the text and can be inferred without it. It is “crucial” if required to solve the problem.

Question Type: multiple choice questions (MCQ), multiple choice questions multiple answers (MCQ-MA), and compound questions containing multiple sub-MCQ questions connected by the same question topic and having some related information in each other.

Complexity of Problem Conditions: “Complex” questions involve multiple subject concepts, while “Simple” ones require only one or two closely related concepts. This distinction does not indicate difficulty—a question may have one hard concept or multiple simple ones. Categorizing questions this way helped assess whether models struggled with interdependent conditions, as simple questions have fewer variables, while complex ones require integrating more information.

Student performance data was collected from historical course records as aggregated statistics, with 5 to 5,686 respondents per question (average: 546). Student performance also served as an indicator of problem difficulty. Our dataset includes 43 problems where fewer than 40% of the students answered correctly, 79 where 40–70% succeeded, and 79 where more than 70% solved the question.

For detailed dataset statistics and student performance, see Appendices A.1 and A.3.

3 Experiments

Our experiments assess model performance across five prompting strategies and compare it with human performance. Details on prompting strategies are provided in Appendix B. For multiple-choice (MCQ, MCQ-MA) and compound questions, we use exact match with the gold answer without partial credit. Model scores are aggregated using two methods: majority vote (assigning the most common score across strategies) and max (taking the highest achieved score). The max approach provides an upper bound estimate, highlighting if at least one strategy yields the correct answer. Model implementation details are in Appendix C.

4 Analysis

This section presents the experimental results, comparing the model and student performance across

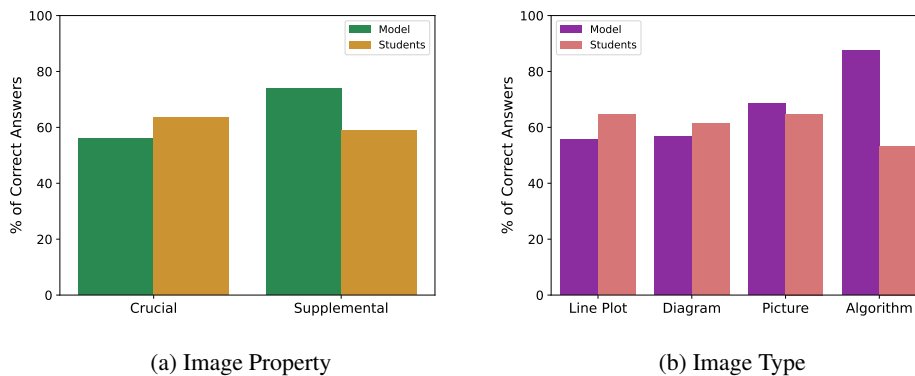


Figure 3: (a) Effect of image properties on model performance aggregated by the majority vote strategy compared to average student performance. (b) Effect of image type on model performance aggregated by the majority vote strategy compared to average student performance.

various dimensions.

4.1 General performance on questions with images

Unless stated otherwise, we use majority vote aggregation. GPT-4o outperforms other models, and we focus on its results throughout. Detailed model comparisons are provided in Appendix D.1. As shown in Figure 9, all prompting strategies perform similarly and roughly match the average human student’s performance on the task.

We analyze model and student accuracy on image-based questions across subjects (Figure 2). Both exhibit subject-specific strengths and weaknesses, but student accuracy varies less (0.52–0.73) than the model’s (0.35–0.87). The model performs exceptionally well in Astronomy, Computer Science (CS), and Microfabrication, likely due to the structured nature of these questions and the model’s ability to apply general concepts. Prior studies have shown that LLMs excel at CS-related tasks (Krüger and Gref, 2023; Song et al., 2024; Borges et al., 2024). In contrast, the model struggles with Quantum Physics, Chemistry, Neuroscience, and Electromagnetism, where complex, content-rich images may pose additional challenges.

4.2 Effect of image features

We examine the role of images in problem-solving, specifically whether they provide essential information absent from the text or if the problem can be solved without them. Figure 3a compares performance based on image necessity. As expected, student accuracy remains similar regardless of image importance, whereas models perform better on questions where images are non-essential. Our

ablation study confirms this trend: removing supplemental images slightly improves model performance, though the effect is minimal (see Appendix D.2 for details).

Next, we analyze performance across image types (Figure 3b). Students perform similarly across line plots, diagrams, and pictures, and struggle the most with algorithm questions. Although the model has no difficulties in processing algorithms, it struggles the most with diagrams and line plots.

4.3 Effect of question features

We observe that students perform similarly across all three question formats. Both students and the model get the best performance on MCQ questions. The model performs slightly better than students on compound questions, a subset of MCQs that are linked to represent steps of a larger problem. However, it struggles the most with MCQMA, often selecting some correct choices but failing to identify all (Figure 4a).

Figure 4b illustrates how the concept count influences performance. Although students perform consistently regardless of the number of concepts in a question, the model struggles when more than two concepts are involved.

We report statistical significance for the model’s and students’ results in Tables 5 and 6.

4.4 Error analysis

To assess the model’s strengths and weaknesses, we analyzed 59 questions split into two sets: ones where the model outperformed students and ones where it underperformed.

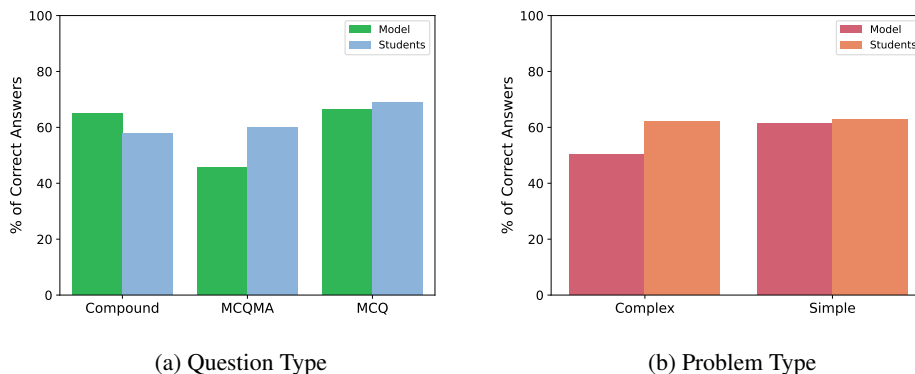


Figure 4: (a) Effect of question type on model performance aggregated by the majority vote strategy compared to average student performance. (b) Effect of problem type on model performance aggregated by the majority vote strategy compared to average student performance.

Questions easy for students, hard for the model

We examined 31 questions where the model scored 0 using the max strategy—failing to produce a correct answer across five prompts—while students achieved over 40% accuracy. We find that humans more easily integrate common sense, domain-specific intuition, and experiential learning, whereas the model struggles to infer conditions or iterations that are not explicitly stated.

One notable category where students outperform the model involves physics-based reasoning and real-world conventions. These problems require understanding implicit relationships, precise numerical or symbolic extraction, and intuition-driven problem-solving. For instance, students effectively interpret diagrams, such as photonic crystal defects or force distributions in mechanical systems, while the model struggles with directional trends and recognizing constraints in visual data. Furthermore, the model has difficulty selecting the correct schema or plot from multiple options, a task that poses less challenge for humans.

Questions hard for students, easy for the model

We analyze 28 questions where students’ performance is below 40% while the model scores above 65%.

A key category where the model outperforms students includes problems requiring structured reasoning, precise pattern recognition, and large-scale knowledge retrieval. These problems follow well-defined rules, abstract mathematical principles, and algorithmic logic. The model’s ability to detect structural patterns allows it to efficiently analyze periodicity in trigonometry, solve algorithmic network problems, and interpret simple electrical

schematics with high accuracy. Unlike intuition-driven tasks, these problems follow clear logical steps. Students often struggle with multi-step reasoning due to cognitive load, whereas the model processes extended contexts effortlessly. As shown in Figure 10, student accuracy declines as question length increases, while the model maintains strong performance. Additionally, models excel in problems requiring abstraction and conceptual knowledge.

5 Conclusion

We show that questions requiring crucial images and multiple concepts, while remaining concise, pose a greater challenge for models without increasing difficulty for students. Additionally, models struggle more than humans in applying domain-specific intuition to problem-solving. However, our analysis reveals that models retain knowledge of the correct answer in 75.5% of questions across at least one prompting strategy but fail to retrieve it consistently. With a majority vote strategy, models achieve 58.5% accuracy, slightly below the human average of 62.7%. While overall performance appears similar, a closer analysis highlights the significant impact of the problem and image features on these results.

Finally, it is important to balance fair accessibility with preventing model misuse, as restrictive measures may inadvertently disadvantage students with vision impairments. For these students, problems with supplemental images are easier to understand through full-text descriptions, similar to how models rely on textual input over visual data.

Limitations

Our study explores how humans and models solve questions involving both images and text. However, it has several limitations.

First, our dataset is relatively small (201 examples). While we ensured high-quality data through manual collection and annotation and confirmed statistical significance, a larger dataset would improve reliability. We opted against automated data augmentation to maintain quality control. To facilitate further research, we publicly release our dataset with annotations.

Second, our grading method does not assign partial credit for multiple-choice multiple-answer (MCQMA) questions, leading to a stricter evaluation of model performance. Additionally, unlike humans, models do not employ elimination reasoning, as we do not adjust prompts for MCQMA responses, potentially disadvantaging them.

Third, when comparing course performance, we do not account for instructor influence, which may affect problem difficulty. This factor can introduce bias also for humans, as different instructors may present varying challenges for students within the same subject.

Acknowledgments

We are grateful to Dr. Jessica Dehler Zufferey for her valuable recommendations on question and image feature design, as well as her feedback on the interpretation of our results. We also thank Christian Vonarburg and Yves Renier for their assistance with data preparation. We appreciate the feedback provided by the EPFL NLP lab members throughout the development of this work. We also gratefully acknowledge the support of the Swiss National Science Foundation (No. 215390), Innosuisse (PFFS-21-29), the EPFL Center for Imaging, Sony Group Corporation, and a Meta LLM Evaluation Research Grant.

References

- Avinash Anand, Janak Kapuriya, Apoorv Singh, Jay Saraf, Naman Lal, Astha Verma, Rushali Gupta, and Rajiv Shah. 2024. [Mm-phyqa: Multimodal physics question-answering with multi-image cot prompting](#). *Preprint*, arXiv:2404.08704.
- Daman Arora, Himanshu Gaurav Singh, and Mausam. 2023. [Have llms advanced enough? a challenging problem solving benchmark for large language models](#). *Preprint*, arXiv:2305.15074.

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. [Qwen2.5-vl technical report](#). *Preprint*, arXiv:2502.13923.

- Beatriz Borges, Negar Foroutan, Deniz Bayazit, Anna Sotnikova, Syrielle Montariol, Tanya Nazaretzky, Mohammadreza Banaei, Alireza Sakhaeirad, Philippe Servant, Seyed Parsa Neshaei, Jibril Frej, Angelika Romanou, Gail Weiss, Sepideh Mamooler, Zeming Chen, Simin Fan, Silin Gao, Mete Ismayilzada, Debjit Paul, Philippe Schwaller, Sacha Friedli, Patrick Jermann, Tanja Käser, Antoine Bosselut, EPFL Grader Consortium, and EPFL Data Consortium. 2024. [Could chatgpt get an engineering degree? evaluating higher education vulnerability to ai assistants](#). *Proceedings of the National Academy of Sciences*, 121(49):e2414955121.

- Rocktim Jyoti Das, Simeon Emilov Hristov, Haonan Li, Dimitar Iliyanov Dimitrov, Ivan Koychev, and Preslav Nakov. 2024. [Exams-v: A multi-discipline multilingual multimodal exam benchmark for evaluating vision language models](#). *Preprint*, arXiv:2403.10378.

- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin,

- Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Tim Krüger and Michael Gref. 2023. [Performance of large language models in a computer science degree program](#). *Preprint*, arXiv:2308.02432.
- Yunshi Lan, Xinyuan Li, Hanyue Du, Xuesong Lu, Ming Gao, Weining Qian, and Aoying Zhou. 2024. [Survey of natural language processing for education: Taxonomy, systematic review, and future trends](#). *Preprint*, arXiv:2401.07518.
- Zhenwen Liang, Kehan Guo, Gang Liu, Taicheng Guo, Yujun Zhou, Tianyu Yang, Jiajun Jiao, Renjie Pi, Jipeng Zhang, and Xiangliang Zhang. 2024. [Scemqa: A scientific college entrance level multimodal question answering benchmark](#). *Preprint*, arXiv:2402.05138.
- OpenAI. 2023. [Gpt-4 technical report](#). *arXiv:2303.08774*.
- Malik Sallam. 2023. [Chatgpt utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns](#). *Healthcare*, 11(6).
- Xiaoshuai Song, Muxi Diao, Guanting Dong, Zhengyang Wang, Yujia Fu, Runqi Qiao, Zhexu Wang, Dayuan Fu, Huangxuan Wu, Bin Liang, Weihao Zeng, Yejie Wang, Zhuoma GongQue, Jianing Yu, Qiuna Tan, and Weiran Xu. 2024. [Cs-bench: A comprehensive benchmark for large language models towards computer science mastery](#). *Preprint*, arXiv:2406.08587.
- Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024a. [Large language models for education: A survey and outlook](#). *Preprint*, arXiv:2403.18105.
- Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. [Scibench: Evaluating college-level scientific problem-solving abilities of large language models](#). *Preprint*, arXiv:2307.10635.
- Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024b. [Scibench: Evaluating college-level scientific problem-solving abilities of large language models](#). *Preprint*, arXiv:2307.10635.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#). *Preprint*, arXiv:2201.11903.
- Wenxuan Zhang, Sharifah Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. [M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models](#). *Preprint*, arXiv:2306.05179.
- Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. [Agieval: A human-centric benchmark for evaluating foundation models](#). *Preprint*, arXiv:2304.06364.

A Data Set Details

Here we present the data set statistics, and examples of questions for every data feature.

A.1 Data Format

Image Type: diagram, line plot, algorithm, and picture.

Image Purpose: supplemental, when the image is non-essential and all information about the problem is stated in the problem text or crucial, when the image is required to solve the problem. One can determine that the question in Figure 5 has a supplemental image since it could be inferred from the text.

Question Type: multiple choice questions (MCQ), multiple choice questions multiple answers (MCQ-MA), and compound questions containing multiple sub-MCQ questions connected by the same question topic and having some related information in each other.

Complexity of Problem Conditions: “Complex” means that the question involves multiple concepts of the subject, while “Simple” would require only one or two closely related concepts to solve the problem. This condition does not directly reflect problem difficulty; a question may involve a single difficult concept or multiple simple ones. Distinguishing between simple and complex questions

In the Hodgkin-Huxley model, the potassium current obeys the equation:

$$I_K = \bar{g}_K n(t)^4 (u(t) - E_K)$$

where \bar{g}_K is the maximal conductance, E_K the potassium reversal potential, and $n(t)^4$ is the proportion of channels that are open at time t . The quantity n obeys a first-order dynamics

$$\frac{dn}{dt} = \frac{n_\infty(u) - n}{\tau_n(u)}$$

with voltage-dependent time constant τ_n and equilibrium value n_∞ .

In order to determine τ_n and n_∞ , Hodgkin and Huxley pharmacologically blocked the sodium current and measured the response of the potassium current to voltage jumps of various amplitudes. The goal of this exercise is to understand this key experiment by studying a simplified version of the Hodgkin-Huxley model. Suppose τ_n and n_∞ have the following form:

$$\tau_n(u) = \begin{cases} 1\text{ms} & \text{if } u \leq 0 \text{ mV} \\ 5\text{ms} & \text{if } 0 < u \leq 25 \text{ mV} \\ 1\text{ms} & \text{if } u > 25 \text{ mV} \end{cases}$$

$$n_\infty(u) = \begin{cases} 0 & \text{if } u \leq 0 \text{ mV} \\ u/50 & \text{if } 0 < u \leq 50 \text{ mV} \\ 1 & \text{if } u > 50 \text{ mV} \end{cases}$$

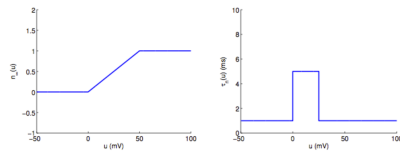


Figure 5: Example of a complex question with supplemental image.

allowed us to evaluate whether models struggled with interdependent conditions. Simple questions involve fewer variables, while complex ones require integrating multiple pieces of information.

One can determine that the question in Figure 5 is “Complex” and the image is “Supplemental”. The question is complex because it involves the Hodgkin-Huxley model, differential equations governing potassium channel dynamics, and voltage-dependent parameters, requiring knowledge of electrophysiology and mathematical modeling. The image is supplemental because it provides graphical representations of $n_\infty(u)$ and $\tau_n(u)$, but all necessary equations and definitions are clearly described in the text, making the image helpful but not essential.

A.1.1 Description of Labels

1. Course_name:

- *Description:* The name or identifier of the course associated with the question.
- *Example:* "Calculus I", "Physics 101"

2. Exercise_name:

- *Description:* The unique exercise id.

3. Question:

- *Description:* The text of the question, may include LaTeX formatting and placeholders for images.

4. Gold_answer:

- *Description:* The correct answer to the question.

5. Question_type:

- *Description:* The format or type of the question.
- *Possible Labels:*
 - "MCQ" (Multiple Choice Question)
 - "MCQMA" (MCQ Multiple Answers)
 - "Compound" (a non-open-ended question with multiple objectives)

6. Image_type:

- *Description:* The type of images included in the question.
- *Possible Labels (Others may be added as we manually label):*
 - "line plot"
 - "bar plot"
 - "scatter plot"
 - "histogram"
 - "pie chart"
 - "table"
 - "image"
 - "diagram"

7. Image_purpose:

- *Description:* The role of the image in the context of the question.
- *Possible Labels:*
 - "Crucial" (Essential for solving the question)
 - "Supplemental" (Doesn't provide additional context)

8. Problem_conditions:

- *Description:* The complexity of the conditions within the problem.
- "Complex" doesn't necessarily mean that the problem is difficult. It simply means that many conditions are in play.
- *Possible Labels:*
 - "Simple" (Conditions are straightforward and not interacting)
 - "Complex" (Multiple conditions interact to find the answer)

9. Question_images:

- *Description:* A list of filenames or identifiers for images included in the question.

10. **Question_length_characters:**
- *Description:* The length of the question text is measured in characters.
11. **Num_objectives:**
- *Description:* The number of sub-questions within the question.
 - *Example:* 1, 2
12. **Language:**
- *Description:* The language in which the question is written.
 - Always in "English" because we translated the French ones.
13. **Original_language:**
- *Description:* The original language of the question before translation.
 - *Example:* "French"
14. **Was_translated:**
- *Description:* Indicates whether the question was translated from another language.
 - *Possible Values:* true or false
15. **Image_file_type:**
- *Description:* The file format of the images used.
 - *Example:* "PNG", "JPEG"
16. **Answer_format:**
- *Description:* The expected format of the answer.
 - *Possible Labels:*
 - "Only MCQ Letter" (previously called MCQ)
 - "Only Numeric Answer"
 - "Derivation"
 - "Text"
 - "Code"
 - "Calculation"
17. **Solution_type:**
- *Description:* Indicates whether the question has a unique correct answer or multiple correct answers.
 - *Possible Labels:*
 - "Unique answer"

– "Multiple answers"

18. **Type_of_text:**

- *Description:* The formatting or typesetting used in the question text.
- *Example:* "LaTeX", "Plain text", "XML"

19. **Objective_dependency:**

- *Description:* Indicates whether the objectives in the question are independent or dependent on previous ones.
- *Possible Labels:*
 - "All Independent" (Objectives can be solved separately)
 - "Dependent" (Some objectives rely on answers from previous parts)

A.2 Data Set Statistics

Data set statistic is presented in Table 1.

Feature	Options	Count
Question Type	MCQ	80
	Compound	59
	MCQMA	46
	Numeric and Formula	16
Image Type	Diagram	109
	Line Plot	45
	Picture	33
	Algorithm	8
	Other	6
Image Purpose	Crucial	162
	Supplemental	39
Problem Conditions	Simple	169
	Complex	32
Course Category	Astronomy	32
	Electrical Engineering	28
	Computer Science	20
	Math	19
	Electromagnetism	17
	Quantum Physics	26
	Mechanical Physics	16
	Neuroscience	15
	Microfabrication	10
	Chemistry	10
Biology	8	

Table 1: Data Statistics

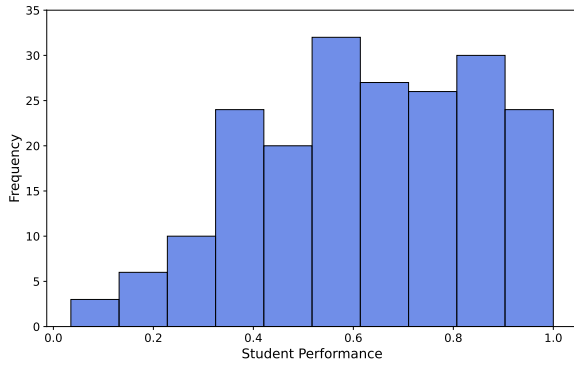


Figure 6: Student accuracy distribution.

A.3 Student Performance

Dataset difficulty illustrated by student performance is presented in Figure 6.

The distribution of students attempting a question is presented in Table 2.

Respondents	Questions
5-20	27
21-50	34
51-100	29
101-500	51
501-1000	33
1001-2000	21
2001-7000	6

Table 2: Distribution of questions by the number of respondents.

B Prompting strategies

B.1 Helper Functions

B.1.1 question_type_prompt

The `question_type_prompt` function creates a tailored instruction based on the type of question being posed. It supports several question types, each associated with a specific directive:

- **MCQ:** Instructs the model to select the correct option by returning only its letter.
- **MCQMA:** Similar to MCQ but expects multiple correct options, concatenated as a single string (e.g., AB rather than A, B).
- **Numeric Question:** Requests that the model output only the numerical answer.
- **Formula Question:** Expects the answer to be provided as a formula.

- **Open Ended:** Directs the model to comprehensively address all parts of the question.

For questions labeled as **Compound**, the function combines the individual instructions corresponding to each subquestion type. It first determines the number of subquestions and then appends the respective prompt text for each, ultimately guiding the model to return its answers as a JSON-formatted list.

B.1.2 generate_format_instruction

The `generate_format_instruction` function provides context-specific formatting advice based on the text’s format:

- **XML:** The instruction reminds the model to interpret XML symbols correctly, ensuring that any formula or question components formatted in XML are properly understood.
- **LaTeX:** Advises careful interpretation of LaTeX expressions, especially for mathematical content.
- **Other:** When the text does not fall into the above categories, no extra formatting instruction is provided.

B.2 Prompting strategies

To generate question-answer pairs, we first conducted an experiment evaluating 12 different prompting strategies. Based on performance results, we selected five strategies for further analysis. Two of these serve as baselines: *direct zero-shot*, the model receives only the question and image without additional instructions or contextual information. *zero-shot chain-of-thought (CoT)* (Wei et al., 2023), the model is asked to produce intermediate reasoning steps before arriving to the final answer. Beyond the baselines, we investigated how the order of multimodal input affects performance. Specifically, we compared cases where the model processes the image at the beginning versus at the end of the text input. Our results indicate that presenting the *image first*, followed by the problem text, leads to better performance. Finally, for models with strong reasoning capabilities but lacking the multimodal component, we implemented a *two-stage* prompting strategy. We first use GPT-4o to generate a textual description of the image. This description is then passed, along with the problem

text, to o1-mini and o1-preview models. The quality of the generated descriptions were manually verified.

B.2.1 Direct zero-shot

A straightforward prompt that presents the question to the model.

```
Direct zero-shot

You are an expert in STEM courses.

Images: <image_names >

[Refer to generate_format_instruction]
Question: <question text >

[Refer to question_type_prompt]
Your Answer:
```

B.2.2 Chain-of-Thought Prompt

This prompt encourages a step-by-step analytical approach, asking the model to think through the problem before answering.

```
Chain-of-Thought Prompt

You are an expert in STEM courses tasked with answering questions with step-by-step analysis. Examine both the image(s) and question text before answering.

Images: <image_names >

[Refer to generate_format_instruction]
Question: <question text >

[Refer to question_type_prompt]
Your Answer: Let's think step by step.
```

B.2.3 Image First Prompt

This prompt prioritizes image analysis by instructing the model to examine the image details before considering the text, and then synthesize a detailed answer.

Image-First Prompt

You are an expert in STEM courses tasked with answering questions. But, first, you must analyze the image(s), which you will follow with the textual analysis.

You will follow the next steps before providing an answer.

Step 1: Analyze the Image(s) First

- Describe elements, patterns, and relationships in the image(s).

Step 2: Use Observations to Analyze the Text

- Use the image understanding to find relevant textual information in the question.

Step 3: Provide a Detailed Answer

- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to generate_format_instruction]

Question: <question text >

[Refer to question_type_prompt]

Make sure to tackle every step mentioned above, before you answer.

Your Answer:

B.2.4 Two Stage Prompt

Image Description Prompt

This prompt requests a detailed description of the provided image, linking its elements to the question context for use by another model. It does not answer the question but aims to provide details that will enable another to answer it.

Image Description Prompt

I am going to provide you with a question with an image. I need you to describe this image in as many details as possible and link those details to the question and its context.

I will then share this description of the image with an LLM which doesn't have vision capabilities, but better reasoning skills than you. In other words, you will be the eyes for that second model. As such, it is primordial that you don't leave out any details!

Note that some details that you think might be useless, may not be, as such make sure that you focus on every aspect.

Here is the Image: <image_names >

Here is the question: <question text >

You may now provide your detailed description. Make sure to follow the instructions that were given to you.

Answer With Image Description Prompt

This prompt asks the model to answer a question based solely on an image description, with a ref-

erence to the detailed image description provided earlier.

Answer With Image Description Prompt

You are an expert in STEM courses and will answer a question that includes an image description..

Here is the description of the image:
<detailed image description >

[Refer to generate_format_instruction]
Here is the question that you need to answer:
<question text >

[Refer to question_type_prompt]
Please, explain the solution and answer in the following format:

```
{  
  "reasoning": "Your explanation.",  
  "answer": "Your answer and nothing more."  
}
```

Your Reasoning and Answer:

B.3 Selecting prompting strategies

Initially, we tested 12 prompting strategies on 10 questions to select the most effective ones for the subsequent experiments. Figure 7 shows a comparison across all strategies. We selected the *two-stage* strategy as the most effective, followed by two baseline strategies, and finally the best strategy for presenting a model with both text and image.

In the basic prompting category, the question was presented along with the image, allowing the models to interpret the visual data without additional instructions. In the second category, prompts directed the models to explicitly consider both the image and text, either together or sequentially, with varying emphasis on fine-grained versus coarse-grained details. Finally, in the third category, models lacking vision capabilities were provided with detailed descriptions of the image instead.

B.3.1 Simultaneous Prompt

This prompt asks the LLM to examine both image and text simultaneously, integrating insights from both modalities before answering. It emphasizes a holistic analysis that considers all available information concurrently.

Simultaneous Prompt

You are an expert in STEM courses tasked with answering questions. Examine both the image(s) and question text before answering.

You will follow the next steps before providing an answer.
Step 1: Analyze the Image(s) and Text Together
- Describe key elements, patterns, and relationships, integrating both sources.
Step 2: Provide a Detailed Answer
- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to generate_format_instruction]
Question: <question text >

[Refer to question_type_prompt]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

B.3.2 Text First Prompt

This prompt directs the LLM to analyze the question text initially and then examine the associated image, using the textual understanding to guide the image analysis. It ultimately expects the model to merge both insights into a coherent, well-informed answer.

Text First Prompt

You are an expert in STEM courses tasked with answering questions. But, first, you must analyze the text, which you will follow with the image analysis.

You will follow the next steps before providing an answer.
Step 1: Analyze the Question Text First
- Understand the question context.
Step 2: Use Observations to Analyze the Image
- Use textual understanding to find relevant visual information (elements, patterns, relationships, etc.)
Step 3: Provide a Detailed Answer
- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to generate_format_instruction]
Question: <question text >

[Refer to question_type_prompt]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

B.3.3 Dual Phase Prompt

This prompt divides the analysis into two distinct phases; first analyzing the image(s) and then the text, before synthesizing the information into a final answer. It ensures that each component is

evaluated independently before being combined for a comprehensive response.

Dual Phase Prompt

You are an expert in STEM courses tasked with answering questions with a dual-phase approach.

You will follow the next steps before providing an answer.

Step 1: Analyze the Image(s) First

- Describe elements, patterns, and relationships in the image(s).

Step 2: Interpret the Question Text Separately

- Identify question context independently of your image findings.

Step 3: Synthesize Textual and Visual Information

- Combine insights from both phases.

Step 4: Provide a Detailed Answer

- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to generate_format_instruction]
Question: <question text >

[Refer to question_type_prompt]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

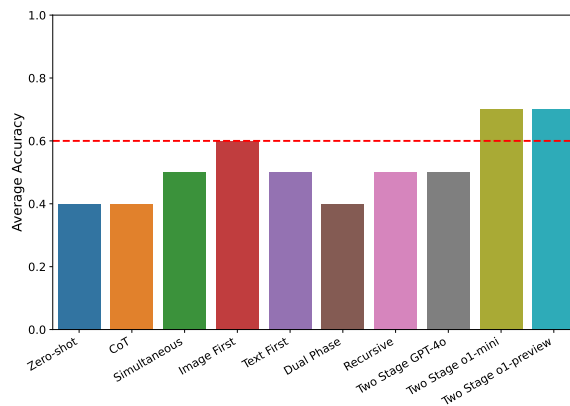


Figure 7: Model performance on the initial set of prompting strategies.

Recursive Prompt

You are an expert in STEM courses tasked with answering questions with recursive analysis.

You will follow the next steps before providing an answer.

Step 1: Analyze the Image(s) First

- Describe elements, patterns, and relationships in the image(s).

Step 2: Use Observations to Analyze the Text

- Use the image understanding to find relevant textual information in the question.

Step 3: Refine Analysis

- Alternate between image and text analysis, refining observations with each pass until a comprehensive understanding of the text and image is reached.

Step 4: Provide a Detailed Answer

- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to generate_format_instruction]
Question: <question text >

[Refer to question_type_prompt]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

B.3.4 Recursive Prompt

This prompt directs the LLM to iteratively alternate between image and text analysis, refining its understanding with each pass until a complete picture is achieved. It is designed to produce a well-considered final answer by progressively integrating and re-evaluating both modalities.

C Model Configuration

To evaluate performance, three OpenAI models were employed: GPT-4o with temperature 0.1, o1-mini-2024-09-12, and o1-preview-2024-09-12. GPT-4o was chosen as the baseline due to its strong vision capabilities. The o1-mini and o1-preview models, in contrast, lack native vision capabilities but exhibit strong reasoning abilities in text-based tasks. While GPT-4o allowed temperature adjustments, the o1 models did not support this feature. The primary focus was on GPT-family models as the ones that students can easily access models to

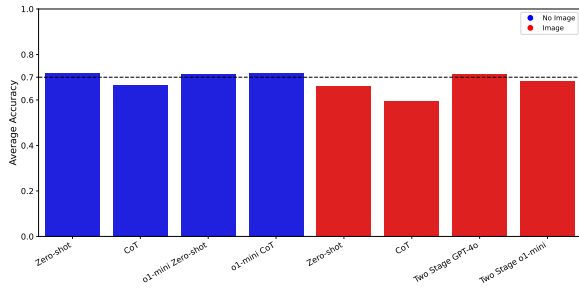


Figure 8: Model performance with and without supplemental image included.

run questions themselves while preparing a take-home assignment. We are trying to provide also some recommendations for educators on how to make such assignments less vulnerable to generative AI use. To explore the effect on different model families, we also test Qwen 2.5 72B VL, r1 Deepseek, and Claude 3.7 Sonnet, 2025.

We used pandas,² json,³ numpy,⁴ and scikit-learn⁵ to process our results, compute accuracy scores, and compute statistical significance.

D Additional Experimental Results

D.1 Model performance comparison

We observe that the GPT model is the most performant one and that, in general, the models follow our findings. The results in various characteristics are presented in Table 4.

Looking at the performance per course in Table 3, we see that our findings hold. Also, sometimes, there are cases when Claude 3.7 or R1 outperform GPT model: in a subject like biology and mechanical physics.

D.2 Removing supplemental image

We tested the same prompts with and without supplemental images. For the *two stage* prompts we removed mentions of the image and didn't pass the descriptions. Figure 8 shows that the presence or absence of the image doesn't affect model performance.

D.3 Model performance vs student performance

In Figure 9, we compare model performance across five prompting strategies and two aggregation strategies with average student performance.

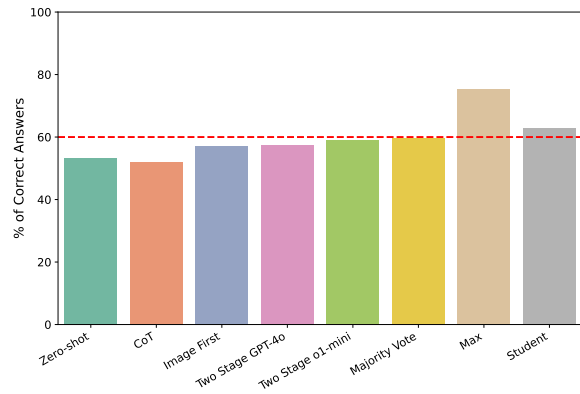


Figure 9: Average GPT-family models performance across five prompting strategies, aggregated results with the majority vote and maximum strategy and student performance.

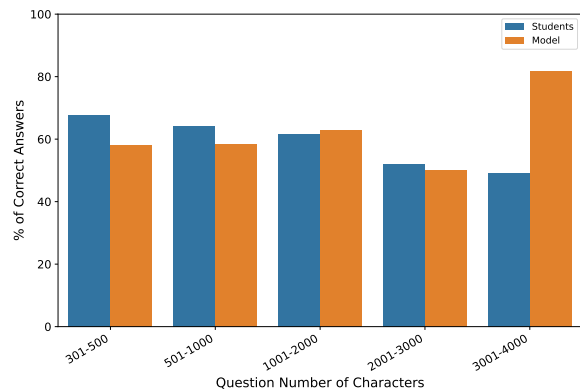


Figure 10: Comparison of student vs model accuracy depending on the question length in characters.

D.4 Student vs model accuracy depending on the question length

Figure 10 shows the comparison of the student and model accuracy (average majority vote) depending on the length of the question in characters.

D.5 Model performance across question and image features

Model and student performance per question and image features with 95% confidence intervals are presented in Tables 5 and 6.

²<https://pandas.pydata.org/docs/index.html>

³<https://docs.python.org/3/library/json.html>

⁴<https://numpy.org/doc/stable/index.html>

⁵<https://scikit-learn.org/stable/>

Course Category	GPT-4o	Claude 3.7	R1	Qwen 2.5-72B
Astronomy	0.78	0.55	0.58	0.63
Biology	0.50	0.75	0.63	0.50
Chemistry	0.47	0.47	0.37	0.30
Computer Science	0.78	0.68	0.74	0.72
Electrical Engineering	0.58	0.48	0.37	0.43
Electromagnetism	0.49	0.45	0.45	0.45
Math	0.56	0.57	0.47	0.56
Mechanical Physics	0.56	0.56	0.63	0.56
Microfabrication	0.87	0.60	0.64	0.52
Neuro Science	0.49	0.31	0.29	0.36
Quantum Physics	0.35	0.29	0.18	0.24

Table 3: Average model performance across different course categories.

Category	Label	GPT-4o	Claude 3.7	R1	Qwen 2.5-72 B
Question feature	Simple	0.613	0.517	0.469	0.481
	Complex	0.503	0.454	0.459	0.502
	MCQ	0.663	0.563	0.538	0.575
	MCQMA	0.457	0.370	0.304	0.283
	Compound	0.650	0.566	0.533	0.538
Image feature	Crucial	0.561	0.473	0.426	0.448
	Supplemental	0.740	0.652	0.641	0.635

Table 4: Average model performance across different question and image features.

Category	Label	Model Accuracy and 95 % CI		
		Accuracy Mean	Lower Bound	Upper Bound
Question feature	Simple	0.613	0.54	0.68
	Complex	0.503	0.35	0.65
	MCQ	0.663	0.59	0.77
	MCQMA	0.457	0.68	0.84
	Compound	0.650	0.55	0.74
Image feature	Crucial	0.561	0.49	0.63
	Supplemental	0.74	0.60	0.86
	Algorithm	0.875	0.63	1.00
	Diagram	0.566	0.48	0.65
	Picture	0.687	0.54	0.84
	Line Plot	0.556	0.43	0.69

Table 5: Model accuracy means and 95% Confidence Intervals. CI is computed with non-parametric bootstrap using 1000 resamples.

Category	Label	Student Accuracy and 95 % CI		
		Accuracy Mean	Lower Bound	Upper Bound
Question feature	Simple	0.628	0.59	0.66
	Complex	0.622	0.54	0.70
	MCQ	0.689	0.64	0.74
	MCQMA	0.599	0.54	0.67
	Compound	0.579	0.52	0.64
Image feature	Crucial	0.637	0.60	0.67
	Supplemental	0.588	0.51	0.67
	Algorithm	0.534	0.40	0.66
	Diagram	0.613	0.57	0.66
	Picture	0.648	0.57	0.71
	Line Plot	0.645	0.58	0.71

Table 6: Student accuracy means and 95% Confidence Intervals for different image types. CI is computed with the non-parametric bootstrap using 1000 resamples

LOOKALIKE: Consistent Distractor Generation in Math MCQs

Nisarg Parikh¹, Alexander Scarlatos^{1*}, Nigel Fernandez^{1*},
Simon Woodhead², Andrew Lan¹
University of Massachusetts Amherst¹, Eedi²
{nkparikh,nigel,ajscarlatos,andrewlan}@cs.umass.edu
simon.woodhead@eedi.co.uk

Abstract

Large language models (LLMs) are increasingly used to generate distractors for multiple-choice questions (MCQs), especially in domains like math education. However, existing approaches are limited in ensuring that the generated distractors are consistent with common student errors. We propose LOOKALIKE³, a method that improves error–distractor consistency via preference optimization. Our two main innovations are: (a) mining synthetic preference pairs from model inconsistencies, and (b) alternating supervised fine-tuning (SFT) with Direct Preference Optimization (DPO) to stabilize training. Unlike prior work that relies on heuristics or manually annotated preference data, LOOKALIKE uses its own generation inconsistencies as dispreferred samples, thus enabling scalable and stable training. Evaluated on a real-world dataset of 1,400+ math MCQs, LOOKALIKE achieves 51.6% accuracy in distractor generation and 57.2% in error generation under LLM-as-a-judge evaluation, outperforming an existing state-of-the-art method (45.6% / 47.7%). These improvements highlight the effectiveness of preference-based regularization and inconsistency mining for generating consistent math MCQ distractors at scale.

1 Introduction

Multiple-choice questions (MCQs) are used in educational assessments (Nitko, 1996; Airasian, 2001; Kubiszyn and Borich, 2016) to evaluate student understanding across various subjects and grades (Thomas et al., 2025). An MCQ consists of a question stem and a set of options, including a correct answer and multiple incorrect alternatives, referred to as *distractors* (Fernandez et al., 2024; Feng et al., 2024). *Distractors* are incorrect answers that students reach by making an *error* while answering the question. It can be rooted in many ways, e.g., the

student overgeneralizing to a new context, exhibiting an ingrained misconception, or simply slipping and being careless. Designing effective distractors can be crucial to the assessment and pedagogical aspects of MCQs (Simkin and Kuechler, 2005), since they help us identify student errors and prepare ways to mitigate them.

Hand-crafting high-quality distractors requires extensive human effort by content designers and teachers since it requires them to anticipate common student errors, which can be difficult in subjects like math. Therefore, recent works have leveraged artificial intelligence, especially large language models (LLMs), to automate this process. Previous works on *distractor* generation for MCQs have attempted to prompt LLMs to generate distractors (Feng et al., 2024), as well as fine-tune LLMs to generate possible student errors and then distractors caused by such errors, as shown in DiVERT (Fernandez et al., 2024). As noted in these works, the bottleneck in distractor generation performance is *consistency*: LLMs are often capable of identifying mathematically feasible errors, but struggle at following such erroneous instructions to arrive at the corresponding distractor (a similar finding was also made in (Sonkar et al., 2024a)). As shown in Table 1, both fine-tuned LLMs and the LLMs in DiVERT sometimes fail to follow the input error explanation to arrive at a consistent distractor. In the second example, the fine-tuned LLM fails to follow the error, “finds 13% of an amount rather than the percentage being asked”, arriving at an inconsistent distractor (12) rather than the consistent distractor (5.2).

To address this limitation, one natural solution is to regularize an LLM-based distractor generator, which takes the question stem and an error as input, to enforce that the generated distractor matches the input error. To this end, we resort to preference optimization, specifically direct preference optimization (DPO) (Rafailov et al., 2023). DPO

*Equal Contribution.

³Code: <https://github.com/umass-ml4ed/LookAlike>

Question stem: Calculate: 130% of 40 = □	
Error	Distractor
Plausible error, plausible and consistent distractor.	
Added the values together instead of finding the percentage.	170
Plausible error, plausible but inconsistent distractor.	
Finds 13% of an amount rather than the percentage being asked.	12
Implausible error, plausible but inconsistent distractor.	
When solving a problem that requires an inverse operation (e.g. missing number problems), does the original operation.	90
Implausible error, implausible and inconsistent distractor.	
Does not understand that 100% is the whole amount.	20

Table 1: Examples of inconsistent error-distractor pairs generated by SFT (second and fourth pairs), and a state-of-the-art method, DiVERT (Fernandez et al., 2024) (third pair). LOOKALIKE mines generation inconsistencies for scalable preference optimization.

training requires *preference pairs* among outputs, i.e., a distractor that matches the input error and a distractor that does not. However, we empirically find two main challenges in using DPO to promote error-distractor consistency:

- Acquiring high-quality preference data typically requires costly manual annotation or unreliable synthetic heuristics (Li et al., 2023; Tan et al., 2024), which is difficult due to the nature of the distractor generation task.
- Models trained with DPO may deteriorate in quality after a few epochs (Pal et al., 2024; Liu et al., 2024b; Yan et al., 2025; Xu et al., 2024), showing training instability.

Contributions

In this paper, we introduce LOOKALIKE, proposing two methods to tackle these challenges and improve error-distractor consistency in math MCQs. For the first challenge, we create preference pairs by generating *synthetic negative samples*: we evaluate LLM-generated errors, in addition to distractors, and use inconsistently generated errors and distractors as informative negative samples. This method creates meaningful signals that, when used in conjunction with consistent errors and distractors in DPO training, improve the consistency of LLMs in distractor generation. For the second challenge, we employ a regularization method in DPO training, which performs supervised finetuning (SFT)

and DPO *alternatively* in consecutive training iterations, which performs better than combining them both into a single objective, as done in recent works (Liu et al., 2024b; Pal et al., 2024).

We conduct extensive experiments on a real-world dataset containing math MCQs used by hundreds of thousands of students, with human-written error descriptions behind each distractor. Results show that LOOKALIKE, compared to state-of-the-art baselines, improves distractor generation performance by up to 6%. We also show that LOOKALIKE improves error generation by up to 10%, using an LLM-as-a-Judge evaluation. We also provide qualitative examples and an error analysis highlighting the improved consistency of generated *errors* and *distractors*.

2 Background

In this section we formally introduce the tasks of *error* and *distractor* generation in math MCQs. We also detail a baseline for preference pair creation and a baseline for DPO regularization, combining preference alignment with supervised learning.

2.1 Task Definition

We consider an MCQ Q defined by its textual components: a **question stem** s , (optionally) its **correct answer** or **key** k , (optionally) an explanation of the key f , (optionally) question topic/concept tags t , and a set of **incorrect answer options** called ground truth distractors D . Each $d_i \in D$ is (optionally) associated with a corresponding ground truth human-written **error explanation** or error $e_i \in E$. All textual components above are represented as sequences of words and math symbols. We aim to model the space of plausible student errors E and their corresponding distractors D . We define two primary tasks:

1. **Error Generation:** Learn an LLM parameterized model, $LLM^{err}(s, k, f, t, d_i) \rightarrow \hat{e}_i$, that outputs an error description \hat{e}_i consistent with the given input distractor d_i and MCQ.
2. **Distractor Generation:** Learn an LLM parameterized model, $LLM^{dis}(s, k, f, t, e_i) \rightarrow \hat{d}_i$, that outputs a distractor \hat{d}_i consistent with the given error description e_i and MCQ.

2.2 Baseline: Preference Pairs from Ground-truth Error-Distractor Pairs

As a natural starting point, following a similar method from (Scarlatos et al., 2024b), one can

construct preference pairs for DPO as follows: For each question, there are multiple distractors ($D = d_1, d_2, \dots, d_n$) and their corresponding errors ($E = e_1, e_2, \dots, e_n$). As a baseline, for the error behind the i^{th} distractor, e_i , we can use d_i itself as the preferred response, and use the remaining distractors ($d_j \in D \setminus \{d_i\}$) as dispreferred responses. We use a similar procedure for the errors. We dub this method for preference pair construction as **DPO-GT** (ground truth). However, the number of dispreferred responses is limited by the number of human-written error-distractor pairs for the question. LOOKALIKE, on the other hand, creates preference pairs by generating *synthetic negative samples*, allowing for an arbitrary number of dispreferred responses for scalable preference optimization, resulting in improved consistency in both error and distractor generation (Section 3.1).

2.3 Baseline: DPO Regularization

Models trained with DPO have been shown to deteriorate in quality after a few epochs due to training instability (Pal et al., 2024; Liu et al., 2024b; Yan et al., 2025; Xu et al., 2024). Existing regularization techniques to improve DPO training stability include Regularized Preference Optimization (**RPO**) (Liu et al., 2024b), and DPO-Positive (**DPOP**) (Pal et al., 2024). RPO optimizes both the DPO loss and the SFT loss jointly, i.e., $L_{RPO} = L_{DPO} + \lambda\beta L_{SFT}$. The SFT loss uses the preferred response as the ground-truth completion. RPO suffers from conflicting gradient directions (Shi et al., 2023; Liu et al., 2024a), especially when the preference-based signal (DPO) incentivizes ranking decisions that are misaligned with the next-token prediction signal (SFT). DPOP uses the SFT objective as a penalty but their improvement is limited to preference pairs with high edit distances between them. LOOKALIKE, on the other hand, proposes an alternating optimization approach to stabilize DPO training, interleaving SFT and DPO training either at the per-batch or per-epoch level, resulting in improved consistency in both error and distractor generation compared to RPO and DPOP (Section 3.2).

3 Methodology

We now detail our framework, LOOKALIKE, which a) creates preference pairs by generating synthetic negative samples, and b) employs a DPO regularization technique of alternating optimization be-

tween SFT and DPO for better training stability, leading to improved error and distractor generation consistency.

3.1 Mining Preference Pairs via Inconsistencies for DPO

Prior work (Fernandez et al., 2024) has highlighted a significant issue of consistency in distractor generation performance, with LLMs struggling to follow error descriptions to arrive at corresponding distractors, examples of which are shown in Table 1. LOOKALIKE mines these generation inconsistencies as *synthetic negative samples* to create preference pairs for DPO training.

We visualize our preference pair creation in LOOKALIKE in Figure 1. For distractor generation, LLM^{dis} overgenerates a set of distractors for an input question stem and a ground-truth error. Each generated distractor is then compared against the ground-truth distractor. In our preference dataset, generated distractors that match the ground-truth distractor exactly are preferred responses, while those that do not exactly match the ground-truth distractor are dispreferred responses. A similar process is applied to create preference pairs for error generation, with exact string match⁴ used to compare generated errors against the ground-truth error to form preference pairs.

Formally, given an MCQ dataset with samples, (s, e, d) , where s is the question stem, e is the error description, and d is the corresponding distractor, we first train a distractor generation model, LLM^{dis} , to output the corresponding distractor through SFT. To create preference pairs, we then overgenerate multiple distractors $\hat{d} \in \hat{D}$ from the fine-tuned LLM^{dis} for each (s, e) pair. For each generated distractor \hat{d} , we check if \hat{d} matches the ground-truth distractor d exactly. If yes, we add \hat{d} as a preferred response, and if no, we add \hat{d} as a dispreferred response in our distractor generation preference dataset. Having constructed the preference dataset, we further train our fine-tuned LLM^{dis} through DPO (Rafailov et al., 2023). A similar process is applied to form our error generation preference dataset which is then applied for DPO training of LLM^{err} . Creating preference pairs from the static ground-truth dataset is limited by the number of human-written annotations (Section 2.2). LOOKALIKE, on the other hand, uses generations from the currently fine-tuned LLM to

⁴LLM-as-a-Judge using GPT-4o-mini as a similarity measure led to lower performance.

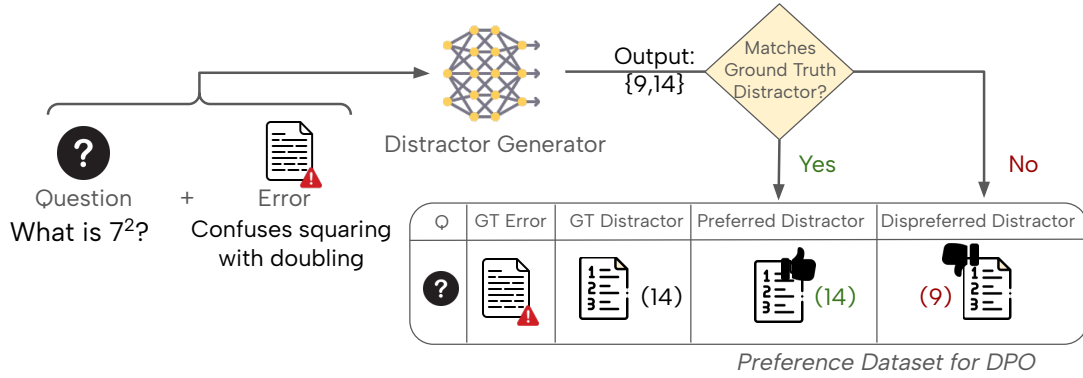


Figure 1: LOOKALIKE creates preference pairs by overgenerating a set of distractors for a question and error, and preferring those that match the ground-truth distractor exactly. An analogous process for error generation.

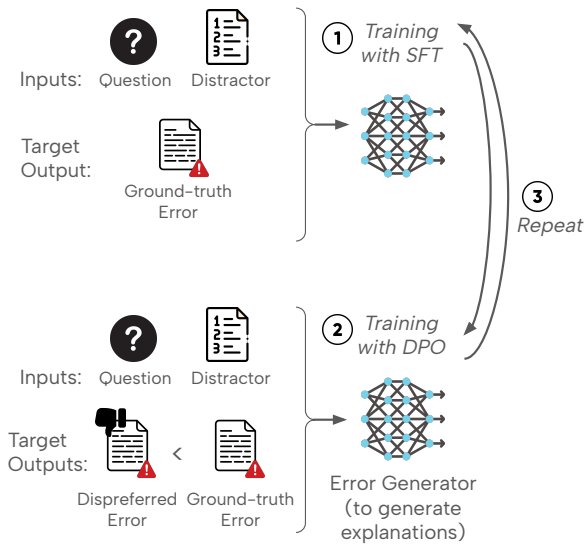


Figure 2: LOOKALIKE employs an alternating optimization strategy, switching between SFT and DPO objectives to regularize DPO training.

create an arbitrary number of dynamic preference pairs, with negative preference signals being more aligned with the inconsistency failure modes of the fine-tuned LLM.

3.2 DPO Regularization Through Alternating Optimization

We empirically observe that models trained with DPO deteriorate in quality after a few epochs due to training instability. We show examples of degradation in error generation quality over three training epochs in Table 2. We observe errors become more verbose with an increase in length and are out-of-distribution from the human-written errors as the number of DPO training epochs increases, as also shown in prior work (Park et al., 2024).

To mitigate this issue, we introduce a regu-

larization strategy that trains the error/distractor-generation LLM by *alternating optimization*, i.e., by switching between SFT and DPO objectives during training, as shown in Figure 2. This alternating optimization allows the LLM to periodically recalibrate to the ground-truth distribution (via SFT) while remaining faithful to learning ranking preferences of consistent generations (via DPO). After each SFT optimization, the preference dataset is recomputed (Section 3.1) for the subsequent DPO optimization, using the currently trained LLM for better alignment, allowing for dynamic and scalable preference pair creation. We experiment with alternating between SFT and DPO optimization at two different levels: per-batch and per-epoch, picking the one giving better performance empirically. For both levels, the preference dataset is recomputed after every epoch.

Alternating Optimization Per-Batch. At each training step t , the LLM parameters θ are updated using a learning rate of η following:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t), \quad (1)$$

where the loss function L alternates based on a batch-level schedule:

$$L(\theta_t) = \begin{cases} L_{SFT}(\theta_t), & \text{if batch } t \text{ is even} \\ L_{DPO}(\theta_t), & \text{otherwise} \end{cases} \quad (2)$$

Alternating Optimization Per-Epoch. As a coarser alternative, the loss function L alternates based on an epoch-level schedule:

$$L(\theta_t) = \begin{cases} L_{SFT}(\theta_t), & \text{if epoch } t \text{ is even} \\ L_{DPO}(\theta_t), & \text{otherwise} \end{cases} \quad (3)$$

Question	$\frac{3}{7}$ of a group of students are boys. What would be a possible ratio of boys to girls?
Key	3 : 4
Ground-truth Distractor	3 : 10
Ground-truth Error	Uses the denominator when converting from fractions to ratio, rather than numerator.
Generated Error (Epoch 1)	Includes the denominator when converting a fraction to a ratio.
Generated Error (Epoch 2)	When converting a fraction to a ratio, puts the other side of the ratio as the denominator.
Generated Error (Epoch 3)	When converting a fraction to a ratio, thinks you just use the numerator and denominator as the numbers in the ratio. Additionally, thinks you can use the denominator on its own as the total number of parts in a ratio.

Table 2: Error generation quality deteriorates over DPO training epochs without using regularization.

4 Experimental Evaluation

In this section, we detail our experiments on a real-world math MCQ dataset, evaluating the efficacy of LOOKALIKE in comparison with state-of-the-art baselines for both distractor generation and error generation.

4.1 Dataset

We conduct our experiments on a real-world math MCQ dataset from a large learning platform used by hundreds of thousands of students. The dataset consists of 1,434 math MCQs, each containing a question stem, key, explanation of the key, topic/concept tags, and 3 distractors along with their respective teacher-written error descriptions explaining why a student might select that distractor. The MCQs are designed for students aged between 10 to 13 and span 41 distinct mathematical subtopics, including *Arithmetic*, *Fractions*, and *Solving Equations*. We split the dataset into training, validation, and test by questions to ensure no overlap across splits using a 72%-16%-12% proportion. See Appendix E for math MCQ examples.

4.2 Baselines

We compare LOOKALIKE with 3 baselines. The **SFT** baseline, used as a baseline in (Fernandez et al., 2024), fine-tunes an LLM to generate the corresponding distractor (or error) given the question and the error (or distractor) as input. The **DiVERT** (Fernandez et al., 2024) baseline employs a variational approach to learn an interpretable error space behind distractors. Post variational training, we use the error generation and distractor generation LLMs from DiVERT as baselines. We also compare against forming preference pairs from the ground-truth error-distractor pairs; we continue training the SFT baseline on this preference dataset using DPO and refer to the resulting

model as **DPO-GT** (Section 2.2). For fairness, we regularize DPO training for DPO-GT by exploring all techniques (RPO, DPOP, our alternating per-batch optimization, and our alternating per-epoch optimization), and choose the regularization (per-epoch) that results in the best performance.

4.3 Metrics

Distractor Evaluation. Following prior work on distractor generation (Fernandez et al., 2024; Feng et al., 2024), we use **Exact match** as our evaluation metric to measure alignment between the generated distractor and the ground-truth distractor corresponding to a question and error.

Error Evaluation. Automated text similarity metrics like exact string match, ROUGE-L F1 (Lin, 2004), or BERTScore F1 (Zhang et al., 2020) are unsuitable for error evaluation given the open-ended and mathematical nature of errors. We therefore adopt an **LLM-as-Judge** (Liu et al., 2023; Zheng et al., 2023) evaluation, prompting GPT-4o-mini to evaluate if the generated error is mathematically equivalent to the ground-truth error given the question and corresponding distractor. We show our prompt in Appendix B.

4.4 Implementation Details

Following prior work (Fernandez et al., 2024), all methods use MetaMath-Mistral 7B (Yu et al., 2024b) as their base LLM, as we found it provides a suitable prior within the 7B parameter size models for mathematical reasoning. At test time, we use standard beam search with 10 beams for distractor generation, and diverse beam search (Vijayakumar et al., 2018) with 10 beams for error generation. Detailed hyperparameter settings for all methods are provided in Appendix A.

To ensure fair comparison, we limit LOOKALIKE’s synthetic generation to 3 distractors and 3

	Distractor Gen (Exact Match \uparrow)	Error Gen (LLM-as-Judge \uparrow)
SFT	44.76	46.68
DiVERT	45.64	47.72
DPO-GT	51.44	57.02
LOOKALIKE	51.56	57.18

Table 3: Cross-validation performance on distractor generation and error generation for all methods across 5 folds. LOOKALIKE outperforms SFT and the prior state-of-the-art method DiVERT (Fernandez et al., 2024), and is comparable to DPO-GT.

errors per training sample per epoch, resulting in a similar order of magnitude of training samples as DPO-GT. We also use the same training budget and regularization for both methods. All fine-tuned models, including SFT and DPO-based variants, were trained with LoRA to ensure parameter efficiency and consistency in comparison.

5 Results, Analysis and Discussion

In this section, we detail our experimental results. We quantitatively evaluate the quality of generated errors and distractors, qualitatively evaluate the consistency of generated errors through human evaluation, conduct an ablation study on DPO regularization techniques, and perform an error analysis on failed cases of error generation.

5.1 Quantitative Evaluation

Table 3 shows the average performance on distractor generation and error generation, across 5 cross-validation folds, for all methods. DPO-based methods, DPO-GT and LOOKALIKE, are trained using our alternating optimization technique for DPO regularization, choosing the alternating level (per-batch or per-epoch) that works best for downstream task performance. DPO-GT works best with per-epoch for both tasks, while LOOKALIKE works best with per-epoch for distractor generation, and per-batch for error generation.

Preference optimization using inconsistent error-distractor pairs improves consistency. LOOKALIKE outperforms SFT and the previous state-of-the-art baseline DiVERT (Fernandez et al., 2024), by a wide margin of 6.8% and 5.92% on distractor generation, and 10.5% and 9.46% on error generation performance, respectively. The improvement is statistically significant with p-values < 0.05 measured using a one-sample Wilcoxon signed-rank test (Rey and Neuhäuser,

	Dis Gen (Exact M. \uparrow)	Error Gen (LLM-as-Judge \uparrow)
DPO-GT w/o Reg.	47.68	53.96
+ DPOP	47.80	52.74
+ RPO	49.14	52.44
+ Per-batch	49.66	55.74
+ Per-epoch	51.44	57.02
LOOKALIKE w/o Reg.	47.98	49.34
+ DPOP	49.38	49.44
+ RPO	49.60	49.66
+ Per-batch	50.84	57.18
+ Per-epoch	51.56	56.64

Table 4: Ablation study of various DPO regularization techniques. Our alternating (per-batch/epoch) optimization performs best for both DPO-GT and LOOKALIKE.

2011). This result validates our idea of mining error-distractor inconsistencies as preference pairs for DPO training to improve both error and distractor generation consistency. Further, LOOKALIKE, although using synthetic negative samples drawn from its own inconsistent generations as preference pairs, is comparable in performance to DPO-GT, which uses human-written annotations as preference pairs, demonstrating the potential and flexibility of LOOKALIKE for scalable, domain-agnostic preference optimization.

Although the performance difference between LOOKALIKE and DPO-GT appears small (0.12% and 0.16% on distractor and error generation respectively), it is important to note that LOOKALIKE achieves this using automatically mined preference pairs from inconsistent generations, without relying on ground-truth labels, highlighting its scalability. Moreover, the improvement over DiVERT (5.9-10.5%) is substantial and statistically significant.

Alternating optimization is an effective DPO regularization. Table 4 shows an ablation study comparing different DPO regularization techniques to combat deterioration in generation quality (Pal et al., 2024) during DPO training. Existing approaches like DPOP (Pal et al., 2024) and RPO (Liu et al., 2024b) provide marginal gains up to 1.62% for distractor generation and 0.32% for error generation. Our alternation optimization, switching between SFT and DPO objective, at either the per-batch or per-epoch level, leads to the best performance for both, DPO-GT and LOOKALIKE, with performance gains up to 1.96% on the distractor generation task and 7.52% on the error generation task. These results show that alternating optimization effectively guides the LLM to periodically recalibrate to the ground-truth distribution (via SFT)

while remaining faithful to learning ranking preferences of consistent generations (via DPO).

5.2 Qualitative Case Studies

LOOKALIKE generates more consistent errors. Table 5 shows errors from LOOKALIKE compared to errors generated from SFT on two math questions. For the question on finding factors, SFT generates an overly generalized error applicable to many potential distractors, “Does not understand the term factor”. On the other hand, LOOKALIKE generates a more specific error, “When asked for factors of an algebraic expression, thinks any part of a term will be a factor”, consistent with the distractor. Similarly, for the question on simplifying algebraic terms, SFT generates an abstract error applicable to many distractors, “Tries to add or subtract unlike terms”. On the other hand, LOOKALIKE generates a more specific and consistent error leading to the input distractor, “When collecting like terms, treats subtractions as if they are additions.” We see similar patterns across other topics, with errors generated by LOOKALIKE being more specific and consistent with the input question and distractor. We also show qualitative examples of generated errors across all methods in Appendix D.

Error Analysis of LOOKALIKE. While LOOKALIKE outperforms SFT in generating more consistent errors and distractors, we observe some examples of generated errors that are inconsistent with the input question-distractor pair. One failure pattern observed is of *template overfitting*, where LOOKALIKE generates an error by overfitting to the error-distractor template of a similar question seen during training, generating errors that are consistent with other distractors from similar questions but not the input distractor. Table 8 in the Appendix shows two examples. We see that the generated error, “Has multiplied by the root power”, is inconsistent with the input distractor 64, but upon inspection, is present as a ground-truth error and consistent with another question-distractor pair on the same topic.

5.3 Human Evaluation

Setup. We conduct a human evaluation on the quality and consistency of generated errors. We instruct two independent annotators with teaching experience to evaluate whether an error is consistent with a given input math question and corresponding distractor, choosing between a) yes, b) partially,

and c) no. Our instructions to human annotators are provided in Appendix F.

We randomly select 40 math questions from our test set spanning a diverse range of topics. For each question, we include its ground-truth human-written error, the error generated by SFT, and the error generated by LOOKALIKE, for human evaluation. This process results in 120 errors, along with their corresponding questions and distractors, for human evaluation. We shuffle the 120 samples to avoid annotator bias.

Results. Table 6 shows the average of annotators’ ordinal ratings on error explanations from the ground truth, SFT, and LOOKALIKE models. Ground truth errors scored the highest (mean = 0.812), followed by LOOKALIKE (0.587), and SFT (0.400). While LOOKALIKE does not match the human-authored ground truth, it significantly outperforms SFT on average, suggesting that preference-based regularization leads to more pedagogically consistent explanations.

We also measured agreement between annotators using quadratic-weighted Cohen’s kappa, and found that error labels generated by LOOKALIKE led to the highest agreement (0.740), surpassing both SFT (0.659) and even the inter-annotator agreement on ground truth labels (0.415). This result suggests that errors generated by LOOKALIKE are easier for humans to interpret consistently, even if they are not always as plausible as ground truth explanations. We see a lower agreement on ground truth errors because their pedagogical nuance and potential generality made consistency judgments more subjective for annotators compared to the often more literal AI-generated errors.

Finally, we compared agreement between evaluations from human annotators to evaluations from GPT-4o-mini-based LLM-as-Judge, our reference metric for error generation. Agreement varied between annotators, with the first annotator showing moderate agreement (linear Kappa) with GPT-4o-mini (0.556 for LOOKALIKE-generated errors and 0.505 for SFT-generated errors), and the second annotator showing low agreement (0.314 for LOOKALIKE-generated errors and 0.409 for SFT-generated errors).

6 Related Work

Error-Distractor Generation for Math MCQs. Automated generation of math MCQs, and particularly their distractors, has progressed from

Topic	Finding factors	Simplifying terms
Question Stem	Which of the following is a factor of: $6n^2 - 9$?	Simplify the following expression by collecting like terms: $6x - 2y - x + 3y$.
Key	3	$5x + y$
Ground-truth Distractor	9	$7x + 5y$
Ground-truth Error	When asked for factors of an algebraic expression, thinks a term will be a factor.	When collecting like terms, treats subtractions as if they are additions.
SFT-Generated Error	Does not understand the term factor.	Tries to add or subtract unlike terms.
LOOKALIKE-Generated Error	When asked for factors of an algebraic expression, thinks any part of a term will be a factor.	When collecting like terms, treats subtractions as if they are additions.

Table 5: Examples showing errors generated from LOOKALIKE are more consistent than errors generated by SFT.

	Human	SFT	LOOKALIKE
Avg. Rating	0.812	0.400	0.587

Table 6: Average error consistency rating by human evaluators. LOOKALIKE generates more consistent errors than SFT.

template-based (rule-based and constraint-based) methods (Shin et al., 2019; Liang et al., 2018; Luo et al., 2024) to Large Language Model (LLM) approaches (Fernandez et al., 2024; Feng et al., 2024; Scarlatos et al., 2024a; Bitew et al., 2023; Chung et al., 2020). A critical challenge, however, remains the generation of high-quality distractors that accurately reflect common student errors and misconceptions (Alhazmi et al., 2024; Stasaski and Hearst, 2017). Current methods advance error representation using variational techniques (Fernandez et al., 2024), RAG-based methods (Yu et al., 2024a), and knowledge-bases (Ren and Q. Zhu, 2021).

Preference Optimization in Education. Preference learning techniques, including Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and its more stable, computationally efficient alternative Direct Preference Optimization (DPO) (Rafailov et al., 2023), are vital for aligning AI outputs with human judgments in education (Fahad Mon et al., 2023). Many recent approaches have used DPO (Lee et al., 2025; Sonkar et al., 2024b; Team et al., 2024; Ashok Kumar and Lan, 2024; Scarlatos et al., 2024b, 2025) but they do not handle some known failure modes of DPO related to inconsistent or out-of-distribution generation which the synthetic data generation of LOOKALIKE utilizes and the regularization of LOOKALIKE addresses. Other works mitigate these issues by providing regularization by using entropy

(Shekhar et al., 2024), length-based rewards (Park et al., 2024), or the SFT objective (Liu et al., 2024b; Pal et al., 2024), LOOKALIKE improved on these by providing a simpler SFT-based regularization approach which requires less hyperparameter tuning and is easier to apply.

Challenges in Erroneous Instruction Following.

Generating distractors from error descriptions, is an instance of the broader challenge of AI instruction following (Lou et al., 2024). AI systems, including LLMs, struggle with complex reasoning (Heo et al., 2024; Son et al., 2024), multi-step tasks (Chen et al., 2024; Wang and Lu, 2023; Fujisawa et al., 2024), and adhering to multiple constraints simultaneously (Wen et al., 2024), sometimes exhibiting a "curse of instructions" where performance degrades as complexity increases (Jang et al., 2022; Son et al., 2024). Generalization also poses a significant hurdle; models often fail to apply instructions to new tasks or in novel combinations (compositional generalization) (Cohen et al., 2025; Dan et al., 2021). These challenges can lead to inconsistencies where the generated output does not faithfully reflect the nuances of the input instruction (Jang et al., 2022; Son et al., 2024; Heo et al., 2024), a problem LOOKALIKE aims to mitigate in the context of error-distractor generation through targeted preference optimization.

7 Conclusion

In this paper, we introduced LOOKALIKE, a method that improves error-distractor consistency in math MCQs via preference optimization. LOOKALIKE uses two main innovations: a) mining synthetic preference pairs from model generation inconsistencies and b) alternating optimization by switching between SFT and DPO objectives to

stabilize training. Through extensive experiments on a real-world math MCQ dataset, we showed that LOOKALIKE outperforms the previous state-of-the-art method by a wide margin on both error generation and distractor generation. These improvements highlighted the potential of inconsistency mining and preference-based regularization for generating consistent math MCQ distractors at scale. We identify several limitations and avenues for future work. First, while LOOKALIKE improves error and distractor generation consistency, examples of inconsistent generations remain. Ideas for creating preference pairs using error generation and distractor generation models together could be a promising direction. Second, testing the generalizability of LOOKALIKE to math MCQs from unseen topics remains unexplored.

Limitations

While LOOKALIKE demonstrates improvements in generating consistent error-distractor pairs, it currently operates within the domain of middle-school mathematics. Extending the approach to other subjects like science or language arts may require minor modifications to the error and distractor representations.

Additionally, the current preference mining strategy relies on model-generated inconsistencies, which assumes the base model is sufficiently trained to surface pedagogically meaningful contrastive samples. In practice, we find that models pretrained on math data (e.g., MetaMath) meet this assumption, suggesting this is a broadly applicable approach rather than a bottleneck.

Our use of exact match to label non-matching outputs as dispreferred is conservative and intentionally strict; it helps emphasize high-confidence inconsistencies. Nonetheless, exploring softer similarity-based criteria or human judgments to refine preference mining is a valuable future direction.

Ethical Considerations

Our goal is to reduce educator workload by automating the generation of plausible distractors and their associated misconceptions, ultimately supporting teachers in providing more personalized feedback. However, we acknowledge a potential concern around over-reliance on AI-generated content in educational settings. While our system is designed to assist, not replace, educators, thoughtful deployment practices and educator-in-the-loop designs are encouraged.

The use of large language models (LLMs) introduces the standard risks of inherited biases or artifacts from pretraining data. In our case, these risks are minimal, as the domain of application (mathematical misconceptions) is highly constrained and less prone to sociolinguistic biases. Nevertheless, we encourage ongoing validation and periodic audits as best practices when deploying AI systems in learning environments.

Acknowledgments

This work is partially supported by Renaissance Philanthropy via the learning engineering virtual institute (LEVI) and NSF grants 2118706, 2237676, and 2341948. We thank Hasnain Heickal and

Zhangqi Duan for helpful discussions and annotations regarding this work.

References

- Peter Airasian. 2001. Classroom assessment: Concepts and applications. *McGraw-Hill, Ohio, USA*.
- Elaf Alhazmi, Quan Z. Sheng, Wei Emma Zhang, Munazza Zaib, and Ahoud Alhazmi. 2024. Distractor generation in multiple-choice tasks: A survey of methods, datasets, and evaluation.
- 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.
- Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Distractor generation for multiple-choice questions with predictive prompting and large language models.
- Xinyi Chen, Baohao Liao, Jirui Qi, Panagiotis Eustratiadis, Christof Monz, Arianna Bisazza, and Maarten de Rijke. 2024. The SIFo benchmark: Investigating the sequential instruction following ability of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics.
- Ho-Lam Chung, Ying-Hong Chan, and Yao-Chung Fan. 2020. A BERT-based distractor generation scheme with multi-tasking and negative answer training strategies. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.
- Vanya Cohen, Geraud Nangué Tasse, Nakul Gopalan, Steven James, Matthew Gombolay, Ray Mooney, and Benjamin Rosman. 2025. Compositional instruction following with language models and reinforcement learning.
- Soham Dan, Xinran Han, and Dan Roth. 2021. Compositional data and task augmentation for instruction following. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.
- Bisni Fahad Mon, Asma Wasfi, Mohammad Hayajneh, Ahmad Slim, and Najah Abu Ali. 2023. Reinforcement learning in education: A literature review. *Informatics*.
- Wanyong Feng, Jaewook Lee, Hunter McNichols, Alexander Scarlatos, Digory Smith, Simon Woodhead, Nancy Ornelas, and Andrew Lan. 2024. Exploring automated distractor generation for math multiple-choice questions via large language models. In *Findings of the Association for Computational*

- Linguistics: NAACL 2024*. Association for Computational Linguistics.
- Nigel Fernandez, Alexander Scarlatos, Wanyong Feng, Simon Woodhead, and Andrew Lan. 2024. DiVERT: Distractor generation with variational errors represented as text for math multiple-choice questions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Ipei Fujisawa, Sensho Nobe, Hiroki Seto, Rina Onda, Yoshiaki Uchida, Hiroki Ikoma, Pei-Chun Chien, and Ryota Kanai. 2024. Procbench: Benchmark for multi-step reasoning and following procedure.
- Juyeon Heo, Miao Xiong, Christina Heinze-Deml, and Jaya Narain. 2024. Do LLMs estimate uncertainty well in instruction-following? In *Neurips Safe Generative AI Workshop 2024*.
- 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*.
- Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2022. Can large language models truly follow your instructions? In *NeurIPS ML Safety Workshop*.
- Tom Kubiszyn and Gary Borich. 2016. Educational testing and measurement. *John Wiley and Sons, New Jersey, USA*.
- Yooseop Lee, Suin Kim, and Yohan Jo. 2025. Generating plausible distractors for multiple-choice questions via student choice prediction.
- Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C. Lee Giles. 2018. Distractor generation for multiple choice questions using learning to rank. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics.
- Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2024a. Conflict-averse gradient descent for multi-task learning.
- Yang Liu, Dan Iter, Yichong Xu, Shuhang 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*. Association for Computational Linguistics.
- Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. 2024b. Provably mitigating overoptimization in RLHF: Your SFT loss is implicitly an adversarial regularizer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Renze Lou, Kai Zhang, and Wenpeng Yin. 2024. Large language model instruction following: A survey of progresses and challenges.
- 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)*. Association for Computational Linguistics.
- Anthony J. Nitko. 1996. Educational assessment of students. *Prentice-Hall, Iowa, USA*.
- 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.
- Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddhartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*.
- Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Siyu Ren and Kenny Q. Zhu. 2021. Knowledge-driven distractor generation for cloze-style multiple choice questions. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Denise Rey and Markus Neuhäuser. 2011. *Wilcoxon-Signed-Rank Test*. Springer Berlin Heidelberg.
- 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 Scarlato, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025. [Training llm-based tutors to improve student learning outcomes in dialogues](#). *Preprint*, arXiv:2503.06424.
- Alexander Scarlato, Digory Smith, Simon Woodhead, and Andrew Lan. 2024b. Improving the validity of automatically generated feedback via reinforcement learning. In *Artificial Intelligence in Education*, pages 280–294, Cham. Springer Nature Switzerland.
- Shivanshu Shekhar, Shreyas Singh, and Tong Zhang. 2024. See-dpo: Self entropy enhanced direct preference optimization.
- Guangyuan Shi, Qimai Li, Wenlong Zhang, Jiabin Chen, and Xiao-Ming Wu. 2023. Recon: Reducing conflicting gradients from the root for multi-task learning. In *The Eleventh International Conference on Learning Representations*.
- Jinnie Shin, Qi Guo, and Mark J. Gierl. 2019. Multiple-choice item distractor development using topic modeling approaches. *Frontiers in Psychology*, Volume 10 - 2019.
- Mark G Simkin and William L Kuechler. 2005. Multiple-choice tests and student understanding: What is the connection? *Decision Sciences Journal of Innovative Education*, 3(1):73–98.
- Guijin Son, Sangwon Baek, Sangdae Nam, Ilgyun Jeong, and Seungone Kim. 2024. Multi-task inference: Can large language models follow multiple instructions at once? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- Shashank Sonkar, Naiming Liu, MyCo Le, and Richard Baraniuk. 2024a. Malalgoqa: Pedagogical evaluation of counterfactual reasoning in large language models and implications for ai in education. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15554–15567.
- Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, and Richard Baraniuk. 2024b. Pedagogical alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.
- Katherine Stasaski and Marti A. Hearst. 2017. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.
- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansoor Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- LearnLM Team, Abhinav Modi, Aditya Srikanth Veerubhotla, Aliya Rysbek, Andrea Huber, Brett Wiltshire, Brian Veprek, Daniel Gillick, Daniel Kasenberg, Derek Ahmed, Irina Jurenka, James Cohan, Jennifer She, Julia Wilkowsky, Kaiz Alarakya, Kevin R. McKee, Lisa Wang, Markus Kunesch, Mike Schaeckermann, and 27 others. 2024. Learnlm: Improving gemini for learning.
- Danielle R Thomas, Conrad Borchers, Sanjit Kakarla, Jionghao Lin, Shambhavi Bhushan, Boyuan Guo, Erin Gatz, and Kenneth R Koedinger. 2025. Does multiple choice have a future in the age of generative ai? a posttest-only rct. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pages 494–504.
- Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. [Diverse beam search: Decoding diverse solutions from neural sequence models](#). *Preprint*, arXiv:1610.02424.
- Tianduo Wang and Wei Lu. 2023. Learning multi-step reasoning by solving arithmetic tasks.
- Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuan Li, Binxin Hu, Wendy Gao, Jiaying Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. 2024. Benchmarking complex instruction-following with multiple constraints composition. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- 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, 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*. Association for Computational Linguistics.
- Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. In *Proceedings of the 41st International Conference on Machine Learning*.
- Yuzi Yan, Yibo Miao, Jialian Li, Yipin Zhang, Jian Xie, Zhijie Deng, and Dong Yan. 2025. [3d-properties: Identifying challenges in DPO and charting a path](#)

forward. In *The Thirteenth International Conference on Learning Representations*.

Han Cheng Yu, Yu An Shih, Kin Man Law, KaiYu Hsieh, Yu Chen Cheng, Hsin Chih Ho, Zih An Lin, Wen-Chuan Hsu, and Yao-Chung Fan. 2024a. Enhancing distractor generation for multiple-choice questions with retrieval augmented pretraining and knowledge graph integration. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024b. MetaMath: Bootstrap your own mathematical questions for large language models.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BertScore: Evaluating text generation with bert.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

A Baselines and their Hyperparameters

We describe LOOKALIKE’s baselines, as well as the hyperparameters used by LOOKALIKE and its baselines. We use MetaMath-Mistral 7B (Yu et al., 2024b) as our base LLM backbone for error and distractor generation across methods. For memory efficiency, we quantize the model weights into 8-bit integer representation and enable gradient checkpointing throughout training. Our implementation utilizes the HuggingFace ecosystem, specifically the transformers (Wolf et al., 2020), peft, and trl libraries for finetuning. We perform training on NVIDIA L40 GPUs.

SFT. For the supervised finetuning (SFT) baseline we train the base model with Low-Rank Adaptation (LoRA) modules (Hu et al., 2022). LoRA is configured with a rank $r = 128$, $\alpha = 256$, and a dropout rate of 0.05. We perform SFT training for 5 epochs, with early stopping based on validation loss. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $2e-5$. We use a batch size of 6.

DPO-based Baselines. For all DPO training, we set the hyperparameter $\beta = 0.5$ and the learning rate as $5e-6$. We use a batch size of 6.

DPO-GT. As specified in 2.2 we have multiple errors and distractors associated with all questions, to create preference pairs for each pair of error and distractor, we place all the non-associated sample of either in the dispreferred pair while placing the specified samples in the preferred pairs.

RPO. For RPO (Section 2.3), we use $\lambda = 0.005$ as reported by them. We use the default implementations of RPO as provided in the trl library.

DPOP DPO-Positive (DPOP) (Pal et al., 2024) enhances DPO by preventing the model from merely reducing the likelihood of rejected examples where the edit distance in all pairs is large by using the SFT objective as a penalty. It introduces a constraint term to balance learning:

$$L_{DPOP} = L_{DPO} - \lambda \cdot \max(0, \log \frac{\pi_{ref}(y_w|x)}{\pi_{\theta}(y_w|x)}). \quad (4)$$

Here, we use $\lambda = 0.1$.

LOOKALIKE (Synthetic Data Generation).

For the LOOKALIKE preference pairs (in Section 3.1) we generate 3 errors and distractors for each epoch of training to create negative preference samples, while considering the ground truth errors and distractors as the positive preference samples. We consider the top- k completions returned by beam search to get a set of \hat{e}_i which augments the set of dispreferred responses further. We note that for all DPO training we use the SFT trained model as a warm start as with previous literature (Rafailov et al., 2023).

LOOKALIKE (Per-epoch and Per-batch Regularization). With the per-epoch and per-batch modes of LOOKALIKE (Section 3.2), we use the learning rate of $5e-6$ for both DPO and SFT. For the per-epoch setting we perform one entire epoch of SFT after one epoch of DPO. Whereas for the per-batch setting if we run out of SFT batches while DPO training hasn’t finished we rollback to the beginning of the SFT training data.

B LLM-as-a-judge

To assess whether two error explanations express the same underlying misconception, we use GPT-4o-mini as an automated judge. The model is provided with the question, distractor, and two error explanations, and asked to determine whether they are *mathematically equivalent* (Table 7), that is, whether they arise from the same conceptual

misunderstanding, regardless of wording. Below, we present an example of the prompt used in this evaluation.

This template was used for all pairwise comparisons of error explanations in the LLM-as-a-Judge evaluation.

C Error Analysis

While LOOKALIKE generally produces more specific and grounded error explanations, Table 8 also reveals some notable limitations. In the cube root example, the explanation “Has multiplied by the root power” reflects a plausible arithmetic confusion but doesn’t clearly connect to the distractor value of 64, which results from cubing rather than misunderstanding cube roots. Similarly, in the number ordering case, the generated error implies digit-level misordering but lacks clarity on how this leads specifically to choosing “Only Katie.” These examples suggest that while LOOKALIKE often captures fine-grained misconceptions, it can occasionally overgeneralize or introduce speculative reasoning not fully aligned with the distractor. This underscores the need for further refinement to ensure tighter alignment between the error explanation and the underlying choice.

D Comparing Errors across LOOKALIKE and its Baselines

Table 9 illustrates how different training methods produce qualitatively distinct reasoning errors across representative math questions. We observe a clear progression in the nature of these errors, reflecting the underlying supervision strategies. Models trained with SFT often generate surface-level mistakes indicative of limited conceptual understanding. In contrast, DiVERT tends to produce more structured but still incorrect procedural reasoning. Errors from DPO-GT reveal partial application of mathematical heuristics, suggesting more sophisticated—though still flawed—mental models. Finally, LOOKALIKE models (both per batch and per epoch) consistently produce errors that resemble common student misconceptions, such as overgeneralizing valid procedures or subtly misapplying familiar rules. This progression supports our claim that LOOKALIKE encourages more pedagogically meaningful error patterns, aligning closely with authentic human reasoning.

E Example MCQs from Real-world Math MCQ Dataset

We show example MCQs from the dataset in Table 10.

F Human Analysis Instructions

To evaluate the consistency of error explanations with corresponding distractor choices in multiple-choice math questions, we provided annotators with detailed guidelines, shown in Table 11. Annotators were instructed to examine each question item, which included a correct answer, a step-by-step solution, a distractor (incorrect answer), and an explanation for why a student might choose that distractor.

Annotators were asked to judge whether the explanation was:

- Yes: Clearly consistent with the distractor and plausibly explains the student error.
- Partially: Somewhat consistent, but vague, generic, or only loosely related to the distractor.
- No: Inconsistent or misleading; does not plausibly explain the choice of the distractor.

The instructions included concrete examples for each category to help calibrate judgment and ensure consistent annotation. These annotations were later used to analyze the quality of generated error explanations.

System Prompt.

You are a math education expert.

Given a question and a distractor (an incorrect student answer), determine whether two error descriptions are *mathematically equivalent*.

Definitions.

- An incorrect answer or distractor is a plausible but incorrect answer choice to the specified question.
- An error explanation or error is the misconception a student might make that leads them to choosing the specified distractor.
- Two error explanations are *mathematically equivalent* if they stem from the same core misunderstanding, regardless of wording.

Your response should include a brief justification (1–2 sentences) for whether the errors reflect the same or different misconceptions.

Always conclude with: “**Answer: Equivalent** or **Answer: Not Equivalent**”.

Question and Metadata.

The question is: <Question>

The question topic is: <Topic>

The question concept is: <Concept>

The solution is: <Solution from question to Correct Answer>

The correct answer is: <Correct Answer>

Distractor (incorrect answer): <Ground Truth Distractor>

Error explanation 1: <Ground Truth Error>

Error explanation 2: <Generated Error>

Table 7: System prompt used to evaluate the mathematical equivalence of error explanations for a given distractor. The prompt positions the model as a math education expert tasked with identifying whether two misconceptions arise from the same underlying error.

Field	Cube Root	Indices, Powers and Roots
Question	$\sqrt[3]{8} = ?$	$3.52 + 2.75 =$
Distractor	64	5.27
Correct Answer	2	6.27
SFT Error Explanation	Divides by the order of the root.	Does not understand place value within a number.
LOOKALIKE Error Explanation	Has multiplied by the root power.	When adding decimals with a different number of decimal places, lines up the digits incorrectly.

Table 8: Comparison of error explanations for two different math topics. Examples show that LOOKALIKE also has some failure modes, discussed in greater depth in Section 5.2.

	Improper Fraction Conversion	Gradient of a Line
Question	Convert this into an improper fraction: $4\frac{2}{3}$	What is the gradient of this line? $5x + 3y = 15$
Correct Answer	$\frac{14}{3}$	$-\frac{5}{3}$
Ground-truth Distractor	$\frac{12}{3}$	$\frac{5}{3}$
Ground-truth Error	Forgetting to add the numerator to the whole part.	Applying the same operation instead of the inverse when solving.
SFT	Does not add the whole to the numerator when converting a mixed number to an improper fraction.	Believes a downward line has a positive gradient.
DiVERT	Thinks you add the number of wholes to the numerator when converting a mixed number to an improper fraction.	When solving an equation, uses the same operation rather than the inverse.
DPO-GT + Per batch	Does not include the whole amount when converting a mixed number to an improper fraction.	Believes the gradient of a line is given by the coefficient of x , even when the equation is not in the form $y = mx + c$.
LOOKALIKE + Per batch	Thinks you add the number of wholes to the numerator when converting a mixed number to an improper fraction.	When solving an equation, uses the same operation rather than the inverse.
DPO-GT + Per epoch	Thinks you can ignore the whole amount when converting a mixed number to an improper fraction.	When finding the gradient from the equation of a line in the form $ax + by = c$, believes b/a is the gradient.
LOOKALIKE + Per epoch	Thinks you add the number of wholes to the numerator when converting a mixed number to an improper fraction.	When finding the gradient from the equation of a line in the form $ax + by = c$, believes b/a is the gradient.

Table 9: Comparison of typical errors generated by each method for two representative math questions.

Question stem	Add brackets to this calculation to make the answer 7. $16 - 6 + 4 \div 2$
Topic	BIDMAS
Concept	Insert brackets to make a calculation correct
Solution	Inside the bracket we work left to right, so we get $14 \div 2$ which is 7.
Correct answer	$(16 - 6 + 4) \div 2$
Distractor 1	$16 - (6 + 4) \div 2$
Error 1	With order of operations brackets are done first, then division is done before subtraction. This would give us $16 - 10 \div 2 = 16 - 5 = 11$ NOT 7.
Distractor 2	$(16 - 6) + \frac{4}{2}$
Error 2	With order of operations brackets are done first, then division is done before subtraction. This would give us $10 + 4 \div 2 = 10 + 2 = 12$ NOT 7.
Distractor 3	$16 - 6 + (\frac{4}{2})$
Error 3	With order of operations brackets are done first, then division is done before subtraction. Putting the brackets around the division, will not change the order. $16 - 6 + (4 \div 2) = 16 - 6 + 2 = 12$ NOT 7.
Question stem	Which of the following answers gives the correct solutions to the quadratic expression below? $(x + 2)(x - 7) = 0$
Topic	Algebra
Concept	Solve quadratic equations using factorisation in the form $(x + a)(x + b)$
Solution	Setting each bracket equal to 0 we have $x + 2 = 0$ and $x - 7 = 0$. This tells us that $x = -2$ and $x = 7$.
Correct answer	$x = -2, x = 7$
Distractor 1	$x = 2, x = -7$
Error 1	Believes the solutions of a quadratic equation are the constants in the factorised form
Distractor 2	$x = 2, x = 7$
Error 2	Believes the solutions of a quadratic equation are the absolute values of the constants in the factorised form
Distractor 3	$x = -2, x = -7$
Error 3	Believes the solutions of a quadratic equation are the negative of the absolute values of the constants in the factorised form

Table 10: Example MCQs from the real-world math MCQ dataset.

In this task, you'll evaluate error explanations for student errors in math multiple-choice questions. For each item, you'll see:

1. The **question**
2. The **correct answer** choice
3. A **solution** which shows how a student can reach the correct answer choice
4. A **distractor** (an incorrect answer choice)
5. An **error** explanation describing why a student might choose the distractor

Your Task

Annotate if each error explanation is consistent with the distractor (mark Yes), is generic, vague, or partially consistent (mark Partially) or has nothing to do with the distractor, or is misleading (mark No).

Use your best judgment when assigning ratings. Some examples are:

Example 1 (Marking **Yes**):

Question: Add brackets to this calculation to make the answer 7. $16 - 6 + 4 \div 2$

Correct Answer: $(16 - 6 + 4) \div 2$

Solution: Inside the bracket we work left to right, so we get $14 \div 2$ which is 7.

Distractor: $16 - (6 + 4) \div 2$

Error: Carries out operations from left to right regardless of priority order.

Mark **Yes**

Example 2 (Marking **Partially**):

Question: $\frac{3}{7}$ of a group of students are boys. What would be a possible ratio of boys to girls?

Correct Answer: 3 : 4

Solution: For every 7 students, 3 are boys and 4 are girls. The ratio is then 3:4.

Distractor: 3 : 7

Error: Uses the denominator when converting from fractions to ratio, rather than numerator.

Mark **Partially**

Example 3 (Marking **No**):

Question: When $h = 5$ $h^2 =$

Correct Answer: 25

Solution: If $h = 5$, $h^2 = h \times h = 5 \times 5 = 25$.

Distractor: 7

Error: Multiplies by the index.

Mark **No**

Table 11: Instructions provided to human annotators used to evaluate the consistency of error explanations for a given distractor.

You Shall Know a Word’s Difficulty by the Family It Keeps: Word Family Features in Personalised Word Difficulty Classifiers for L2 Spanish

Jasper Degraeuwe

Ghent University (LT³ / MULTIPLES research groups)

Groot-Brittanniëlaan 45, 9000 Ghent, Belgium

Jasper.Degraeuwe@UGent.be

Abstract

Designing vocabulary learning activities for foreign/second language (L2) learners highly depends on the successful identification of difficult words. In this paper, we present a novel personalised word difficulty classifier for L2 Spanish, using the LexComSpaL2 corpus as training data and a BiLSTM model as the architecture. We train a *base* version (using the original LexComSpaL2 data) and a *word family* version of the classifier (adding word family knowledge as an extra feature). The base version obtains reasonably good performance ($F1 = 0.53$) and shows weak positive predictive power ($\phi = 0.32$), underlining the potential of automated methods in determining vocabulary difficulty for individual L2 learners. The “word family classifier” is able to further push performance ($F1 = 0.62$ and $\phi = 0.45$), highlighting the value of well-chosen linguistic features in developing word difficulty classifiers.

1 Introduction

In the rapidly evolving digital era of the 21st century, language and technology are growing closer together than ever. Language technology tools – especially those driven by large language models (LLMs) – have become very adept at performing a wide range of tasks, ranging from summarising documents to translating texts from one language into another. In some cases, their output has even shown to be virtually indistinguishable from human-written materials (Else, 2023).

At the same time, despite the wealth of technological assistance, being able to understand and speak a foreign/second language (L2) can be said to remain an indispensable skill for anyone who wants to fully engage in foreign cultures, as building sustainable intercultural relationships involves tasks that are much more difficult to achieve by means of technological tools alone, such as interpreting humour and facial expressions (Godwin-Jones, 2019).

What language technology tools do possess, however, is the ability to play the role of a valuable assistant in the L2 learning process.

This interface between second language acquisition (SLA) and computer assistance has commonly been referred to as Computer-Assisted Language Learning or CALL. Recently, the field of CALL has witnessed a growing interest in the specific subdomain of Intelligent CALL (ICALL). With techniques coming from the field of Artificial Intelligence (AI) and its subdomain of Natural Language Processing (NLP) as their source of “intelligence”, ICALL environments aim to — among other goals — facilitate and/or (partially) automate the creation of language learning materials. Although the origins of ICALL can be traced back as far as to the 1980s (Nyns, 1989), it was not until the advent of static (Mikolov et al., 2013) and contextualised word embeddings (Devlin et al., 2019) followed by full-fledged generative LLMs that a true paradigm shift from CALL towards ICALL has started to take shape.

ICALL platforms can foster virtually any language skill, but in this paper we will specifically focus on ICALL for *vocabulary learning* purposes. With a large body of studies showing that text comprehension and vocabulary knowledge are positively correlated (e.g., Laufer and Ravenhorst-Kalovski, 2010; Schmitt et al., 2011), we know that a wide vocabulary is a fundamental requisite to be able to function in a language. Or, in the words of Wilkins (1972, p. 111): “without grammar very little can be conveyed, without vocabulary *nothing* can be conveyed”. To help learners expand their L2 vocabulary, a combination of implicit and explicit learning activities is to be recommended (Nation, 2019; Schmitt, 2010a). Explicit vocabulary learning activities (e.g., fill-in-the-blanks exercises) require paying deliberate attention to vocabulary items, while in implicit activities the increase in vocabulary knowledge is achieved as a by-product,

because the main goal of the activity is the successful completion of an authentic task such as understanding the plot of a book.

In both of these strands, knowing which words might be difficult for target learners to understand or produce is a valuable source of information. In the case of the implicit approach, one of the key notions is that learners acquire vocabulary when they are exposed to input that is comprehensible but slightly beyond their current knowledge (Lichtman and VanPatten, 2021), implying that it should be known which parts of the input are comprehensible and which are not. For explicit learning, on the other hand, informed decisions need to be made on which words to in- and exclude from the activities, a task that becomes considerably easier if we know which words are (un)known by the learners.

In other words, identifying difficult words (or word uses) plays a pivotal role during the development of vocabulary learning materials. As an alternative to the labour-intensive process of identifying these words by hand, research within the field of ICALL has explored NLP-driven approaches to perform this task. Methods exploiting computer-readable resources in which words are linked to difficulty levels (or frequency bands, since frequency correlates with difficulty¹; Schmitt, 2010b) constitute a first option, as they can automatically assign words in digital texts to their corresponding difficulty/frequency label (Finlayson et al., 2023).

However, apart from having limited coverage (only words included in the resources will be assigned a label), this approach does not take into account individual differences among learners. To overcome these limitations, machine learning systems can be designed, which offer more flexibility: in theory, they can classify any text, sentence or word into any set of difficulty levels, and tailor predictions to individual learner profiles (Tack, 2021).

In this paper, we present a first-of-its-kind individualised word difficulty classifier for L2 Spanish. As our training data, we make use of LexComSpaL2 (Lexical Complexity for Spanish L2; Degrauwe and Goethals, 2024), a publicly available dataset² containing 2,240 in-context target

¹It should be noted that this difficulty - frequency correlation does not mean that word difficulty *equals* word frequency. As shown in previous research (Pintard and François, 2020), word difficulty cannot be predicted by frequency values alone.

²The dataset is made available through a [GitHub repository](#) and was released under the [ODC-By license](#), which grants the right to freely use and adapt the data as long as any use of the dataset is adequately attributed.

words with the corresponding difficulty judgements of 26 L2 Spanish students. We compare the results of training two different versions of the classifier: a *base* version (only using the original LexComSpaL2 data) and a *word family* version (adding word family knowledge as an extra feature).

2 Related Research

This literature overview discusses lexical difficulty/complexity as defined in the field of linguistics in Section 2.1, the technique of lexical complexity prediction (i.e. the approach adopted to build LexComSpaL2) in Section 2.2, individualised learning in Section 2.3, and a detailed account of word families (used to create a “linguistically enriched” version of LexComSpaL2) in Section 2.4.

2.1 Difficulty and Complexity in SLA

In linguistics, the concept of “word difficulty/complexity” is usually subdivided into several dimensions, often dichotomous in nature. One of the most prominent distinctions is the one between *absolute* (or *objective*) and *relative* (or *agent-related*) complexity (Kortmann and Szmrecsanyi, 2012). In the former type, complexity is understood in terms of the linguistic properties of words, ranging from their length over the number of vowels and diphthongs they contain to their homonymic and/or polysemous character (i.e. the number of different meanings/senses they have). Especially the last feature plays an important role in an SLA setting, as lexically ambiguous items have shown to be more challenging to process and learn than single-meaning words (Bensoussan and Laufer, 1984).

Relative complexity (also denominated “difficulty”; Bulté et al., 2025), on the other hand, corresponds to the complexity as perceived by a particular language learner, meaning that psycholinguistic factors and world knowledge can come into play (North et al., 2023; Kortmann and Szmrecsanyi, 2012). In an L2 setting, an additional crucial factor in determining agent-related complexity is L1 influence, which can manifest itself through false friends (e.g., ES *listo* [‘ready’] - NL *list* [‘ruse, trick’]) or cognates (e.g., ES *individuo* - NL *individu* - EN *individual*).

2.2 Lexical Complexity Prediction

Computational approaches to identifying difficult/complex words focus on “operationalising” the abovementioned linguistic concepts. A crucial aspect of this operationalisation is the presence of

some kind of “inventory” in which words are linked to discrete difficulty/complexity labels. One possible way to build such inventories is exploiting computer-readable versions of graded vocabulary lists (Dang et al., 2017), frequency lists (Davies and Hayward Davies, 2018), or graded coursebooks (in which case words are assigned to the level at which they first occur; Alfter, 2021). Another approach is to collect human annotations, either through online (crowdsourcing) platforms (Shardlow et al., 2021) or by means of dedicated research experiments (Tack, 2021).

The dataset used in this study, LexComSpaL2, falls in the last category (for more details on the corpus, see Section 3.1.1). The LexComSpaL2 annotations were gathered according to the principles of lexical complexity prediction (LCP; see Table 1 for an example), a relatively new strand within the field of NLP that provides an alternative to the binary complex word identification (CWI) approach (which labels words as either complex or non-complex; Yimam et al., 2018).

By using a five-point scale going from “very easy” to “very difficult” (for the full descriptors, see Section 3.1.1), LCP not only yields more fine-grained judgements than the binary CWI labels, it

Sentence	
They do hold elections, but candidates have to be endorsed by the conservative clergy, so dissenters are by definition excluded.	
Target word	LCP Label
<i>do</i>	1
<i>hold</i>	2
<i>elections</i>	3
<i>candidates</i>	1
<i>have</i>	1
<i>be</i>	1
<i>endorsed</i>	4
<i>conservative</i>	2
<i>clergy</i>	5
<i>dissenters</i>	5
<i>definition</i>	1
<i>excluded</i>	2

Table 1: Fictitious example of LCP annotations. The target sentence is taken from the CompLex dataset, the first LCP corpus ever created (Shardlow et al., 2020). In line with the LexComSpaL2 corpus, only nouns, verbs, and adjectives are considered in the example.

also enables making predictions based on “comparative complexity” (i.e. whether a word is more or less complex than another target word; North et al., 2023). Importantly, the term “complexity” as used in the field of LCP represents an amalgam of the concepts of complexity and difficulty described in Section 2.1, as it refers to the difficulty an individual may experience in understanding a given word as a result of both their personal knowledge and a word’s linguistic properties (North et al., 2023). In this paper we adopt the same comprehensive definition but will give preference to the term “difficulty” instead of “complexity”, since the individualisation of the predictions puts slightly more emphasis on the (personal knowledge of the) learner than on the linguistic properties of the target words.

To the best of our knowledge, LexComSpaL2 is the only available LCP dataset that (1) specifically targets L2 learners and (2) enables training personalised word difficulty classifiers. Other LCP datasets are mostly constructed for the purpose of training models that can be integrated in a lexical simplification pipeline (Paetzold and Specia, 2017). A comprehensive overview of existing LCP datasets can be found in Shardlow et al. (2024).

Regarding the features used in LCP classifiers, recent research has revealed that a hybrid approach combining linguistic features (see Section 2.1) and LLM embeddings (e.g., BERT embeddings; Devlin et al., 2019) results in the highest performance (Ortiz-Zambrano et al., 2025). Earlier research, however, showed that also with static word embeddings good performance levels can be achieved (Tack, 2021). In this paper, we build on this line of research by combining static word embeddings with linguistic information on word families. By focusing on word families we aim to gain new insights into the value of linguistic features in automated word difficulty prediction, as previous research has mainly paid attention to lexical features related to the word itself (e.g., word length and number of syllables) and semantic features taken from resources such as WordNet (Fellbaum, 1998; e.g., number of synonyms, hypernyms, and/or hyponyms of a given target word).

2.3 Individualised Learning

As already touched upon above, another core and unique aspect of our word difficulty classifier is that it outputs *personalised* predictions. This way, we aim to integrate findings from the literature on indi-

vidual differences in SLA³. In brief, research in this domain has demonstrated that a variety of factors related to the individual can impact the learning process and learning outcomes. As mentioned in Section 2.1, a first crucial dimension of individual differences is the linguistic background of the learner, particularly (proficiency in) their L1 and experience with (learning) other languages (Degani and Goldberg, 2019). Other individual differences that have a considerable impact on the vocabulary learning process of L2 learners include cognitive factors such as memory capacity (Martin and Ellis, 2012) and the degree of out-of-school exposure to the L2 (De Wilde et al., 2022).

The domains of ICALL and NLP started to devote increasingly more attention to individualising system outputs. The most comprehensive approach to personalising the L2 learning process can be found in Intelligent Language Tutoring/Teaching Systems (ILTSs), which tailor learning materials to the specific needs of individual users on a macro (selecting and sequencing activities) and/or micro level (providing scaffolded feedback) (Meurers et al., 2019; Ruiz et al., 2023). Regarding word difficulty prediction, both in the domain of CWI (Gooding and Tragut, 2022; Tack, 2021) and LCP (Degraeuwe and Goethals, 2024; North et al., 2023) efforts have been undertaken to adopt a learner-centred and personalised perspective. In the present study, we aim to continue this line of research.

2.4 Word Families

Finally, we briefly discuss the concept of word families (Bauer and Nation, 1993), based on which we expanded the LexComSpaL2 dataset and trained a separate version of the classifier. As defined by Webb (2021, p. 941), “[w]ord families are made up of a headword, its inflections, and derivations”. For the headword *address*, for example, this means that the word family consists of both the nominal (*addresses*) and verbal inflections (*addresses*, *addressed*, *addressing*), as well as derivations of the two (e.g., *addressee*, *readdress*, *unaddressed*) plus their inflected forms (e.g., *addressees*, *addresses*, *readdressed*). Supported by empirical evidence from cognitive linguistics (Zhang and Lin, 2021), one of the main arguments in favour of using word family information in an L2 learning setting is that, once learners have acquired knowledge of the form-meaning connection of a given family mem-

ber (e.g., *legal*), they can use their knowledge of the morphological system to infer the meaning of other members of the family (e.g., *legally*, *illegal*) (Nation and Webb, 2011; Nation, 2016).

3 Methodology

The methodology consists of two main steps: (1) the preparation of the dataset on which the different versions of the classifier should be trained (Section 3.1) and (2) the actual development and training of the classifier (Section 3.2).

3.1 Dataset Preparation

3.1.1 Original LexComSpaL2 Dataset

To train the base version of the personalised word difficulty classifier, we used the LexComSpaL2 dataset in its original format (Degraeuwe and Goethals, 2024; see Table 2 for a dataset sample). LexComSpaL2 includes 2,240 target words distributed over 200 sentences coming from four different domains (economics, health, law, and migration). The sentences were selected from L1 newspaper corpora using a dedicated method specifically designed to extract pedagogically suitable sentences from corpus data (Pilán et al., 2016). Regarding the annotations, 26 L2 Spanish learners (from different proficiency levels but all L1 Dutch) were asked to rate the (in-context) difficulty of all nouns, verbs, and adjectives in the 200 sentences according to the five-point LCP scale. Importantly, Degraeuwe and Goethals (2024) tailored the original LCP descriptors to L2 learners as the target audience by projecting the LCP labels onto the vocabulary knowledge continuum (Schmitt, 2019), which conceptualises vocabulary knowledge as a construct that gradually moves from “no knowledge” over “receptive mastery” to “productive mastery” (see Table 3 for the adapted scale).

In summary, the 58,240 self-perceived judgements of word difficulty included in LexComSpaL2 constitute relevant and representative data to train personalised word difficulty classifiers for L2 learners, as the annotations were (1) provided by actual L2 learners and (2) taken from pedagogically suitable sentences that were selected in an attempt to mimic the often thematic organisation of real-life vocabulary learning courses and materials.

3.1.2 Word Family-Enriched Dataset

To enrich the original LexComSpaL2 dataset with word family information, we considered the following three word family levels: the word’s **token**

³For extensive overviews of this domain, we refer to Dörnyei (2014) or Skehan (1991).

Sentence ID	Sentence text	Target word	Individual judgements
1_1	El <u>directivo</u> , que ha <u>celebrado</u> un almuerzo de <u>Navidad</u> con la prensa, ha asegurado que [...] ('The manager, who has held a Christmas lunch with the press, has assured that [...]')	directivo	{P1: 3, P2: 2, P3: 2, [...], P24: 3, P25: 1, P26: 1}
		celebrado	{P1: 2, P2: 1, P3: 1, [...], P24: 2, P25: 1, P26: 1}
		...	
...			
4_50	Las investigaciones sobre <u>atención</u> <u>primaria</u> , <u>neurología</u> , <u>oncología</u> <u>médica</u> y <u>microbiología</u> <u>van</u> después, [...] ('Research into primary care, neurology, medical oncology and microbiology comes after, [...]')	investigaciones	{P1: 1, P2: 1, P3: 1, [...], P24: 1, P25: 1, P26: 1}
		atención	{P1: 2, P2: 1, P3: 1, [...], P24: 1, P25: 1, P26: 1}
		...	

Table 2: Sample from the LexComSpaL2 corpus that was also presented in Degraeuwe and Goethals (2024). Aggregated judgements (per proficiency level and overall) were omitted from the sample, since we only used the individual judgements to train the classifier. Target words are underlined and “P” stands for participant.

Rating	Original LCP description	Adapted description
1	Very easy: this word is very familiar to me	I know this word and its meaning, and I also use it actively in speaking/writing.
2	Easy: I am aware of the meaning of this word	I know this word and its meaning, but I might not be able to use it on the top of my head in an oral/written conversation. When I have some time to think, however, I do think I would use it naturally.
3	Neutral: this word is neither difficult nor easy	I have heard/seen this word before and given the context I think that I more or less know what it means, but I do not see myself using this word actively.
4	Difficult: the meaning of this word is unclear to me, but I may be able to infer it from the sentence	This word sounds vaguely familiar and based on the context I could make an educated guess about its meaning, but I would still need a dictionary to be able to understand its exact meaning.
5	Very difficult: I have never seen this word before / this word is very unclear to me	This word does not sound familiar at all to me, and even based on the context I do not know what it means, so I would definitely need a dictionary to get to know its meaning.

Table 3: Original LCP descriptions compared to adapted descriptions proposed by Degraeuwe and Goethals (2024). The adapted descriptions are based on the vocabulary knowledge continuum (Schmitt, 2019).

form (also called the “type”), the word’s **lemma**, and the **source** from which the word’s lemma is derived (i.e. the “parent” of the lemma in the “family tree”)⁴. As an illustration, let us consider the word *desaparecido* (‘disappeared’): the **token** level consists of all occurrences of this exact word form in the LexComSpaL2 dataset; the **lemma** level corresponds to the lemma of *desaparecido* (i.e. the infinitive *desaparecer* - ‘to disappear’) and includes all of its other conjugated forms (e.g., *desaparezco*, *desaparecieron*); the **source** level corresponds to the “parent” of *desaparecer* (i.e. *aparecer* - ‘to appear’) and encompasses all inflected forms of this parent (e.g., *aparezco*, *aparecía*).

⁴The *token* and *lemma* levels correspond to, respectively, Level 1 and Level 2 of the Bauer and Nation (1993) taxonomy. The *source* level is specific to this study.

Next, we applied the following procedure to every target word in the LexComSpaL2 dataset:

1. Check if the exact **token** of the target word occurs more than once in the corpus. If so, we (1) calculate if there is a statistically significant difference ($p \leq 0.05$) between the annotations and (2) gather, for all participants individually, the lowest and highest annotated LCP value for the token in question.
2. Check if the **lemma** of the target word occurs more than once in the corpus. If so, we repeat the process described in the first step, but in this case for the target word’s lemma.
3. Check if the **source** of the target word’s lemma occurs more than once in the corpus. If so, we repeat the process described in the first

step for the source lemma. If the target word’s lemma is a headword, this step is skipped.

All words were disambiguated for part of speech using Stanza⁵, meaning that words such as *humano* (‘human’), which can both be a noun and adjective, constitute two different tokens. To compute statistical significance, we used the non-parametric Kruskal-Wallis H-test (Kruskal and Wallis, 1952), which can be applied to two or more samples and does not assume normally distributed data. To look up the source lemma of a given target word, we used the publicly available word family resource⁶ developed within the Spanish Corpus Annotation Project (Goethals, 2018).

We added the data to the original LexComSpaL2 corpus by creating three new versions of the corpus (one per word family level), in which we added four extra columns: one indicating if the target word occurs multiple times (*True* or *False* as value), one indicating if the annotations differ significantly (*True*, *False*, or *N/A* if the target word only occurs once), and two columns including the lowest and highest annotation per participant (or again *N/A*). This “word family-enriched” version of the dataset are made available as a part of the original LexComSpaL2 GitHub repository⁷.

The descriptive statistics of the word family enrichment are presented in Table 4. For the token and lemma levels, *#candidates* refers to the number of, respectively, unique tokens and lemmas that occur more than once in the corpus. For the source level, *#candidates* refers to the number of unique tokens whose source lemma also occurs in the corpus. The *#statSignDiff* column indicates for how many of those candidates the learners’ annotations differed significantly. Although these numbers are a by-product of the research and should also be interpreted as such, it does seem opportune to highlight that they seem to confirm as well as contradict the assumption that knowledge of one word family member means that learners also know the meaning of other family members (Section 2.4). At the token and lemma levels, the low number of statistically significant differences reveals that the annotations (and therefore also the degree of word knowledge) were consistent across the different occurrences. For the lemma level, this means that if learners acquired a certain degree of knowledge for

Level	#candidates	#statSignDiff	#enriched
Token	159	2	355 / 2,240
Lemma	273	6	632 / 2,240
Source	248	106	297 / 2,240

Table 4: Descriptive statistics of word family enrichment of LexComSpaL2. The *#enriched* column contains the number of target words for which information other than *N/A* was added.

one inflected form, they are highly likely to have the same degree of knowledge for other inflected forms. At the source level, however, we see that this conclusion does not hold, as in 106 of the 248 cases there was a statistically significant difference in the learners’ difficulty judgements. The results clearly indicate that knowledge acquired at the token level is not necessarily transferred to the source level in the family tree (or vice versa). Returning to the example above, this means that if learners know *desaparecido* it does not necessarily imply that they also know *aparecer* (or vice versa).

3.2 Classifier Training

3.2.1 Base Classifier

As the architecture for the base classifier, we used a Bidirectional Long Short-Term Memory (BiLSTM) model that follows a similar design as the CWI classifier presented in Tack (2021), who found that this type of neural network is able to personalise difficulty predictions. For each observation (i.e. a word linked to a learner’s LCP annotation), the base model takes the following features as input: a character embedding⁸, the word’s fastText embedding (Cañete, 2019), and the participant information (unique identifier, proficiency level, years of experience, and L1⁹). Based on a softmax activation function, the output layer yields a probability distribution for the different classes, with the class for which the highest probability is obtained being selected as the predicted difficulty level for the word in question. A simplified visualisation of the model’s architecture is presented in Figure 1 (for a full visualisation, see Appendix A) and the underlying code is made available in a GitHub repository¹⁰.

⁸Randomly initialised and trained with a convolutional neural network (De Hertog and Tack, 2018).

⁹It should be noted that, since all annotators in the LexComSpaL2 corpus have Dutch as their L1, this feature will not contribute to personalising the predictions.

¹⁰<https://github.com/JasperD-UGent/personalised-word-difficulty-classifier>

⁵<https://stanfordnlp.github.io/stanza/>

⁶<https://scap.ugent.be/overview-resources/>

⁷<https://github.com/JasperD-UGent/LexComSpaL2>

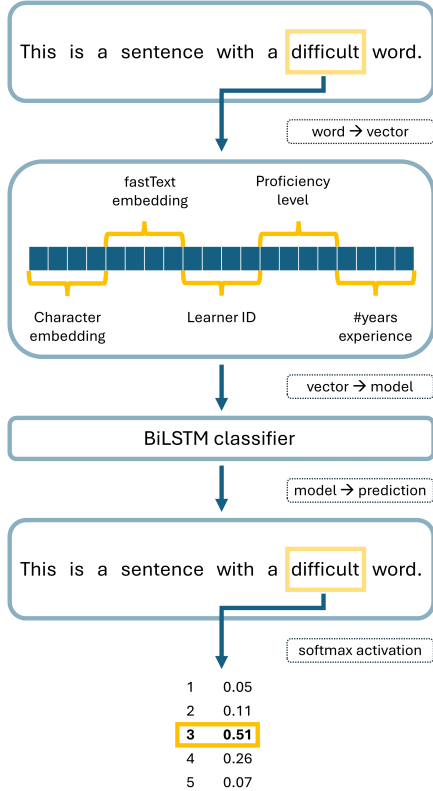


Figure 1: Simplified representation of BiLSTM word difficulty classifier.

To evaluate the model, a tenfold cross-validation setup was adopted. As our dataset split, we used the split added by Degrauwe and Goethals (2024) to the LexComSpaL2 repository in anticipation of the corpus being used to train future machine learning models. This sentence-level dataset split includes ten different “folds” of the LexComSpaL2 data into a training (160 sentences or 80%), validation, and test set (20 sentences or 10% each). The complete overview of the number of training instances per fold is included in Appendix B.

Regarding training parameters, we set the number of epochs to 50, the batch size to 64, and the loss strategy to *sparse_categorical_crossentropy*. *Adam* was used as the optimiser and an early stopping monitor on the validation loss (with a patience of 10) was added to the training process. In each cross-validation run, the weights for the training samples were calculated (Appendix C) and used for weighting the loss function. A mask was set to all “non-target words” (i.e. all tokens which are not a noun, verb, or adjective and thus did not receive a label during the data collection) and their input vectors and sample weight were set to 0. This way, the sentence context was still correctly represented

but the masked tokens were ignored during training. Finally, zero-padding (to the maximum sentence length of 35) was applied to all inputs and outputs.

3.2.2 Word Family-Enriched Classifier

The word family version of the classifier was built based on the exact same architecture as the base version. The only difference is that one additional feature was added to the word vector, containing the content of the four columns that were added to the dataset (Section 3.1.2). A simplified visualisation of this updated word vector is presented in Figure 2. To gain insights into the impact of each word family level, we trained the classifier based on (1) only the *token* level information as extra data, (2) only the *lemma* level data, (3) only the *source* level data, and (4) all three levels combined (*combi*). The word vector for *combi* was obtained by concatenating the “word family feature” from the three individual levels (i.e. the light-coloured part on the right-hand side of the vector visualisation in Figure 2) and appending these values to the “base features” (i.e. the dark-coloured part on the left-hand side of the vector in Figure 2).

4 Results and Analysis

The performance scores are presented in Table 5. We compare the results against a naïve most frequent label (MFL) baseline, which always predicts the most frequent difficulty label in the dataset (i.e. label 1). For evaluation, we first calculate two measures that are insensitive towards changes in class distribution: (1) the D' coefficient, which determines the degree of certainty in the predictions (Smith et al., 2021), and (2) the Matthews correlation coefficient (abbreviated as MCC and denoted as ϕ), which determines the quality of the predictions by estimating the strength of association between the true and predicted classes (Matthews, 1975). Changes in the values of these

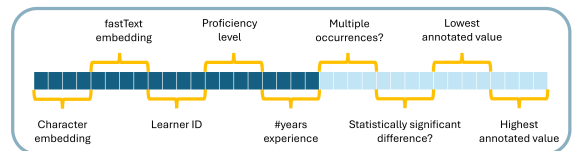


Figure 2: Simplified representation of word vector enriched with word family information. Word family values are different for each of the three possible word family levels (*token*, *lemma*, and *source*). Highest and lowest annotated value are provided per participant.

Classifier type	D' \uparrow	MCC \uparrow	F1 \uparrow	MSE \downarrow	RMSE \downarrow	Accuracy \uparrow
MFL baseline	0	0	0.32	2.61	1.62	0.49
Base	0.18 (\pm 0.01)	0.32 (\pm 0.02)	0.53 (\pm 0.02)	1.32 (\pm 0.1)	1.15 (\pm 0.04)	0.56 (\pm 0.02)
Word family (token)	0.23 (\pm 0.01)	0.37 (\pm 0.02)	0.56 (\pm 0.01)	1.25 (\pm 0.07)	1.12 (\pm 0.03)	0.59 (\pm 0.02)
Word family (lemma)	0.26 (\pm 0.01)	0.4 (\pm 0.02)	0.59 (\pm 0.02)	1.18 (\pm 0.08)	1.09 (\pm 0.04)	0.61 (\pm 0.02)
Word family (source)	0.23 (\pm 0.01)	0.38 (\pm 0.02)	0.57 (\pm 0.02)	1.24 (\pm 0.11)	1.11 (\pm 0.05)	0.59 (\pm 0.02)
Word family (combi)	0.32 (\pm 0.01)	0.45 (\pm 0.02)	0.62 (\pm 0.02)	1.11 (\pm 0.1)	1.05 (\pm 0.05)	0.63 (\pm 0.02)

Table 5: Performance of the different personalised word difficulty classifiers. We report the mean score across the ten cross-validation runs for each of the six performance metrics. Standard deviation values are included between parentheses and the top score per metric is presented in bold.

metrics can be fully attributed to changes in the model, and not to aspects inherent to the data such as class imbalance (Tack, 2021). MCC values can go from -1 (inverse prediction) over 0 (average random prediction) to 1 (perfect prediction), while the D' coefficient ranges between 0 (no discriminative power) and 1 (full discriminative power).

In addition, we also report three commonly used metrics in machine learning: weighted F1 (i.e. the harmonic mean of precision and recall), mean squared error or MSE (i.e. the average squared difference between the true and predicted values), and root mean squared error or RMSE (which converts MSE values to the same units as the dependent variable, in our case the 1-5 LCP scale). Finally, we also include the intuitive accuracy metric (i.e. the number of correct predictions divided by the total number of predictions).

Since, to the best of our knowledge, this is the first study which specifically analyses the potential of including word family information as an input feature, our research provides valuable new insights into the added value of linguistic features in LCP-based classifiers. The results in Table 5 unequivocally show that word family information has a noticeable positive impact on model performance, with the top-performing *combi* classifier achieving a large increase on all metrics in comparison to both the MFL baseline ($+0.32$ for D' ; $+0.45$ for MCC; -0.57 for RMSE) and the base classifier ($+0.14$ for D' ; $+0.13$ for MCC; -0.1 for RMSE).

When breaking down the results per type of classifier, a first finding to be highlighted is that, though leaving ample room for improvement, the **base classifier** already achieves reasonably good performance. The mean D' and MCC values (0.18 and 0.32 , respectively) suggest that the model has ac-

quired weak positive discriminatory and predictive power, while the RMSE score reveals that – on average and including penalisation – the model’s prediction is only 1.15 away from the true label. Importantly, the base model also outperforms the MFL baseline by a large margin (e.g., MCC of 0 compared to 0.32 , weighted F1 of 0.53 compared to 0.32 , and RMSE of 1.15 compared to 1.62).

When comparing the base to the **word family-enriched classifier**, the results clearly show that any type of word family information is helpful for the model, as all subtypes outperform the base version on every metric. The increase in performance is most notable at the lemma level ($+0.08$ for MCC; -0.06 for RMSE), suggesting that the model successfully leveraged information on the knowledge a given learner has acquired for one inflected form of a lemma to predict the label of other inflected forms of that lemma. However, it should be noted that the lemma level is also the level at which most instances in the dataset were enriched (Table 4), which may have played an important role in this particular subtype obtaining the largest increase. Regarding the results for the source subtype, it should be highlighted that – next to the lower number of enriched instances – the 106 statistically significant differences in annotations between the target word and its source (see Section 3.1.2) might be a second reason for the smaller increase at the source level compared to the lemma level.

In summary, the findings of our study provide strong evidence in favour of integrating word family features (and well-chosen linguistic features in general) into personalised word difficulty classifiers. Particularly, with the *combi* classifier obtaining the highest scores, the take-home message is that the more relevant data on word family knowl-

edge are added, the better the classifier’s predictions of word difficulty become.

5 Conclusion

In this paper, we presented a personalised word difficulty classifier for L2 Spanish, trained on the LexComSpaL2 dataset (Degraeuwe and Goethals, 2024). Based on a straightforward BiLSTM architecture with a softmax activation function, the classifier can take any Spanish target sentence as input and will predict a difficulty label ranging from 1 to 5 for every content word in the sentence. Moreover, thanks to the inclusion of learner-specific features in the training process (e.g., proficiency level and years of experience), the model attempts to tailor its output to the unique profile of every learner individually. In doing so, the classifier goes beyond the generic, one-size-fits-all difficulty levels often used in L2 vocabulary learning resources (e.g., based on the Common European Framework of Reference for Languages [CEFR]).

By comparing a base classifier to a “word family-enriched” one, we highlighted the notable added value of feeding information on word families – and of adding linguistic features in general – to word difficulty classifiers. With the top-performing model obtaining an MCC value of 0.45, an F1 score of 0.62, and an RMSE score of 1.05, our classifier shows great potential to be included in real-life ICALL scenarios, for instance as a “difficult word detector” in a personalised reading assistant.

In future studies, we aim to test other machine learning architectures (e.g., using LLMs) and compare them against the BiLSTM classifier presented in this study. Other directions for future research include (1) studying the effect of replacing the static fastText embeddings as an input feature by contextualised word embeddings, (2) analysing the addition of more linguistic features next to word family information, and (3) collecting – in a GDPR-compliant fashion – more information about the participants in order to expand the “learner profile” input feature. Possible additional types of participant information include personal interests (e.g., hobbies), reading behaviour (in L1 or L2), and mastery of other languages than Dutch as L1 and Spanish as L2.

Limitations

A first important observation to be made concerns the real-life applicability of the classifier. Despite

the promising results, it could be argued that the predicted values of the model – even for the best-performing classifier – are not yet close enough to the expected values for the model to be integrated *as is* in real-life settings. As shown by the (R)MSE scores¹¹ of 1.11 and 1.05 for *combi*, there is still a considerable difference between what the classifier should predict and what it actually predicts. Especially in a pedagogical setting it is crucial to obtain relatively high accuracy and precision rates (which is why the F1 and accuracy metrics were included in the analysis), because it should be avoided at all cost that learners lose precious time over errors in their learning materials. For example, if the classifier were used to identify vocabulary items that are known passively but not actively by a given learner (i.e. label 3) and ended up selecting words which are known (very) well by the learner, the exercise would lose most of its pedagogical value.

As obtaining promising but not (yet) pedagogically usable results is a recurrent finding in research on automated word difficulty prediction – Pintard and François (2020), for example, report a top accuracy score of 0.54 for their French CEFR classifier –, one might be left to wonder if the concept of word difficulty (especially for individual learners) is too sophisticated for machine learning classifiers to fully capture. In fact, despite the existence of clear patterns (e.g., high-frequency words tend to be easier than low-frequency words and long words tend to be more difficult than short words), there is also a wide range of factors that may affect perceived word difficulty but that are much harder to model using computational techniques (e.g., a word’s degree of abstractness/concreteness, a learner’s world knowledge, or a learner’s ability to deduce meaning based on morphological knowledge or contextual clues). Yet, using (generative) LLMs as classifiers and/or expanding the number of features related to the learner (see also Section 5) are two research avenues which have the potential of providing the domain of automated word difficulty prediction with a new *élan* and leading to a considerable increase in performance. Additionally, the integration of features indicating how specialised words are for a given domain – for instance using “keyness” (Gabrielatos, 2018) or “termhood” (Rigouts Terryn et al., 2021) metrics – could also be a direction worth pursuing.

¹¹These metrics penalise predictions that are far from the true label more severely than near-correct predictions (e.g., a predicted value of 1 while the true label is 5).

Secondly, in the current setup, every learner who wants to get personalised predictions from the classifier first needs to annotate all 200 sentences in the LexComSpaL2 corpus, as this information is used to build the “learner profile” input feature. As suggested by Degraeuwe and Goethals (2024), to facilitate the implementation in a real-life setting, an item analysis could be performed on the dataset to identify the most “informative sentences” and have new learners annotate this “concentrated” set of sentences instead. Another limitation of the dataset is that, currently, only annotations from L1 Dutch speakers are included. To assess the true personalisation potential of the classifier, the LexComSpaL2 dataset would need to be expanded with annotations coming from L2 Spanish who do not have Dutch as their L1.

Finally, regarding the analyses conducted, the present paper did not provide an in-depth evaluation of the personalisation potential of the classifier. In future research, we aim to isolate this aspect of the model, for example by performing a comparative analysis of the results per learner in order to identify potential differences and look for factors that might explain these differences (e.g., by studying if they correlate with the learners’ proficiency level). Additionally, the study did not address if and how the sentence context impacts the perceived difficulty of a word (perspective of the learner) and how this relates to the predicted difficulty of that word (perspective of the computer). Finally, it should be noted that we did not apply any word sense disambiguation (WSD) method to the data. As a result, homonymic and polysemous words (e.g., *banco* as a bench and as a financial institution) were not considered as two separate tokens or lemmas.

Acknowledgments

The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government – department EWI. Additionally, a sincere word of gratitude goes to Janiça Hackenbuchner (for acting as a soundboard and proofreading), to the people who attended the LATILL workshop at the University of Tübingen (for giving me the inspiration that led to this paper), and to the anonymous reviewers (for their valuable feedback and suggestions).

References

- David Alfter. 2021. *Exploring natural language processing for single-word and multi-word lexical complexity from a second language learner perspective*. Göteborgs Universitet, Göteborg.
- Laurie Bauer and I.S.P. Nation. 1993. **Word Families**. *International Journal of Lexicography*, 6(4):253–279.
- Marsha Bensoussan and Batia Laufer. 1984. **Lexical Guessing in Context in EFL Reading Comprehension**. *Journal of Research in Reading*, 7(1):15–32.
- 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.
- José Cañete. 2019. **Spanish Word Embeddings**.
- Thi Ngoc Yen Dang, Averil Coxhead, and Stuart Webb. 2017. **The Academic Spoken Word List**. *Language Learning*, 67(4):959–997.
- Mark Davies and Kathy Hayward Davies. 2018. *A frequency dictionary of Spanish: Core vocabulary for learners*, 2 edition. Routledge frequency dictionaries. Routledge, London ; New York.
- 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.
- Vanessa De Wilde, Marc Brysbaert, and June Eyckmans. 2022. **Formal versus informal L2 learning: How do individual differences and word-related variables influence french and English L2 vocabulary learning in Dutch-speaking children?** *Studies in Second Language Acquisition*, 44(1):87–111.
- Tamar Degani and Miri Goldberg. 2019. **How Individual Differences Affect Learning of Translation-Ambiguous Vocabulary**. *Language Learning*, 69(3):600–651.
- Jasper Degraeuwe and Patrick Goethals. 2024. **LexComSpaL2: A lexical complexity corpus for Spanish as a foreign language**. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10432–10447, Torino, Italia. ELRA and ICCL.
- 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.

- Zoltán Dörnyei. 2014. *The Psychology of the Language Learner*. Routledge.
- Holly Else. 2023. Abstracts written by ChatGPT fool scientists. *Nature*, 613(7944):423–423.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.
- Natalie Finlayson, Emma Marsden, and Laurence Anthony. 2023. Introducing MultilingProfiler: An adaptable tool for analysing the vocabulary in French, German, and Spanish texts. *System*, 118:103122.
- Costas Gabrielatos. 2018. Keyness analysis: Nature, metrics and techniques. In C. Taylor and A. Marchi, editors, *Corpus Approaches To Discourse: A critical review*, pages 225–258. Routledge, Oxford.
- Robert Godwin-Jones. 2019. In a World of SMART Technology, Why Learn Another Language? *Educational Technology & Society*, 22(2):4–13.
- Patrick Goethals. 2018. Customizing vocabulary learning for advanced learners of Spanish. In *Technological innovation for specialized linguistic domains: languages for digital lives and cultures, proceedings of TISLID'18*, pages 229–240, Gent, Belgium. Éditions Universitaires Européennes.
- 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.
- Bernd Kortmann and Benedikt Szmrecsanyi, editors. 2012. *Linguistic Complexity: Second Language Acquisition, Indigenization, Contact*. Mouton de Gruyter.
- William H. Kruskal and W. Allen Wallis. 1952. Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*, 47(260):583–621.
- Batia Laufer and Geke C. Ravenhorst-Kalovski. 2010. Lexical threshold revisited: Lexical text coverage, learners' vocabulary size and reading comprehension. *Reading in a Foreign Language*, 22(1):15–30.
- Karen Lichtman and Bill VanPatten. 2021. Was Krashen right? Forty years later. *Foreign Language Annals*, 54(2):283–305.
- Katherine I. Martin and Nick C. Ellis. 2012. The roles of phonological short-term memory and working memory in L2 grammar and vocabulary learning. *Studies in Second Language Acquisition*, 34(3):379–413.
- B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.
- 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.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space.
- I.S.P. Nation. 2016. *Making and Using Word Lists for Language Learning and Testing*. John Benjamins Publishing Company, Amsterdam.
- I.S.P. Nation. 2019. The Different Aspects of Vocabulary Knowledge. In Stuart Webb, editor, *The Routledge Handbook of Vocabulary Studies*, pages 15–29. Routledge, London.
- I.S.P. Nation and Stuart Webb. 2011. *Researching and analyzing vocabulary*, 1 edition. Heinle, Cengage Learning, Boston, MA.
- Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical Complexity Prediction: An Overview. *ACM Computing Surveys*, 55(9):1–42.
- Roland R. Nyns. 1989. Is intelligent computer-assisted language learning possible? *System*, 17(1):35–47.
- Jenny A. Ortiz-Zambrano, César 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.
- Gustavo H. Paetzold and Lucia Specia. 2017. A Survey on Lexical Simplification. *Journal of Artificial Intelligence Research*, 60:549–593.
- Ildikó Pilán, Elena Volodina, and Lars Borin. 2016. Candidate sentence selection for language learning exercises: From a comprehensive framework to an empirical evaluation. *Revue Traitement Automatique Des Langues*, 57(3):67–91.
- Alice Pintard and Thomas François. 2020. Combining Expert Knowledge with Frequency Information to Infer CEFR Levels for Words. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with READING Difficulties (READI)*, pages 85–92, Marseille, France. European Language Resources Association.
- Ayla Rigouts Terryn, Véronique Hoste, and Els Lefever. 2021. HAMLET: Hybrid Adaptable Machine Learning approach to Extract Terminology. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 27(2):254–293.
- Simón Ruiz, Patrick Rebuschat, and Detmar Meurers. 2023. Supporting Individualized Practice through Intelligent CALL. In *Practice and Automatization in Second Language Research*, 1 edition, pages 119–143. Routledge, New York.

- Norbert Schmitt. 2010a. [Key Issues in Teaching and Learning Vocabulary](#). In Rubén Chacón-Beltrán, Christian Abello-Contesse, and María Del Mar Torreblanca-López, editors, *Insights into Non-native Vocabulary Teaching and Learning*, pages 28–40. Multilingual Matters.
- Norbert Schmitt. 2010b. *Researching Vocabulary*. Palgrave Macmillan UK, London.
- Norbert Schmitt. 2019. [Understanding vocabulary acquisition, instruction, and assessment: A research agenda](#). *Language Teaching*, 52(02):261–274.
- Norbert Schmitt, Xiangying Jiang, and William Grabe. 2011. [The Percentage of Words Known in a Text and Reading Comprehension](#). *The Modern Language Journal*, 95(1):26–43.
- Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. [CompLex — A New Corpus for Lexical Complexity Prediction from Likert Scale Data](#). In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with READING Difficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.
- Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 Task 1: Lexical Complexity Prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.
- Matthew Shardlow, Kai North, and Marcos Zampieri. 2024. [Multilingual resources for lexical complexity prediction: A review](#). In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 51–59, Torino, Italia. ELRA and ICCL.
- Peter Skehan. 1991. [Individual Differences in Second Language Learning](#). *Studies in Second Language Acquisition*, 13(2):275–298.
- Thomas J. Smith, David A. Walker, and Cornelius M. McKenna. 2021. [A coefficient of discrimination for use with nominal and ordinal regression models](#). *Journal of Applied Statistics*, 48(16):3208–3219.
- Anaïs Tack. 2021. *Mark My Words! On the Automated Prediction of Lexical Difficulty for Foreign Language Readers*. PhD thesis, UCLouvain & KU Leuven, Louvain-la-Neuve, Belgium.
- Stuart Webb. 2021. [The lemma dilemma: How should words be operationalized in research and pedagogy?](#) *Studies in Second Language Acquisition*, 43(5):941–949.
- D. A. Wilkins. 1972. *Linguistics in language teaching*. Edward Arnold, London.
- 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.
- Haomin Zhang and Jiexin Lin. 2021. [Morphological knowledge in second language reading comprehension: examining mediation through vocabulary knowledge and lexical inference](#). *Educational Psychology*, 41(5):563–581.

A Full Visualisation Classifier Architecture

The full visualisation of the architecture of the Bi-LSTM classifier is presented in Figure 3.

B Overview of Observations per Cross-Validation Fold

The overview of the observations per cross-validation fold (overall and per LCP label) is provided in Table 7.

C Class Weights for Cross-Validation

The class weights used by the BiLSTM classifier are presented in Table 6.

Fold	1	2	3	4	5
1	0.41	0.9812	1.2587	2.2369	3.333
2	0.4137	0.9859	1.2493	2.1925	3.2078
3	0.4126	0.9924	1.2446	2.1808	3.2616
4	0.4068	0.9899	1.2553	2.2354	3.4804
5	0.4094	0.9847	1.2505	2.2173	3.4355
6	0.4129	0.9877	1.2444	2.2119	3.2232
7	0.4116	0.9984	1.2529	2.1998	3.161
8	0.411	0.9974	1.2524	2.1979	3.2137
9	0.4059	0.9845	1.2613	2.2673	3.4856
10	0.405	0.9801	1.267	2.2647	3.5718

Table 6: Class weights of BiLSTM word difficulty classifier per cross-validation fold.

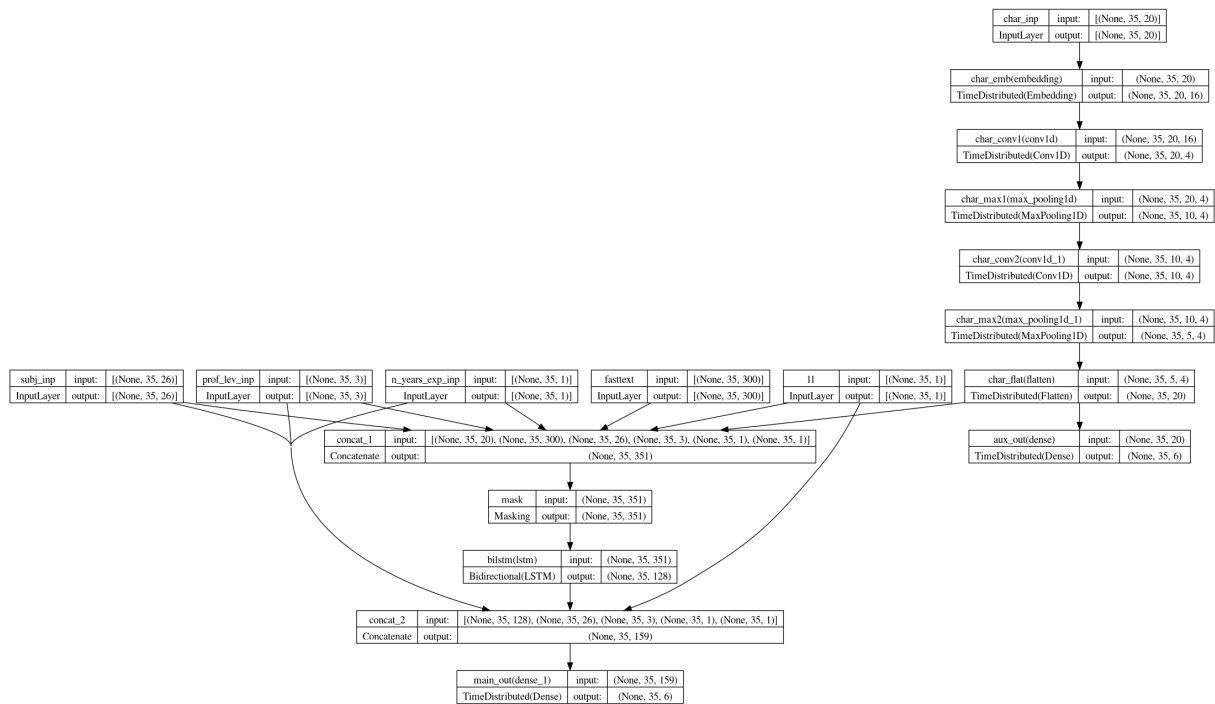


Figure 3: Full representation of BiLSTM word difficulty classifier.

Fold	#observations (target words annotations)			#observations per label				
	TR	VA	TE	1	2	3	4	5
1	1,796 46,696	216 5,616	228 5,928	TR: 22,781 VA: 2,747 TE: 2,889	TR: 9,518 VA: 1,146 TE: 1,123	TR: 7,420 VA: 924 TE: 948	TR: 4,175 VA: 516 TE: 556	TR: 2,802 VA: 283 TE: 412
2	1,797 46,722	227 5,902	216 5,616	TR: 22,589 VA: 3,081 TE: 2,747	TR: 9,478 VA: 1,163 TE: 1,146	TR: 7,480 VA: 888 TE: 924	TR: 4,262 VA: 469 TE: 516	TR: 2,913 VA: 301 TE: 283
3	1,787 46,462	226 5,876	227 5,902	TR: 22,522 VA: 2,814 TE: 3,081	TR: 9,364 VA: 1,260 TE: 1,163	TR: 7,466 VA: 938 TE: 888	TR: 4,261 VA: 517 TE: 469	TR: 2,849 VA: 347 TE: 301
4	1,775 46,150	226 5,876	226 5,876	TR: 22,692 VA: 2,911 TE: 2,814	TR: 9,324 VA: 1,203 TE: 1,260	TR: 7,353 VA: 1,001 TE: 938	TR: 4,129 VA: 601 TE: 517	TR: 2,652 VA: 498 TE: 347
5	1,799 46,774	202 5,252	239 6,214	TR: 22,851 VA: 2,655 TE: 2,911	TR: 9,500 VA: 1,084 TE: 1,203	TR: 7,481 VA: 810 TE: 1,001	TR: 4,219 VA: 427 TE: 601	TR: 2,723 VA: 276 TE: 498
6	1,818 47,268	220 5,720	202 5,252	TR: 22,893 VA: 2,869 TE: 2,655	TR: 9,571 VA: 1,132 TE: 1,084	TR: 7,597 VA: 885 TE: 810	TR: 4,274 VA: 546 TE: 427	TR: 2,933 VA: 288 TE: 276
7	1,803 46,878	217 5,642	220 5,720	TR: 22,776 VA: 2,772 TE: 2,869	TR: 9,391 VA: 1,264 TE: 1,132	TR: 7,483 VA: 924 TE: 885	TR: 4,262 VA: 439 TE: 546	TR: 2,966 VA: 243 TE: 288
8	1,804 46,904	219 5,694	217 5,642	TR: 22,822 VA: 2,823 TE: 2,772	TR: 9,405 VA: 1,118 TE: 1,264	TR: 7,490 VA: 878 TE: 924	TR: 4,268 VA: 540 TE: 439	TR: 2,919 VA: 335 TE: 243
9	1,775 46,150	246 6,396	219 5,694	TR: 22,738 VA: 2,856 TE: 2,823	TR: 9,375 VA: 1,294 TE: 1,118	TR: 7,318 VA: 1,096 TE: 878	TR: 4,071 VA: 636 TE: 540	TR: 2,648 VA: 514 TE: 335
10	1,766 45,916	228 5,928	246 6,396	TR: 22,672 VA: 2,889 TE: 2,856	TR: 9,370 VA: 1,123 TE: 1,294	TR: 7,248 VA: 948 TE: 1,096	TR: 4,055 VA: 556 TE: 636	TR: 2,571 VA: 412 TE: 514

Table 7: Overview of observations per cross-validation fold for training (“TR”), validation (“VA”), and test (“TE”) sets. The training set always contains 160 sentences, the validation and test sets always contain 20. The sets always contain an equal number of sentences per domain (economics, health, law, and migration).

The Need for Truly Graded Lexical Complexity Prediction

David Alfter

Gothenburg Research Infrastructure in Digital Humanities (GRIDH)
University of Gothenburg
Sweden
david.alfter@gu.se

Abstract

Recent trends in NLP have shifted towards modeling lexical complexity as a continuous value, but practical implementations often remain binary. This opinion piece argues for the importance of truly graded lexical complexity prediction, particularly in language learning. We examine the evolution of lexical complexity modeling, highlighting the “data bottleneck” as a key obstacle. Overcoming this challenge can lead to significant benefits, such as enhanced personalization in language learning and improved text simplification. We call for a concerted effort from the research community to create high-quality, graded complexity datasets and to develop methods that fully leverage continuous complexity modeling, while addressing ethical considerations. By fully embracing the continuous nature of lexical complexity, we can develop more effective, inclusive, and personalized language technologies.

1 Introduction

Lexical complexity prediction (LCP) is the task of assigning a complexity score to a word or phrase, indicating how difficult it is to understand for a given target population, such as language learners or readers with disabilities (Shardlow et al., 2022). In recent years, the field of Natural Language Processing (NLP) has witnessed a shift in approach to lexical complexity prediction. There has been a growing recognition that lexical complexity is not a binary concept, but rather exists on a continuum (Shardlow et al., 2020). This acknowledgment has led to efforts to model lexical complexity as a continuous value, promising more nuanced and accurate representations of word difficulty across various contexts and for different readers.

However, despite this conceptual advancement, the practical implementation of truly graded lexical complexity prediction remains limited. This discrepancy between theoretical understanding and

applied research is evident in recent shared tasks and datasets in the field. While the 2021 SemEval shared task on LCP (Shardlow et al., 2021) made strides by including multiple contexts for about half of its training instances, subsequent initiatives have not fully embraced this approach. Notably, the 2024 MLSP (Multilingual Lexical Simplification and Prediction) task (Shardlow et al., 2024) included only a single word with two contexts, effectively reverting to a predominantly one-to-one mapping of words and complexities.¹

This persistent focus on one-to-one complexity mapping not only fails to capture the full spectrum of lexical difficulty but also hinders progress in areas where truly graded predictions are crucial. One such domain is language learning, where learners progress through various levels of proficiency and require finely-tuned assessments of word difficulty (Crossley et al., 2017; Gooding et al., 2021). In this context, binary classifications of “simple” or “complex” are insufficient to guide effective vocabulary acquisition strategies or to develop adaptive learning materials. Further, polysemous words generally show a spread of word senses over different levels, and not all meanings are learned or known at each level (Alfter et al., 2022).

This opinion piece argues that the field of NLP must move beyond its current limited implementation of continuous lexical complexity modeling. We contend that embracing truly graded predictions is not just a matter of theoretical correctness, but a necessity for advancing practical applications in areas such as language learning, text simplification, and readability assessment. By doing so, we can develop more sophisticated and useful tools that accurately reflect the nuanced nature of lexical complexity across diverse contexts and user needs.

Consider the word “crane”. In the context of

¹The test set ($n = 5123$), which may be used as additional training data after the completion of the task, contains 4% of sentences for a word with more than one context.

construction, “crane” refers to a machine used for lifting and moving heavy objects, which might be a familiar concept to most adult readers. However, in the context of ornithology, “crane” refers to a family of large, long-legged birds, which might be less familiar to readers without a background in bird watching or biology. A genuinely graded lexical complexity prediction system should be able to assign different complexity scores to “crane” based on its context, reflecting the varying levels of difficulty for different readers.

A crucial dimension currently underrepresented in lexical complexity research is an explicit theoretical analysis of the construct itself. Lexical complexity is inherently multidimensional, encompassing orthographic difficulty (Just and Carpenter, 1987; Perfetti et al., 2005; Alfter, 2021), conceptual complexity (Nation and Nation, 2001), atypical contextual usage (Erk and Padó, 2008; Peters et al., 2019), and figurative or metaphorical meanings (Steen et al., 2010; Thibodeau and Boroditsky, 2011). These distinct aspects significantly impact different user groups in varied ways; for example, native children encountering conceptual complexity differ from adult second-language learners struggling primarily with orthographic unfamiliarity or contextual atypicality (Akamatsu, 2005; Crossley and McNamara, 2012).

2 Current State of the Field

The field of lexical complexity prediction and simplification has evolved significantly over the past decade, with researchers exploring various approaches to model and predict word difficulty. This section provides an overview of key developments and current trends in the field.

2.1 The Divide Between Two Worlds

In this section, we highlight two related yet disconnected main fields active in lexical complexity prediction: lexical complexity prediction for lexical simplification, and lexical complexity prediction for language learning applications.

Lexical simplification can have a broad range of applications, most aiming at making texts easier to read for certain audiences such as children (De Belder et al., 2010), language learners (Petersen and Ostendorf, 2007; Rets and Rogaten, 2021), people with reading disabilities (Devlin, 1998; Chung et al., 2013), simplifying medical texts (Deléger and Zweigenbaum, 2009) or judicial

texts (LoPucki, 2014), to name but a few. In this line of research, an important first step is to identify *complex* words (Specia et al., 2012). This line of research in lexical complexity prediction started as *complex word identification* (Shardlow, 2013), a binary classification tasks of words into *simple* and *complex* words. Shardlow (2013) presented one of the first comprehensive studies on automatic lexical simplification, focusing on identifying complex words and suggesting simpler alternatives. This binary approach was further developed in subsequent studies, such as Paetzold and Specia (2016b), who introduced a feature-based machine learning approach to complex word identification.

At around the same time, another line of research emerged: graded lexical complexity prediction (Gala et al., 2013, 2014). The main difference to complex word identification is that the aim is to predict a *grade* for each word, corresponding to different school levels for native language learners, and later second language learner proficiency levels (Tack et al., 2016; Alfter et al., 2016; Alfter and Volodina, 2018b; Tack et al., 2018; Pintard and François, 2020). This line of research is tightly connected to (second) language acquisition, with applications such as adaptive learning content (Burstein et al., 2017; Alfter and Graën, 2019) and personalized models for vocabulary learning (Avdiu et al., 2019; Ehara et al., 2018; Yancey and Lepage, 2018).

Over time, the two fields moved closer together, with complex word identification becoming *lexical complexity prediction*, with the aim of predicting a continuous complexity value instead of binary labels. Despite this, it remains that LCP for lexical simplification is concerned with finding words that should be simplified, while LCP for language learning purposes is concerned with finding words that are suitable for learners of a given proficiency level.

2.2 Shared Tasks and Datasets

Shared tasks have played a crucial role in advancing the field. In 2016, the first Shared Task on Complex Word Identification (Paetzold and Specia, 2016a) was organized, followed by the 2018 CWI Shared Task on Complex Word Identification (Yimam et al., 2018). In 2016, the data targeted only English, while in 2018, the task introduced multilingual and cross-lingual complex word identification, but still treating the problem as bi-

nary.² A significant shift occurred with the 2021 SemEval shared task on Lexical Complexity Prediction (Shardlow et al., 2021), which introduced a dataset with continuous complexity scores derived from Likert scale annotations and multiple contexts for many words. This task represented a major step towards more nuanced modeling of lexical complexity.

Despite the progress towards continuous modeling, recent work still shows a tendency to simplify the problem. The 2024 MLSP task (Shardlow et al., 2024), while advancing the multilingual aspect, largely reverted to a one-to-one mapping with limited contextual variation. The training data ($n = 300$) contains a single word with exactly two different contexts and almost identical complexity values. We argue that this is egregiously insufficient to learn different complexities for the same word in different contexts. This setup effectively reduced the task to a one-to-one mapping of words and complexities, disregarding the context-dependent nature of lexical complexity that was captured in the CompLex dataset. In opposition, the 2021 shared task training data ($n = 3487$) contains 1701 words with multiple contexts and different complexity values.

2.3 The Problem

Ideally, one would want to capture context-specific complexity and train systems to automatically predict such complexity. In order to train a system to recognize context-specific complexity, or *truly* graded complexity, the training data would have to include multiple contexts per word with *varying* complexity values. Even though complex word identification moved towards continuous modeling of complexity, it still often only gives one context per word, effectively mapping one word to one complexity value.

Recent research shows that out-of-the-box large language models are not capable of efficiently grading vocabulary (Alfter, 2024; Kelious et al., 2024). This at least to some degree precludes the use of large language models for synthetic data creation. If one were to for example build a system to automatically generate proficiency-adapted definitions, one would need to fine-tune a model with truly graded data (Yuan et al., 2022).

²The task consisted of two subtasks, binary and continuous prediction. However, the continuous labels were obtained by averaging the binary labels over all annotations. We thus regard this task as mainly binary.

3 Data Bottleneck

While theoretical advancements in lexical complexity prediction have pushed towards more nuanced, continuous modeling, a significant obstacle impedes practical implementation: the data bottleneck. This section explores the challenges in obtaining and creating the rich, context-aware datasets necessary for truly graded lexical complexity prediction.

3.1 Data Scarcity

The shift from binary to continuous lexical complexity modeling demands datasets that capture fine-grained distinctions in word difficulty. However, such resources are rarely available at the scale required for robust model training. As noted by Shardlow et al. (2022), creating datasets with continuous complexity ratings is significantly more resource-intensive than binary labeling tasks. Their study found that annotators spent an average of 21.61 seconds per annotation for graded complexity ratings.

The CompLex dataset (Shardlow et al., 2020) represented a step forward by providing continuous complexity scores, but even this resource was limited in size and scope compared to larger binary datasets. CompLex contained 10,800 instances across three genres, which, while substantial, pales in comparison to binary datasets like the one used in the 2018 CWI Shared Task, which contained over 65,000 instances (Yimam et al., 2018).

3.2 Challenges in Dataset Creation

Several factors contribute to the difficulty in conceiving and creating appropriate datasets for graded lexical complexity prediction. One significant challenge lies in the subjective nature of assigning precise, continuous complexity scores to words in context. This task demands skilled annotators yet often leads to low inter-annotator agreement (North et al., 2023), although attempts at mitigating this issue have been made using comparative judgments (Gooding et al., 2019; Alfter et al., 2021, 2022).

Another obstacle is the contextual variation inherent in language. Capturing the full spectrum of contextual variations for each word exponentially increases the annotation effort. The 2024 MLSP task’s inclusion of only one word with two contexts illustrates the practical challenges in scaling contextual annotations.

Furthermore, considerations regarding annotator

characteristics such as linguistic background and language proficiency further complicate dataset creation. Differences between native speakers, teachers, and language learners with varying language proficiency levels can lead to significant variations in perceived lexical complexity, thus limiting the comparability and interpretability of the data. Therefore, a clear definition and control of annotator demographics is essential to ensure the validity and usefulness of complexity-annotated corpora.

In addition, lexical complexity can vary significantly across domains, genres and tasks (e.g., reading aloud, reading for comprehension). Creating datasets that adequately represent this diversity while maintaining consistent annotation quality is a formidable task.

Moreover, complexities introduced by figurative language, including metaphors and metonymies, pose challenges, as such uses often deviate substantially from literal meanings, complicating complexity assessment. Similarly, multi-word expressions (MWEs) introduce unique difficulties because their complexity cannot be straightforwardly derived from the complexity of their constituent words (Alfter and Volodina, 2018a).

Finally, extending graded complexity prediction to multiple languages compounds the resource scarcity. Multilingual datasets like the one used in the 2018 CWI Shared Task are rare and often revert to simpler, binary annotations to maintain feasibility across languages.

3.3 Impact on Model Development and Evaluation

The data bottleneck has cascading effects on the field. Without access to large-scale, graded complexity datasets, researchers often default to simpler binary models or resort to synthetic data generation, potentially limiting model sophistication and real-world applicability. Large-scale extensive annotated datasets allow for more comprehensive coverage of phenomena such as ambiguous words, figurative language use, and multi-part expressions that may be inadequately represented in smaller datasets. Furthermore, larger datasets increase model sensitivity to subtle contextual variations, reduce bias, and improve prediction accuracy in diverse linguistic contexts.

The scarcity of diverse, graded datasets also makes it difficult to comprehensively evaluate models' performance across different contexts, domains, and languages. This can lead to overfit-

ting to specific datasets and poor generalization. Additionally, the relative abundance of binary complexity datasets inadvertently reinforces the continued use of binary approaches, creating a cycle that slows the adoption of truly graded prediction methods.

4 Addressing the Data Bottleneck

To move towards truly graded lexical complexity prediction, it is crucial to develop strategies for creating large-scale, diverse datasets that capture the context-dependent nature of word complexity. In this section, we propose several approaches that could help overcome the data bottleneck.

4.1 Collaborative Annotation Efforts

One approach to creating larger, more diverse datasets is to foster collaborative annotation efforts within the research community. By pooling resources and expertise, one can develop shared annotation guidelines and distribute the workload across multiple institutions. This collaborative approach has been successful in other NLP tasks, such as the creation of the Universal Dependencies treebanks (Nivre et al., 2016). Establishing a similar initiative for lexical complexity annotation could help accelerate the development of high-quality datasets. Further shared tasks on the subject may also help.

4.2 Crowdsourcing and Human Computation

Crowdsourcing platforms, such as Amazon Mechanical Turk, have been used extensively in NLP for data collection and annotation (Snow et al., 2008). By leveraging the power of human computation, one can gather lexical complexity annotations from a diverse pool of participants, potentially covering a wider range of contexts and reader backgrounds. This approach has been successfully used to annotate the CompLex data (Shardlow et al., 2022), to gather *comparative* judgments on lexical difficulty (Alfter et al., 2021, 2022), and to collect age-of-acquisition data (Kuperman et al., 2012; Green et al., 2025). However, quality control mechanisms must be put in place to ensure the reliability of crowdsourced annotations (Sheng et al., 2008).

4.3 Mining Graded Textbook Corpora for Lexical Complexity

Graded textbook corpora, which consist of textbooks designed for language learners at different proficiency levels, offer a promising resource for

creating lexical complexity datasets. These textbooks are carefully crafted to introduce vocabulary and grammatical structures in a gradual, level-appropriate manner, making them a valuable source of information about word complexity in context.

Graded textbook corpora can be leveraged to derive lexical complexity scores by aligning the vocabulary in each level with language proficiency frameworks like CEFR (Council of Europe, 2001). The relative difficulty of words can be determined by analyzing their distribution across proficiency levels. Words frequently appearing in beginner-level textbooks but rarely in advanced ones would receive lower complexity scores compared to those introduced at higher levels. This approach has been explored in the English Vocabulary Profile (Capel, 2010) and the CEFRLex project³, which created CEFR-aligned vocabulary lists from graded textbook corpora.

To extend this approach to lexical complexity prediction, one could leverage techniques from natural language processing, such as word embedding models (Mikolov and Dean, 2013) and contextual language models (Devlin et al., 2018), to capture the semantic and syntactic properties of words in context. By combining these models with the complexity information derived from graded textbook corpora, it may be possible to develop more accurate and context-aware lexical complexity prediction systems.

4.4 Leveraging Large Language Models

Recent advances in large language models offer an attractive avenue for addressing the shortage of richly annotated lexical complexity data. By leveraging large language models (LLMs), researchers can systematically generate diverse contexts for vocabulary items, varying key factors such as linguistic complexity, domain specificity, or target proficiency levels (Alfter, 2024; Kelious et al., 2024). Such synthetic data creation methods could transform even simple, context-free word lists into extensive datasets (Yuan et al., 2022; Green et al., 2025). However, the reliability of LLM-generated complexity annotations would require careful validation, as the generated contexts and associated complexity levels may not align accurately with intended proficiency targets, necessitating subsequent human verification or iterative refinement processes.

³<https://cental.uclouvain.be/cefrlex/>

5 Conclusion

As we have explored throughout this opinion piece, the field of lexical complexity prediction stands at a critical juncture. While recent trends have acknowledged the continuous nature of word difficulty, practical implementations largely remain tethered to binary and one-to-one paradigms. This disconnect between theoretical understanding and applied research impedes progress in areas where truly graded predictions are not just beneficial, but essential.

The persistence of binary and one-to-one mapping methods is not due to a lack of theoretical understanding, but rather stems from a critical data bottleneck. Creating rich, context-aware datasets with continuous complexity ratings is a formidable challenge, requiring significant resources and expertise. This scarcity of nuanced data has cascading effects, limiting model sophistication and evaluation, and inadvertently reinforcing simpler binary paradigms.

Careful consideration of potential user groups is essential to effectively guide the creation and evaluation of lexical complexity datasets. While our discussion primarily focused on second language learners, graded lexical complexity is also suitable for native speakers and various user contexts, such as readability assessments, literacy support, text accessibility, and the development of educational resources. Each user group may require different complexity scales (e.g., continuous numerical scales suitable for NLP systems to discrete scales aligned with educational frameworks such as CEFR proficiency levels or school grades). Future research should explicitly explore these diverse user needs, considering practical implications such as scale granularity and annotation methods to ensure that lexical complexity annotations are both practically relevant and broadly applicable.

In conclusion, the future of lexical complexity prediction lies in not only fully embracing its continuous nature but also in creating resources that reflect various complexity values per word, allowing for the training of *truly* graded lexical complexity prediction systems. By moving beyond binary simplifications and overcoming the data bottleneck, we can develop tools and applications that more accurately reflect the nuanced reality of language complexity.

Limitations

As this is an opinion piece, our focus has been on identifying theoretical limitations and potential avenues for future research within the field of computational lexical complexity modeling. We have not conducted empirical experiments or proposed specific algorithms or datasets. Instead, we have highlighted general shortcomings in existing data and methods and suggested potential directions for advancement.

For the sake of conciseness, we focus on two areas only, namely lexical simplification and language learning. We acknowledge that the implications may reach further than just these two fields.

Ethical Concerns

While the potential benefits are significant, implementing truly graded lexical complexity prediction also presents challenges and ethical considerations. Complexity predictions must account for cultural and linguistic diversity to avoid perpetuating biases. What is considered complex in one cultural or linguistic context may not be in another.

The detailed learner data required for personalized systems raises privacy concerns. Ethical guidelines for data collection and use in educational technology must be carefully considered.

Acknowledgements

This work was financially supported by the Royal Society of Arts and Sciences in Gothenburg (Kungliga Vetenskaps- och Vitterhets-Samhället i Göteborg).

References

- Nobuhiko Akamatsu. 2005. Effects of second language reading proficiency and first language orthography on second language word recognition. *Second language writing systems*, pages 238–259.
- David Alfter. 2021. *Exploring natural language processing for single-word and multi-word lexical complexity from a second language learner perspective*. Ph.D. thesis, University of Gothenburg, Sweden.
- David Alfter. 2024. Out-of-the-box graded vocabulary lists with generative language models: Fact or fiction? In *Proceedings of the 13th Workshop on NLP for Computer Assisted Language Learning*.
- David Alfter, Yuri Bizzoni, Anders Agebjörn, Elena Volodina, and Ildikó Pilán. 2016. From Distributions to Labels: A Lexical Proficiency Analysis using Learner Corpora. In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition at SLTC, Umeå, 16th November 2016*, 130, pages 1–7. Linköping University Electronic Press.
- David Alfter, Rémi Cardon, and Thomas François. 2022. A dictionary-based study of word sense difficulty. 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 17–24.
- David Alfter and Johannes Graën. 2019. Interconnecting lexical resources and word alignment: How do learners get on with particle verbs? In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 321–326.
- David Alfter, Therese Lindström Tiedemann, and Elena Volodina. 2021. Crowdsourcing relative rankings of multi-word expressions: Experts versus non-experts. In *Northern European Journal of Language Technology, Volume 7*.
- David Alfter and Elena Volodina. 2018a. Is the whole greater than the sum of its parts? A corpus-based pilot study of the lexical complexity in multi-word expressions. In *Proceedings of SLTC 2018, Stockholm, October 7-9, 2018*.
- David Alfter and Elena Volodina. 2018b. Towards Single Word Lexical Complexity Prediction. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 79–88.
- Drilon Avdiu, Vanessa Bui, Klára Ptacinová Klimci, et al. 2019. Predicting learner knowledge of individual words using machine learning. In *Proceedings of the 8th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2019), September 30, Turku Finland*, 164, pages 1–9. Linköping University Electronic Press.
- Jill Burstein, Nitin Madnani, John Sabatini, Dan McCaffrey, Kietha Biggers, and Kelsey Dreier. 2017. Generating Language Activities in Real-Time for English Learners using Language Muse. In *Proceedings of the Fourth (2017) ACM Conference on Learning@Scale*, pages 213–215. ACM.
- Annette Capel. 2010. A1–B2 vocabulary: insights and issues arising from the English Profile Wordlists project. *English Profile Journal*, 1(1):1–11.
- Jin-Woo Chung, Hye-Jin Min, Joonyeob Kim, and Jong C Park. 2013. Enhancing readability of web documents by text augmentation for deaf people. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, pages 1–10.
- Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Press Syndicate of the University of Cambridge.

- Scott A Crossley and Danielle S McNamara. 2012. Predicting second language writing proficiency: The roles of cohesion and linguistic sophistication. *Journal of Research in Reading*, 35(2):115–135.
- Scott A Crossley, Stephen Skalicky, Mihai Dascalu, Danielle S McNamara, and Kristopher Kyle. 2017. Predicting text comprehension, processing, and familiarity in adult readers: New approaches to readability formulas. *Discourse Processes*, 54(5-6):340–359.
- Jan De Belder, Koen Deschacht, and Marie-Francine Moens. 2010. **Lexical simplification**. In *Proceedings of ITEC2010 : 1st International Conference on Interdisciplinary Research on Technology, Education and Communication*.
- Louise Deléger and Pierre Zweigenbaum. 2009. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora (BUCC)*, pages 2–10.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Siobhan Devlin. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic databases*.
- Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2018. Mining Words in the Minds of Second Language Learners for Learner-specific Word Difficulty. *Journal of Information Processing*, 26:267–275.
- Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 897–906.
- Núria Gala, Thomas François, Delphine Bernhard, and Cédric Fairon. 2014. Un modèle pour prédire la complexité lexicale et graduer les mots. In *TALN 2014*, pages 91–102.
- Núria Gala, Thomas François, and Cédric Fairon. 2013. Towards a French lexicon with difficulty measures: NLP helping to bridge the gap between traditional dictionaries and specialized lexicons. *E-lexicography in the 21st century: thinking outside the paper*, Tallin, Estonia.
- Sian Gooding, Ekaterina Kochmar, Advait Sarkar, and Alan Blackwell. 2019. Comparative judgments are more consistent than binary classification for labelling word complexity. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 208–214.
- Sian Gooding, Ekaterina Kochmar, Seid Muhie Yimam, and Chris Biemann. 2021. Word complexity is in the eye of the beholder. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4439–4449.
- Clarence Green, Anthony Kong, Marc Brysbaert, and Kathleen Keogh. 2025. Crowdsourced and AI-generated Age of Acquisition (AoA) Norms for Vocabulary in Print: Extending the Kuperman et al.(2012) norms. Preprint.
- Marcel Adam Just and Patricia Ann Carpenter. 1987. *The psychology of reading and language comprehension*. Allyn & Bacon.
- Abdelhak Kelious, 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 NLP for Computer Assisted Language Learning*.
- Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 English words. *Behavior research methods*, 44:978–990.
- Lynn M LoPucki. 2014. System and method for enhancing comprehension and readability of legal text. US Patent 8,794,972.
- Tomas Mikolov and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.
- Ian SP Nation and ISP Nation. 2001. *Learning vocabulary in another language*, volume 10. Cambridge university press Cambridge.
- Joakim Nivre, Marie-Cathrine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of LREC*.
- Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.
- Gustavo Paetzold and Lucia Specia. 2016a. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.
- Gustavo Paetzold and Lucia Specia. 2016b. Sv000gg at semeval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 969–974.
- Charles A Perfetti, Edward W Wlotko, and Lesley A Hart. 2005. Word learning and individual differences in word learning reflected in event-related potentials. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(6):1281.

- Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge Enhanced Contextual Word Representations. 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 43–54.
- Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *Workshop on Speech and Language Technology in Education*.
- Alice Pintard and Thomas François. 2020. Combining Expert Knowledge with Frequency Information to Infer CEFR Levels for Words. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with READING Difficulties (READI)*, pages 85–92.
- Irina Rets and Jekaterina Rogaten. 2021. To simplify or not? Facilitating English L2 users’ comprehension and processing of open educational resources in English using text simplification. *Journal of Computer Assisted Learning*, 37(3):705–717.
- Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st annual meeting of the association for computational linguistics proceedings of the student research workshop*, pages 103–109.
- Matthew Shardlow, Fernando Alva-Manchego, Riza Theresa Batista-Navarro, Stefan Bott, Saul Calderon-Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, and Anna Huelsing. 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.
- Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex: A New Corpus for Lexical Complexity Prediction from Likert Scale Data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with READING Difficulties (READI)*, page 57.
- 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.
- 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.
- Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622.
- Rion Snow, Brendan O’connor, Dan Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263.
- Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics—Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355.
- Gerard J Steen, Aletta G Dorst, Tina Krennmayr, Anna A Kaal, and J Berenike Herrmann. 2010. A method for linguistic metaphor identification.
- Anaïs Tack, Thomas François, Piet Desmet, and Cédric Faron. 2018. NT2Lex: A CEFR-Graded Lexical Resource for Dutch as a Foreign Language Linked to Open Dutch WordNet. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 137–146.
- Anaïs Tack, Thomas François, Anne-Laure Ligozat, and Cédric Faron. 2016. Evaluating Lexical Simplification and Vocabulary Knowledge for Learners of French: Possibilities of Using the FLELex Resource. In *LREC*.
- Paul H Thibodeau and Lera Boroditsky. 2011. Metaphors we think with: The role of metaphor in reasoning. *PloS one*, 6(2):e16782.
- Kevin Yancey and Yves Lepage. 2018. Korean L2 Vocabulary Prediction: Can a Large Annotated Corpus be Used to Train Better Models for Predicting Unknown Words? In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.
- 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, United States. Association for Computational Linguistics.
- Jiaxin Yuan, Cunliang Kong, Chenhui Xie, Liner Yang, and Erhong Yang. 2022. **COMPILING: A Benchmark Dataset for Chinese Complexity Controllable Definition Generation**. In *Proceedings of the 21st Chinese National Conference on Computational Linguistics*, pages 921–931, Nanchang, China. Chinese Information Processing Society of China.

Towards Automatic Formal Feedback on Scientific Documents

Louise Bloch^{1,2,3,*}, Johannes Rückert^{1,*}, Christoph M. Friedrich^{1,2}

¹Department of Computer Science, University of Applied Sciences and Arts Dortmund,
Emil-Figge-Str. 42, 44227 Dortmund, Germany,

²Institute for Medical Informatics, Biometry and Epidemiology (IMIBE),
45147 Essen, Germany,

³Institute for Artificial Intelligence in Medicine (IKIM),
University Hospital Essen, 45147 Essen, Germany,

*These authors contributed equally to the work,

Correspondence: christoph.friedrich@fh-dortmund.de

Abstract

This paper introduces IPPOLIS Write, an open source, web-based tool designed to provide automated feedback on the formal aspects of scientific documents. Aimed at addressing the variability in writing and language skills among scientists and the challenges faced by supervisors in providing consistent feedback on student theses, IPPOLIS Write integrates several open source tools and custom implementations to analyze documents for a range of formal issues, including grammatical errors, consistent introduction of acronyms, comparison of literature entries with several databases, referential integrity of figures and tables, and consistent link access dates.

IPPOLIS Write generates reports with statistical summaries and annotated documents that highlight specific issues and suggest improvements while also providing additional background information where appropriate. To evaluate its effectiveness, a qualitative assessment is conducted using a small but diverse dataset of bachelor's and master's theses sourced from arXiv. Our findings demonstrate the tool's potential to enhance the quality of scientific documents by providing targeted and consistent feedback, thereby aiding both students and professionals in refining their document preparation skills.

1 Introduction

Feedback on scientific documents, for example within a peer-review process, usually and understandably focuses on the discipline-specific content first and foremost. Writing, language, and other formal aspects are a secondary focus and often only commented upon when glaring or repeating issues are present. And while every scientist is expected to have a good grasp of their discipline-specific content, experiences and skills in the areas of writing and language vary greatly among the scientific community, making such feedback less consistent

and more subjective (Shashok, 2008; Wei and Liu, 2024).

For students writing their first scientific documents, such feedback on formal issues is especially useful. These students often make similar mistakes, and supervisors are faced with the task of repeatedly providing feedback about the same issues or focusing feedback on the more important areas, which are usually related to the discipline-specific content and not so much to writing, language, and other formal aspects. This is especially problematic in study courses where writing is not the main way to communicate results or solve tasks.

Existing tools for automated scientific document analysis either focus on analyzing the contents of the documents with regard to their accuracy and veracity, work only for certain document formats, or provide feedback or corrections for specific aspects only. In addition, most of the tools are commercial and closed source. Some of the existing tools are introduced in Section 2.

In this paper, we introduce a web-based open source software¹, which aims to combine a number of existing open source tools and libraries with custom implementations into a single application for analyzing scientific documents under formal aspects pertaining to document structure, readability, literature, referential integrity, tables, and figures. Based on a number of independent document analyzers, it generates reports with statistics and annotated documents with feedback.

2 Related Work

Various tools provide feedback on scientific manuscripts, each with a distinct focus or supported input formats. A detailed review on automated paper review systems (Lin et al., 2023) explained the underlying concepts, recent tools, and challenges.

¹https://gitlab.com/ippolis_wp3/write, Accessed: 2025-06-05

In (Lu and Liu, 2014), a tool was presented which validates the formal compliance of dissertations submitted as DOCX documents with given templates. The tool checks line spacing, font, font size, alignment style, and other formal aspects of DOCX documents. As a pre-processing step, the documents were converted to the eXtensible Markup Language (XML) format. An experiment on 50 dissertations compared automated annotations with manual ones, yielding a 94.5 % detection rate and a 3.7 % false detection rate.

The IEEE PDF eXpress² was developed to validate the consistency of IEEE-related conference and journal submissions in the Portable Document Format (PDF) with respective guidelines. Among other aspects, the proprietary tool checks page margins and the copyright footer.

The ACL pubcheck tool³ performs similar checks for ACL venues. It detects common formatting issues related to the ACL template in PDF documents. These issues include font inconsistencies, improper author formatting, margin violations, and outdated citations.

TeXtidote⁴ is a tool that detects formal issues in LaTeX and Markdown. It checks the style (e.g., proper title formatting, reference capitalization, caption punctuation), citations and references (e.g., consistent citation commands, reference summaries), figures (e.g., presence of captions and references), document structure (e.g., singular subsections, valid section order, stacked headings, and short sections), and hard-coding (e.g., relative paths for figures, hard-coded section/figure/table references, manual line and page breaks). Spelling, grammar, and punctuation errors were detected using the LanguageTool (Naber, 2003), an open source proofreading software based on rule-based correction algorithms and Machine Learning (Brenneis, 2018).

Penelope AI⁵ is a proprietary Artificial Intelligence (AI)-based tool that checks whether DOCX manuscripts meet configurable journal requirements. The tool performs validations, including the availability, position, and title of the ethical approval statement, along with the necessary declarations. It checks the formatting and completeness of

²<https://www.ieee.org/conferences/publishing/pdfexpress.html>, Accessed: 2025-06-05

³<https://github.com/acl-org/aclpubcheck>, Accessed: 2025-06-05

⁴<https://github.com/sylvainhalle/textidote>, Accessed: 2025-06-05

⁵<https://www.penelope.ai/>, Accessed: 2025-06-05

the title page and abstract, as well as the presence of pre-defined sections. Figures and tables are verified for correct integration, proper positioning, logical order, and accurate referencing. The manuscripts are evaluated for accurate referencing styles, proper citation order, and completeness of reference lists. Compliance with journal-specific limits on words, references, tables, and figures is checked. Endnote citations, metadata completeness, page numbers, and line spacing are annotated.

The proprietary YesNoError tool⁶ focuses more on content-related errors than formal feedback. It was designed to process PDF, DOCX, and LaTeX files and validates the methodological process, statistical correctness, and interpretational comprehensibility using OpenAI's o1 model. The analyzers detect issues including mathematical (e.g., arithmetic operations, bracket mismatches), methodological (e.g., study design, sample sizes, statistical tests), literature (e.g., citation and reference consistency), and logical errors (e.g., consistency of statements, conclusions, and argument flow).

Unlike these tools, our tool is open source, supports various file formats, is configurable for different formatting requirements in different disciplines, and emphasizes a broad range of comprehensive formal feedback.

3 Methods and Materials

IPPOLIS Write is a web-based tool that analyzes documents provided by its users based on the configured analysis profile and generates feedback and statistics as annotations directly in the original document and as a report which can be viewed in the web interface. An overview of the general workflow is shown in Figure 1 and is described in more detail in the following sections.

3.1 Cloud Share Link

The documents to be analyzed, analysis progress information and analysis results are stored in a cloud share. Many popular cloud providers are supported through the Web-based Distributed Authoring and Versioning (WebDAV) (Whitehead and Goland, 1999) protocol. Users can utilize their own cloud provider by generating a cloud share link with full read and write permission, allowing the tool to not only read the documents to be analyzed but also save analysis progress information

⁶<https://yesnoerror.com/whitepaper>, Accessed: 2025-06-05

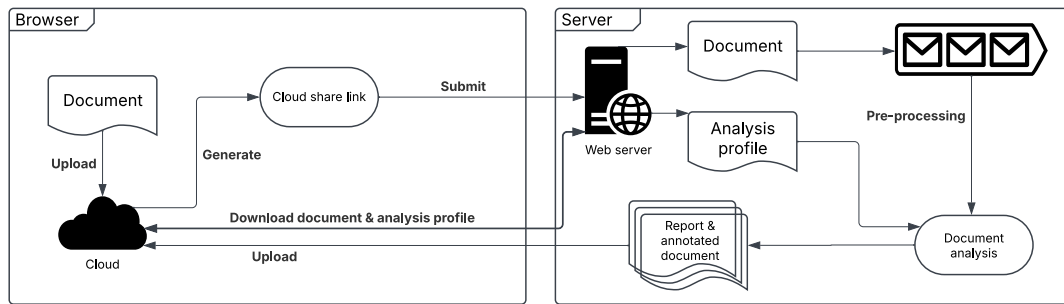


Figure 1: Overview of the document analysis pipeline. Users share a document with the tool through a cloud share link, the document is converted and analyzed on the server, and the annotated document and report are saved in the user’s cloud share and can be viewed on the tool’s website.

and analysis results which can later be displayed on the website. That way, all data is always in the hands of the users. For convenience, a direct upload of documents is planned.

3.2 Analysis Profile

The analysis profile allows configuring, as well as disabling and enabling analysis options. Its goal is to make the tool useful for different disciplines, study courses, formal requirements, and phases in the writing process.

3.3 Pre-processing

After a document in one of the supported formats (PDF, DOCX, LaTeX, BibTeX, RIS, or ZIP) has been submitted, different pre-processing steps are performed to prepare it for analysis. The following sections outline this process for different input formats.

3.3.1 PDF

Although metadata such as the structure can be included in PDFs using PDF tags, this feature is rarely used (Schmitt-Koopmann et al., 2022) making the documents inaccessible to computer systems. IPPOLIS Write combines Poppler⁷ to extract the content, positions, and fonts of texts with document layout detection to determine the functional role of each element. Document layout detection based on You Only Look Once version 8 (YOLOv8) (Jocher et al., 2023) were trained on the DocLayNet (Pfitzmann et al., 2022) and ArxivFormula⁸ datasets. GeneRation Of Bibliographic Data (GROBID) v0.8.1⁹ is used to identify citations and

⁷<https://poppler.freedesktop.org/>, Accessed: 2025-06-05

⁸<https://github.com/microsoft/ArxivFormula>, Accessed: 2025-06-05

⁹<https://github.com/kermitt2/grobid>, Accessed: 2025-06-05

references and to convert bibliographic information into the BibTeX format.

3.3.2 DOCX

Apache Poor Obfuscation Implementation (POI) v5.2.5¹⁰ is used to extract the content, structure, and formatting from DOCX documents.

3.3.3 LaTeX/ZIP

TeXtidote is used for pre-processing LaTeX files, where the text and a mapping to the original LaTeX file is extracted from the document. ZIP archives are supported to allow users to upload LaTeX projects, which usually consist of at least a BibTeX file in addition to one or more LaTeX source files. The archives are extracted and both LaTeX and BibTeX files are analyzed separately.

3.3.4 BibTeX

For literature analysis, the tool focuses on the BibTeX format. The tool uses pybtex v0.24.0¹¹, and jbibtex v1.0.20¹² to extract literature information and identify invalid entries and fields. Bibtool v2.68+ds-1¹³ is used to identify literature entries cited in a LaTeX document, and bibutils v7.2-1¹⁴ converts Research Information System Format (RIS) files to BibTeX.

3.4 Document Analysis

The actual analysis is performed by dozens of independent analyzers based on the configuration in

¹⁰<https://poi.apache.org/>, Accessed: 2025-06-05

¹¹<https://pybtex.org/>, Accessed: 2025-06-05

¹²<https://github.com/jbibtex/jbibtex>, Accessed: 2025-06-05

¹³<https://github.com/ge-ne/bibtool>, Accessed: 2025-06-05

¹⁴<https://ctan.org/pkg/bibutils>, Accessed: 2025-06-05

the analysis profile and the artifacts produced during pre-processing. The analyzers are summarized in nine categories. These are general/formal, document structure, language, readability, literature, reproducibility, referential integrity, images, and tables. The analyzers refer to different input formats and provide analysis results, such as report statistics and annotations. More information on analyzers, their implementations, advantages and disadvantages is described in Section 4. A complete list of implemented and planned analyzers can be found in the Git repository¹⁵. All analyzers can be enabled or disabled for individual requirements via the analysis profile. During analysis, users are informed about the current analysis progress in the web interface. This is entirely asynchronous, and the analysis is not canceled if the user closes the website.

3.5 Report and Annotated Document

The results, statistics and annotations produced by the analyzers are converted into a report and, for some document types, an annotated document which is saved in the user-provided cloud share. For DOCX documents, annotations are embedded as comments, implemented using Apache POI. PDF documents were annotated using iText7 v8.0.3¹⁶. Optional Content Group (OCG) layers are utilized to conveniently enable or disable annotations directly within the PDF document. The report contains statistics and gamification elements, such as comparisons to analysis results of earlier versions of the document.

3.6 Dataset

To present and evaluate the analyzers, experiments were performed on a dataset containing bachelor and master theses in PDF format. The dataset was extracted from the arXiv (Ginsparg, 1994, 2011) preprint server, which was searched for the terms “master thesis” and “bachelor thesis” in the field of computer science (arXiv category: CS.*) on 2025-03-11. This query leads to 492 master theses and 91 bachelor theses. One master thesis cannot be downloaded, and one bachelor thesis is an invalid PDF document. Documents with fewer than 20 pages were excluded, as theses are typically longer, leading to 451 master theses and 79 bachelor theses. It was decided to concentrate on more recent theses,

¹⁵https://gitlab.com/ippolis_wp3/write, Accessed: 2025-06-05

¹⁶www.itextpdf.com, Accessed: 2025-06-05

Dataset	# documents	Avg. # pages
Bachelor train	19	66.63
Bachelor test	19	61.84
Master train	53	73.64
Master test	53	76.83
Σ	144	72.33

Table 1: Distribution of the primary arXiv categories across the training and test set.

which were submitted in arXiv after the 2023-01-01, leading to 110 master’s and 39 bachelor’s theses. One bachelor thesis and four master theses not written in English or German, the languages currently supported by the tool, were excluded. The four master theses were written in Persian, French, partially in Greek, and partially in Japanese. The bachelor thesis was written in Indonesian. The resulting 106 master’s and 38 bachelor’s theses are randomly split into a 50 % training, and a 50 % test set. The most frequently arXiv primary category was cs.LG (Machine Learning; 14.58 %) followed by cs.CL (Computation and Language; 11.81 %). An overview of the dataset is given in Table 1 and a list containing the arXiv IDs as well as a script to download the PDF documents is published in a git repository¹⁷. The dataset includes documents created in LaTeX and Word, featuring a wide variety of templates.

4 Results

The tool was evaluated on the test set. None of the theses were used to develop or optimize the tool prior to evaluation. All documents in the test set were processed without problems during the analysis. In the following sections, an overview of the currently implemented PDF analyzers is given, along with examples from the test set, showing which annotations are generated most frequently, in which areas the tool works well, and for which aspects incorrect annotations are generated most often. In addition, common reasons for formatting violations are explained.

Figure 2 presents boxplots depicting the average number of annotations per page for each annotation category in the test set. The overall number of annotations per page differs between 2.17 and 16.08. Master’s theses exhibit a slightly lower number of

¹⁷https://gitlab.com/ippolis_wp3/bea-2025-ippolis-write-dataset. Accessed: 2025-06-05

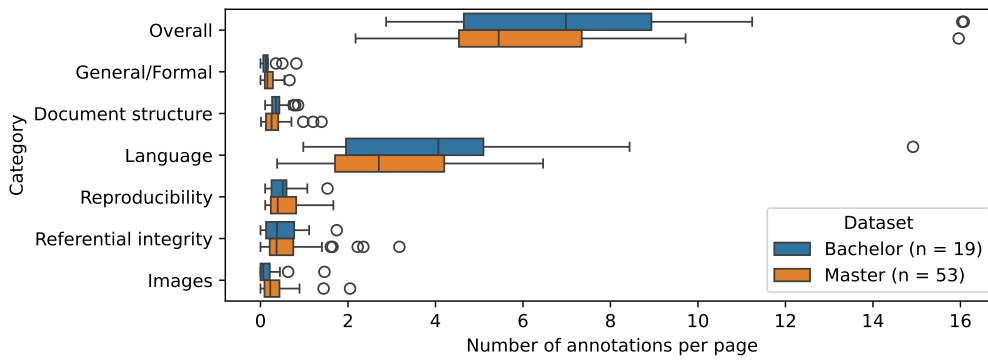


Figure 2: Boxplot showing the average number of annotations per page across annotation categories in the test set.

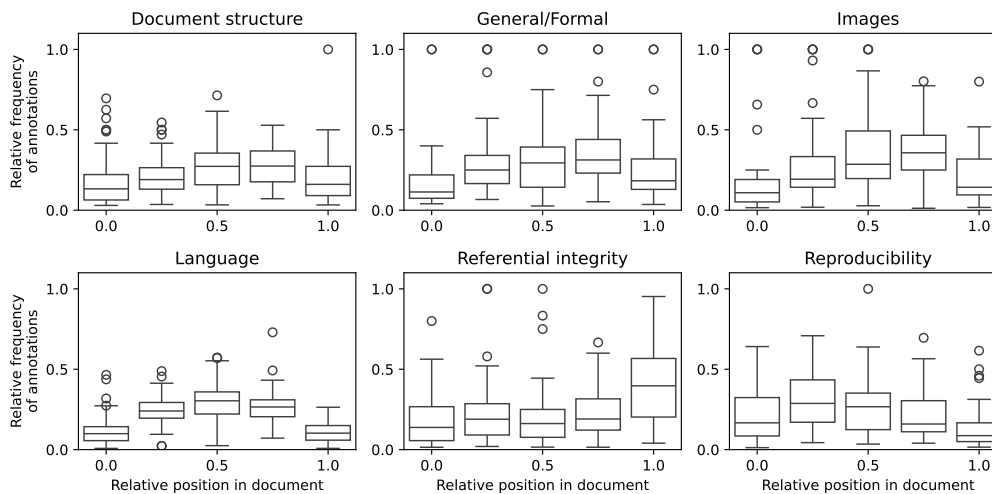


Figure 3: Boxplots showing annotation distribution by category across page positions in the test set.

annotations per page. This indicates that master students have more experience writing scientific theses. The most frequent annotations were language annotations that differed between 0.38 and 14.92 annotations per page. The remaining categories are detected less frequently with a maximum of 3.18 annotations per page, which was reached for a master thesis and the referential integrity category.

Figure 3 illustrates the distribution of annotation frequencies across document pages for each category. The plots show raw annotations that have not been validated. In the document structure, images, and language categories, annotations are more frequent in the middle sections of the documents. One possible explanation is that the middle of the document contains the main body of text along with most of the figures. General/formal and reproducibility annotations exhibit similar patterns, although general/formal annotations tend to accumulate slightly more in the later sections, while re-

producibility annotations are more frequent in the earlier sections. Referential integrity annotations accumulate at the end of the documents, indicating a large number of annotations in the reference sections, and thus references that were not recognized in the text.

A collage of several screenshots showing individual annotations for the master’s thesis (Singh, 2024) is visualized in Figure 4.

4.1 General/Formal

The analyzers that provide general and formal feedback ensure a well-organized presentation, including the identification of changing fonts, line spacing, and text alignments, as well as texts that exceed page margins. In addition, incorrect decimal and thousands separators are recognized, as well as missing punctuation marks at the end of captions.

The test set did not reveal any issues related to font changes. One reason for this might be that the documents are submitted theses, which are often

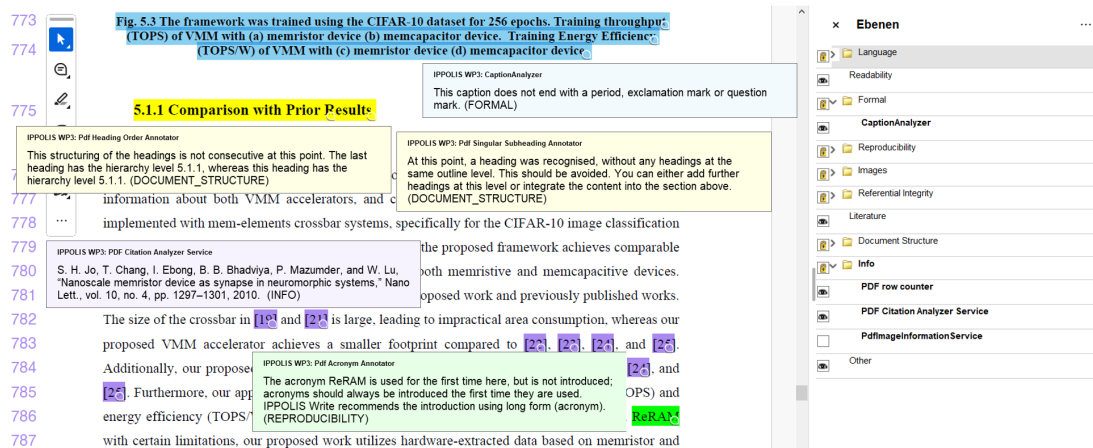


Figure 4: Collage of several screenshots showing individual annotations for the master’s thesis (Singh, 2024) from the test set. The OCG layers are shown on the right-hand side, which can be used to deactivate individual annotators.

revised multiple times. Occasional page margin violations are reliably identified, especially in LaTeX-generated PDFs. These violations often result from incorrect hyphenation settings, improperly formatted equations, or misaligned tables. When such issues are detected, IPPOLIS Write offers feedback, outlines common causes, and suggests ways to prevent them. At the moment, it does not detect figures that exceed page margins, which also occurs in LaTeX-based PDF documents but a future extension is planned for this feature.

In addition, many documents had missing punctuation marks in the captions. For this analyzer, it was observed that the layout detection described in Section 3.3.1 occasionally fails to detect captions correctly, depending on the template used.

4.2 Document Structure

All analyzers in this category are designed to maintain the structural integrity of the manuscript. These include analyses on heading structure (e.g., inconsistent heading numbers and orders, lowercase headings, missing heading layer), page numbers (e.g., missing and inconsistent page numbers), section content (e.g., empty sections), as well as table and figure captions (e.g., availability, length, and consistent positions).

Problems most often and reliably detected in the dataset are empty sections (i.e., a section heading is immediately followed by a subsection heading), and sections without another section at the same level. Due to the difficulty of extracting the document layout from the PDF, captions are sometimes not identified correctly. For unusual formatting (e.g., small caps), section headings are sometimes

not identified correctly. Tables and equations included as images are currently identified and analyzed as images.

4.3 Language

The language-based analyses include the investigation of the spelling, grammar, and punctuation (implemented using the LanguageTool¹⁸) as well as the vocabulary (occurring nouns, verbs, and n-grams). In addition, the detection of filler words and judgmental words (both adapted from the Readability Analysis Tool (Holdorf, 2016) and the angry-reviewer tool¹⁹), hype terms (Millar et al., 2023), and ChatGPT phrases²⁰ are implemented. First-person pronouns are detected as they may diminish the objectivity and neutrality of texts, particularly in specific languages and research fields. Duplicated sentences are annotated to prevent redundancies that could reduce the reader’s attention.

Language problems most often detected in the analyzed theses are grammar errors, frequent use of first-person pronouns, and the use of judgmental words. Most incorrect language annotations are generated because of a lack of context awareness, e.g., inside mathematical expressions, or when a word has several meanings (e.g. “clearly”).

4.4 Readability

The readability analysis focuses on identifying long sentence and reporting several readability scores

¹⁸<https://languagetool.org/>, Accessed: 2025-06-05

¹⁹<https://github.com/anufrievroman/Angry-Reviewer>, Accessed: 2025-06-05

²⁰<https://www.twixify.com/post/most-overused-words-by-chatgpt>, Accessed: 2025-06-05

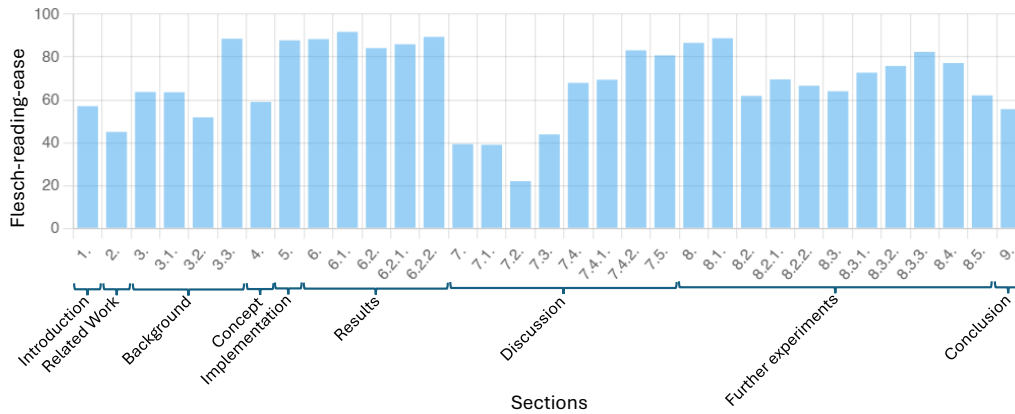


Figure 5: Flesch-reading-ease for document (Lachner, 2021) from the test set.

(e.g., the Coleman-Liau index (Coleman and Liau, 1975), or the Flesch-reading-ease (Flesch, 1948)). The readability scores are implemented to identify changes in language quality across the document. It was anticipated that more theoretical sections, mostly at the beginning of a thesis such as the introduction, related work, or methods sections, would have higher complexity compared to sections focusing on applied content (e.g., implementations or evaluations). A similar pattern was found for the Flesch-reading-ease across one test bachelor thesis (Lachner, 2021) which is visualized in Figure 5. The Flesch-reading-ease considers the average length of sentences and the average number of syllables per word. Higher values indicate easier text comprehension. The example exhibits lower values and thus more difficult text comprehension in the introduction, related work, background, and concept sections, followed by higher values in the implementation and results sections. The discussion section shows mixed scores, while the further experiments and conclusion sections display moderate values. Additional readability scores can be added in future versions of the tool.

4.5 Literature

Literature annotations are generated for citations with incomplete information, incorrect citation types, and published arXiv preprints. Information is retrieved from SemanticScholar (Ammar et al., 2018; Kinney et al., 2023; Lo et al., 2020), DBLP (Ley, 2002), OpenAlex (Priem et al., 2022), and CrossRef (Rachael, 2014).

The extraction of citations and reference lists from PDF documents using GROBID and additional extraction using regular expressions shows acceptable results. However, some citations remain

incorrectly identified, independent of the citation style used. In addition, some false-positive detections occur within equations or references to figures or captions. One reason for the incorrect detection is that GROBID was mainly trained on scientific articles, and the formatting differs from these. The detection accuracy of published arXiv preprints was investigated in more detail in previous work (Bloch et al., 2023).

4.6 Reproducibility

Reproducibility-based analyses include verifying the availability and consistency of hyperlink access dates, detecting URLs that are not properly linked, and identifying acronyms that are used without being introduced.

During the evaluation, it was observed that many documents lack access dates when mentioning URLs. These are often detected reliably, with some exceptions for date formats that omit the day. Missing hyperlinks, i.e. links that are not clickable, are reliably detected.

Almost all documents in the test set contain acronyms that are not introduced properly when first mentioned or employ inconsistent patterns to introduce them (e.g., Long Form (LF) vs. LF (Long Form)). Currently, the tool sometimes mistakenly identifies certain names, such as those mentioned in references (e.g., LeCun), as well as parts of equations (e.g., $A - B$), as acronyms, resulting in incorrect annotations. These issues can be addressed by improving reference detection and recognizing mathematical equations more accurately. The tool currently does not check the availability of acronym indexes, which could serve as an alternative to introducing acronyms upon first use.

4.7 Referential Integrity

The referential integrity of a document is important for the clarity and comprehensibility of scientific texts. IPPOLIS Write ensures updated table of contents, as well as complete and consistent references for tables, figures, sections and references.

In the test set, some PDF documents produced with DOCX and LaTeX suffer from outdated tables of content. For LaTeX documents, the page number of the bibliography section is often incorrect. DOCX documents are affected by inconsistent page numbers and section numbers, often due to manually created or outdated tables of contents. The tool identifies such problems reliably, with some issues when identifying page numbers.

Few figures and tables in the test set have missing captions, which were reliably identified. In contrast, many figures or tables are never or distantly mentioned in the text. In addition, inconsistent reference types were sometimes identified. Similarly to previous observations, occasionally incorrect annotations were produced, resulting mostly from incorrect layout detection, line breaks, or page breaks.

Citations are cross-checked with the literature index based on the previously mentioned GROBID software. The identification and mapping of citations show some issues especially for multiple consecutive references, as well as references in tables. One reason is that the formatting of theses differs from the documents, which were used to train and optimize GROBID.

4.8 Images

The tool examines whether the resolution of figures is sufficient for print publications and ensures they are not distorted during document creation. Furthermore, experiments were conducted to validate the images in more detail. These analyses include the detection of missing axis labels, tick marks, the number of colors, and image artifacts and were implemented using Vision-Language Models (VLMs) (Rückert et al., 2025). These features will be integrated into the tool in the future.

One issue present in almost every analyzed thesis is the inclusion of images with low quality (less than 300 DPI), with some of these including images of clearly poor quality with less than 100 DPI. During the implementation and test of the image analysis features, it became apparent that this is not always due to the quality of the original image,

but often unintentional image compression during document compilation or conversion can lead to poor image quality in the final PDF.

4.9 Tables

At the moment, tables are only checked for valid decimal and thousands separators by the tool. Analyzers regarding the validation of the table structure and the availability of units in table columns are planned. Table captions are analyzed as described in Section 4.2.

5 Discussion

The IPPOLIS Write tool is a web-based open source tool that provides automated feedback on formal aspects of scientific theses and papers to help students, but also researchers fulfill formal aspects of research theses and scientific papers. In comparison to previously developed tools, it is able to process a large number of document formats (PDF, DOCX, LaTeX, BibTeX, RIS, ZIP) and imposes no restriction on used templates. In addition, feedback is provided on a wide variety of formal aspects which are consolidated into nine categories (formal/general, document structure, language, readability, literature, reproducibility, referential integrity, images, and tables). Data privacy is maintained by temporarily sharing documents with the software through a cloud share, while all analysis results are directly stored in the cloud share. This solution enables the implementation of a gamification element, which can motivate users without having to create a user account. The analysis pipeline includes pre-processing of the documents to convert them into machine-readable data and document analyzers, which can be manually configured via the analysis profile. This profile makes it possible to customize the tool to individual requirements, for example, in different departments. The analysis results are converted into a report, and, for some document types, an annotated document was generated. The annotated document makes it easier for the user to quickly understand the annotations and correct the document.

The tool was validated using a PDF dataset from the arXiv preprint server containing bachelor's and master's theses with diverse formatting. The qualitative evaluation investigates which formal issues are identified most frequently, in which areas the tool works well, and for which aspects incorrect annotations are generated. The tool successfully

processes all theses and identified various issues in the test set. These include violations of the page margins, especially in LaTeX-based PDFs, empty sections, and sections without sibling sections at the same hierarchical level. Furthermore, various grammatical errors, missing access dates for URLs, and acronyms with missing or inconsistently introduced long forms were correctly identified. Some documents had outdated tables of content and figures or tables that were never mentioned in the text. Many documents suffered from low image quality, in part with resolutions smaller than 100 DPI. The high frequency of annotations found for theses published on arXiv illustrates the value of the tool.

The most incorrect annotations were identified as being caused by inaccurate layout detection, among other things leading to the detection of non-introduced acronyms or grammatical errors in mathematical equations or citations. This problem will be addressed in future releases by using improved layout analysis. As document layout analysis remains an active area of research (Gemelli et al., 2024), with demonstrated potential to enhance the robustness of Large Language Models (LLMs) for document understanding (Sciurus-Bertrand et al., 2024; Lamott et al., 2024), further advancements in this domain are anticipated in the near future. Additional problems, which will be addressed in future developments, will be improved context awareness during the detection of judgmental words, as well as improved detection accuracy for citations and references using LLMs.

Overall, the results demonstrate that IPPOLIS Write can identify various issues in academic theses, offering valuable formal feedback to assist students and researchers in revising their theses and papers.

6 Conclusion

In summary, this paper introduces IPPOLIS Write, a web-based tool that automatically provides feedback on the formal aspects of scientific theses and papers, assisting both students and researchers in meeting these requirements. IPPOLIS Write covers the most common document formats and a wide variety of formal aspects. In addition, it can be customized to meet the requirements of various disciplines. Data privacy is maintained by temporarily sharing documents through a cloud share. The tool was qualitatively evaluated on a diverse test set containing bachelor and master theses. The validation

shows that IPPOLIS Write detects a wide variety of issues in these documents. Incorrect annotations are mostly caused by inaccurate document layout detection, which is related to the high number of templates used in the dataset.

Limitations

IPPOLIS Write is not a finished product and in some areas lacks in robustness and consistency. Most issues stem from the difficulty of extracting information systematically from documents. Handling subtle differences introduced by diverse document creation tools, layouts, fonts, and formatting has consumed more development time than anticipated, and new documents still sometimes reveal new problems. Better pipelines based on new technologies may alleviate this issue in the future. Another limitation is the lack of a systematic quantitative evaluation of the generated annotations including a manual analysis of false positives and false negatives, as well as a user study, all of which would be useful to provide stronger evidence of the tool's impact. The evaluation dataset is currently limited to student theses from the computer science field. A more varied dataset could help expand and generalize the evaluation results. The tool is meant as a learning resource and does not provide immediate corrections but only suggestions which have to be manually applied, this limitation is an intentional design decision. Limitations that could be addressed in future development iterations of the tool include annotations available directly in the browser (currently only the Adobe PDF viewer fully supports the annotations), marking suggestions as solved/irrelevant for future document analyses, Optical Character Recognition (OCR) pipelines for images, better extraction of bibliography information from PDF, and support for additional languages beyond English and German.

Acknowledgments

This work is part of the BMBF-funded project "Intelligente Unterstützung projekt- und problemorientierter Lehre und Integration in Studienabläufe" (IPPOLIS) (Funding code: 16DHBKI050).

The work of Louise Bloch was partially funded by a PhD grant from University of Applied Sciences and Arts Dortmund, Dortmund, Germany.

References

- Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. [Construction of the literature graph in semantic scholar](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018): Human Language Technologies*, volume 3, pages 84 – 91, New Orleans - Louisiana. Association for Computational Linguistics. Industry Papers.
- Louise Bloch, Johannes Rückert, and Christoph M. Friedrich. 2023. PreprintResolver: Improving citation quality by resolving published versions of ArXiv preprints using literature databases. In *Linking Theory and Practice of Digital Libraries*, pages 47 – 61, Cham. Springer Nature Switzerland.
- Markus Brenneis. 2018. Development of neural network based rules for confusion set disambiguation in LanguageTool. In *SKILL 2018 - Studienkonferenz Informatik*, pages 181 – 192, Bonn. Gesellschaft für Informatik e.V.
- Meri Coleman and Ta Lin Liau. 1975. [A computer readability formula designed for machine scoring](#). *Journal of Applied Psychology*, 60:283 – 284.
- Rudolph Flesch. 1948. [A new readability yardstick](#). *Journal of Applied Psychology*, 32(3):221 – 233.
- Andrea Gemelli, Simone Marinai, Lorenzo Pisaneschi, and Francesco Santoni. 2024. [Datasets and annotations for layout analysis of scientific articles](#). *International Journal on Document Analysis and Recognition (IJ DAR)*, 27(4):683 – 705.
- Paul Ginsparg. 1994. [First steps towards electronic research communication](#). *Computers in Physics*, 8(4):390 – 396.
- Paul Ginsparg. 2011. [ArXiv at 20](#). *Nature*, 476(7359):145 – 147.
- Matthias Holdorf. 2016. Computer support for the analysis and improvement of the readability of IT-related texts. Master’s thesis, Master Thesis at Department of Informatics - Technische Universität München (TUM), Munich, Germany.
- Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023. [Ultralytics YOLO](#). <https://github.com/ultralytics/ultralytics>, Accessed: 2025-06-05.
- Rodney Michael Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Robert Evans, Sergey Feldman, Joseph Gorney, David W. Graham, F.Q. Hu, Regan Huff, Daniel King, Sebastian Kohlmeier, Bailey Kuehl, Michael Langan, Daniel Lin, Haokun Liu, Kyle Lo, Jaron Lochner, Kelsey MacMillan, Tyler Murray, Christopher Newell, Smita Rao, Shaurya Rohatgi, Paul L Sayre, Zejiang Shen, Amanpreet Singh, Luca Soldaini, Shivashankar Subramanian, A. Tanaka, Alex D Wade, Linda M. Wagner, Lucy Lu Wang, Christopher Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Madeleine van Zuylen, and Daniel S. Weld. 2023. [The semantic scholar open data platform](#). Preprint. *Preprint*, arXiv:2301.10140v1.
- Michael Lachner. 2021. [Linear and non-linear machine learning attacks on physical unclonable functions](#). Master’s thesis, Master Thesis at Institute for Computer Science - Ludwig Maximilian University Munich (LMU), Munich, Germany.
- Marcel Lamott, Yves-Noel Weweler, Adrian Ulges, Faisal Shafait, Dirk Krechel, and Darko Obradovic. 2024. Lapdoc: Layout-aware prompting for documents. In *Document Analysis and Recognition - IC-DAR 2024*, pages 142 – 159, Cham. Springer Nature Switzerland.
- Michael Ley. 2002. [The DBLP computer science bibliography: Evolution, research issues, perspectives](#). In *String Processing and Information Retrieval*, pages 1 – 10, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Jialiang Lin, Jiaxin Song, Zhangping Zhou, Yidong Chen, and Xiaodong Shi. 2023. [Automated scholarly paper review: Concepts, technologies, and challenges](#). *Information Fusion*, 98:101830.
- Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. [S2ORC: The semantic scholar open research corpus](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 4969 – 4983.
- Xin Lu and Jing Liu. 2014. [An XML-based model method for review of academic dissertation format](#). In *Proceeding os the seventh International Symposium on Computational Intelligence and Design (IS-CID 2014)*, volume 2, pages 174 – 178.
- Neil Millar, Bojan Batalo, and Brian Budgell. 2023. [Promotional language \(hype\) in abstracts of publications of national institutes of health-funded research, 1985-2020](#). *JAMA Network Open*, 6(12):e2348706 – e2348706.
- Daniel Naber. 2003. A rule-based style and grammar checker. Master’s thesis, Diploma thesis at Technical Faculty, University of Bielefeld, Bielefeld, Germany.
- Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. [DocLayNet: A large human-annotated dataset for document-layout segmentation](#). In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022)*, pages 3743 – 3751.

- Jason Priem, Heather A. Piwowar, and Richard Orr. 2022. [OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts](#). Preprint. *Preprint*, arXiv:2205.01833v2.
- Lammy Rachael. 2014. [CrossRef developments and initiatives: an update on services for the scholarly publishing community from CrossRef](#). *Science Editing*, 1(1):13 – 18.
- Johannes Rückert, Louise Bloch, and Christoph M. Friedrich. 2025. Evaluating compliance with visualization guidelines in diagrams for scientific publications using large vision language models. In *Accepted at the International Conference on Document Analysis and Recognition (ICDAR 2025)*.
- Felix M. Schmitt-Koopmann, Elaine M. Huang, and Alireza Darvishy. 2022. [Accessible PDFs: Applying artificial intelligence for automated remediation of STEM PDFs](#). In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2022)*, ASSETS '22, New York, NY, USA. Association for Computing Machinery.
- Anna Scius-Bertrand, Atefeh Fakhari, Lars Vögtlin, Daniel Ribeiro Cabral, and Andreas Fischer. 2024. Are layout analysis and ocr still useful for document information extraction using foundation models? In *Document Analysis and Recognition - ICDAR 2024*, pages 175 – 191, Cham. Springer Nature Switzerland.
- Karen Shashok. 2008. [Content and communication: How can peer review provide helpful feedback about the writing?](#) *BMC Medical Research Methodology*, 8(1):3.
- Ankur Singh. 2024. [Mem-elements based neuro-morphic hardware for neural network application](#). Preprint, arXiv:2403.03002v1.
- Yuzhu Wei and Donghong Liu. 2024. Incorporating peer feedback in academic writing: a systematic review of benefits and challenges. *Frontiers in Psychology*, 15:1506725.
- E. James Whitehead and Yaron Y. Goland. 1999. [Web-DAV](#). In *Proceedings of the Sixth European Conference on Computer Supported Cooperative Work (ECSCW '99) 12–16 September 1999, Copenhagen, Denmark*, pages 291 – 310, Dordrecht. Springer Netherlands.

Don't Score too Early!

Evaluating Argument Mining Models on Incomplete Essays

Nils-Jonathan Schaller¹, Yuning Ding², Thorben Jansen¹, Andrea Horbach¹

¹Leibniz Institute for Science and Mathematics Education at the University of Kiel, Germany

²FernUniversität in Hagen, Germany
schaller@leibniz-ipn.de

Abstract

Students' argumentative writing benefits from receiving automated feedback, particularly throughout the writing process. Argument Mining (AM) technology shows promise for delivering automated feedback on argumentative structures; however, existing systems are frequently trained on completed essays. Although they provide rich context information, concerns have been raised about their usefulness for offering writing support on incomplete texts during the writing process. This study evaluates the robustness of AM algorithms on artificially fragmented learner texts from two large-scale corpora of secondary school essays: the German DARIUS corpus and the English PERSUADE corpus. Our analysis reveals that token-level sequence-tagging methods, while highly effective on complete essays, suffer significantly when the context is limited or misleading. Conversely, sentence-level classifiers maintain relative stability under such conditions. We show that deliberately training AM models on fragmented input substantially mitigates these context-related weaknesses, enabling AM systems to better support dynamic educational writing scenarios.

1 Introduction

Providing in-process and constructive feedback is integral to fostering argumentative writing skills in educational settings. (Argument Mining) AM has emerged as a promising approach for analyzing and evaluating the structure and quality of argumentative learner essays. However, the majority of AM systems operate on complete texts and their applicability in dynamic writing scenarios with incomplete drafts is unknown. A *conclusion*, for example, might only be recognized as such because it appears at the end of an essay and might not be recognized if the learner starts with the concluding statement and requests feedback early on.

To assess the severity of this problem, we investigate the robustness of existing AM algorithms

when applied to incomplete essay texts. We do so by emulating work-in-progress texts by applying artificial changes and perturbations to two datasets used previously for educational AM: the German DARIUS (Digital Argumentation Instruction for Science) dataset (Schaller et al., 2024b), containing about 4,500 texts on socio-scientific issues, and the English PERSUADE dataset (Crossley et al., 2024), with over 25,000 essays written by US secondary school students on various topics.

These benchmark datasets are probed with two kinds of AM classifiers: a sequence tagger that assigns a label to each token within an essay and a sentence classification approach that labels individual sentences without context. Although sequence tagging exploits contextual information and thus seems suitable for complete essays, sentence classification might have an advantage when only little or misleading context is available. To pave the way for feedback algorithms that can already provide support during the writing process, we explore ways to train a more robust classifier by applying similar perturbations to the training data.

Our paper makes the following contributions:

- We provide benchmark datasets of essays that we corrupted in several more or less realistic ways, which we obtained from the existing PERSUADE and DARIUS datasets to foster the development of robust AM methods.
- We conduct experiments on these datasets to highlight the detrimental effect of incomplete input in educational AM (up to 22 percentage points in the F1 score compared to full texts).
- We train baseline classifiers on similarly corrupted data that can reduce this performance drop to less than half.

All data is made available: <https://github.com/darius-ipn/dontscoretoearly>

2 Related Work

We review related work regarding three aspects. First, we discuss existing datasets for educational AM and select those suitable for our experiments. We then examine machine learning approaches in AM, focusing on the distinction between sequence tagging and sentence classification methods. Finally, we discuss other studies using perturbed essay texts in automated scoring scenarios.

2.1 Educational Argument Mining Datasets

We review datasets according to their suitability for our experiments considering their size, language, and the argumentative units annotated in the data.

Stab and Gurevych (2017) developed an argument detection system for English persuasive essays, achieving substantial inter-annotator agreement (Krippendorff’s α_V of 0.77). Their data consist of 402 English essays annotated for *major claim*, *claim*, and *premise*, as well as their argumentative relationships (*support/attack*).

Wambsganss et al. (2020) built a feedback system for the argumentation structure of German students’ essays. For that purpose, they collected 1,000 peer-reviews from a business innovation course in which students evaluated each other’s business models. The corpus was annotated by three native German-speakers for argumentative components (*claim* and *premise*) and their relationships.

The PERSUADE dataset of Crossley et al. (2022) consists of over 25,000 argumentative essays written by secondary school students in the US. Each essay was annotated for seven distinct argumentative components. The dataset was expanded (Crossley et al., 2024) with effectiveness scores per unit and holistic scores for the overall essay quality. In our previous work (Schaller et al., 2024b) we compiled the DARIUS corpus of 4,589 argumentative essays written by secondary school students in Germany. The corpus consists of two writing prompts on socio-scientific topics. The corpus features detailed annotations of argumentative elements, including *content zone*, *major claim*, *position*, and *warrant*.

Stahl et al. (2024) presented a German corpus of 1,320 school student essays annotated for argumentative structure and quality. Their four-level annotation scheme achieved high agreement ($\alpha = 0.74$ - 0.89). Their analysis revealed significant correlations between structural elements and essay

quality. The corpus provides another reference point alongside DARIUS for student writing in German secondary schools.

Velentzas et al. (2024) presented KUPA-KEYS, a dataset of keystroke logs from 1,006 participants’ English essays, including both L1 and L2 writers. Each essay was evaluated on the CEFR scale by three human assessors and an automated system. The dataset captures detailed keystroke patterns, pauses, and revisions during writing tasks, with the analysis showing moderate correlations between keystroke patterns and writing proficiency.

While keystroke logging would be ideal for studying incomplete texts as it captures the authentic writing process, existing keystroke datasets such as KUPA-KEYS lack the specific argumentative annotations needed for our work.

We decided to use the DARIUS and PERSUADE datasets for our experiments because they offer several key advantages: First, they cover different languages (German and English), allowing us to verify whether our findings hold across languages. Second, they are very similar in their annotation schemes, allowing us to focus on sequences instead of whole documents. Third, with over 4,500 and 25,000 essays respectively, they provide sufficient data to train robust machine learning models and conduct comprehensive robustness evaluations

2.2 Machine Learning in Argumentation Mining

AM consists of two subtasks: the detection of argument units and their classification as a certain type of argument, e.g., a claim or a conclusion.

Approaches to Argument Unit Detection There are two main strategies for detecting argumentative units, although variations are possible: Some use sentence classification, treating entire sentences as argumentative units. Alternatively, sequence tagging works at the token level to identify more flexible argument boundaries. We further review both approaches and hybrid methods.

Sentence Classification Approaches An easy (but not necessarily optimal) method for the selection of units is sentence classification, thus omitting explicit argument detection and classifying individual sentences in a text as belonging to a certain type of argument (or as being non-argumentative). Wambsganss et al. (2020) built a feedback system for the argumentation structure of students’ Ger-

man texts based on a sentence-level multiclass classification task. They developed an SVM-based system for claim/premise identification (65.4% accuracy) and relationship classification (72.1% accuracy). Their later ArgueTutor system (Wambgsanß et al., 2021) improved performance using BERT (F1 = .73). Similarly, in their fairness investigation, Schaller et al. (2024a) employed sentence classification approaches among others, comparing a supervised SVM, a BERT-based classifier, and zero-shot GPT-4 on the DARIUS corpus of students’ German essays.

Sequence Tagging Approaches At the other end of the spectrum, many researchers have employed sequence tagging on tokens to allow for more flexible argument boundaries. This approach has gained significant attention in the field, particularly with the rise of transformer-based models. In our fairness study, (Schaller et al., 2024a) we additionally employed sequence tagging approaches, finding that a task-specific fine-tuned BERT model consistently outperformed other approaches, including more powerful decoder-based language models such as GPT-4 when used in a zero-shot setting.

Stahl et al. (2024) trained sequence labeling models based on mDeBERTaV3-adaptor, which achieved a F1 scores up to .68 for discourse functions.

We previously investigated AM using the English PERSUADE and MEWS and the German DARIUS datasets (Ding et al., 2024). Our sequence tagging used a Longformer-based model, achieving an F1 score of .66 on English essays and providing important baseline performances on complete essays. The analysis revealed that educational context differences impacted performance more than language differences did.

Comparative and Hybrid Approaches Several researchers have compared these approaches or examined the advantages of hybrid approaches.

Trautmann et al. (2020) explicitly examined sentence versus token classification for argument recognition on annotated Common Crawl data (IAA α_{unom} = .61). Their experiments with various BERT and FLAIR models showed that a BERT_LARGE sentence classifier was only outperformed by the BERT_LARGE token classifier when combined with a CRF model, demonstrating that sentence classification can achieve comparable results to token classification approaches.

Stab and Gurevych (2017) showed the advantages of combining both sequence labeling and classification for AM. They first used a CRF for sequence labeling to identify argument boundaries in the text and then used SVM classification to determine each argument’s type and relationships. This combined approach significantly outperformed using either method alone, achieving an F1 score of .86 compared to .79 for individual classification and .64 for baseline approaches. These studies highlight that the optimal approach for complete texts may depend on specific tasks, domains, and available data, and that one approach may not always be superior.

2.3 Influence of Rearranged Sequences

As our work investigates the robustness of AM on incomplete texts, we review previous work on educational scoring that examined model behavior with incomplete texts or nonstandard text order.

Farag et al. (2018) tested AES robustness against shuffled sentences. Their LSTM model performed well on regular essays but declined with shuffled texts. They addressed this by combining models trained on both regular and permuted essays, maintaining scoring performance while detecting shuffled texts. This highlights neural models’ reliance on expected sequence orders, informing our work on incomplete texts.

In a previous paper (Ding et al., 2023), we used sequence tagging with fine-tuned RoBERTa to identify EFL email segments. When tested with scrambled segments, performance dropped strongly (F1 = .89 to F1 = .60), but the model was able to adapt when trained on scrambled data (F1 = .85). On the basis of this work, we anticipate that sentence classification models will outperform sequence tagging models due to lower dependence on position and context.

3 Data

This section presents the two datasets used in our evaluations, DARIUS and PERSUADE, and discusses our decision regarding which annotations to include. Table 1 gives an overview of the key statistics of the datasets.

DARIUS The DARIUS corpus (Schaller et al., 2024b) contains 4,589 argumentative texts by 1,839 secondary school students in 33 German schools. The task consists of two writing prompts on socio-scientific topics: energy and automotive (gener-

	DARIUS	PERSUADE
Language	German	English
Essay genre	argumentative	argumentative
Writing prompts	2	15
Grade	9-13	6-12
# Essays	4,675	25,996
avg. Word count	150	399

Table 1: Key statistics for DARIUS and PERSUADE

ating 2,307 and 2,282 essays, respectively), with students completing both a draft and a revision for one prompt and a single essay for the other. The corpus features detailed annotations of argumentative elements, including *content zone*, *major claim*, *position*, and of argumentative units. The average essay has 150 words.

For our experiments, we focused on *content zone* annotations. *Content zone* describes the macro structure of an essay: *introduction*, *main part*, and *conclusion* (see Fig. 1). Not all essays contain all three sequences, as students may have returned unfinished texts or skipped introductions, etc. Each sequence consists of one or more complete sentences and must end with a sentence-final punctuation mark. In contrast to the other annotations in DARIUS, their sequences are not based on sentences but on spans of multiple sentences, similar to the annotations of PERSUADE.

PERSUADE The PERSUADE dataset (Crossley et al., 2022) consists of over 25,000 argumentative essays written by secondary school students from grades six through twelve in the US. The essays are based on 15 prompts, eight independent and seven source-based. Each essay was annotated for seven distinct argumentative components: (see Fig. 2 and Fig. A.6 in Appendix A) *lead*, *position*, *claim*, *counterclaim*, *rebuttal*, *evidence*, and *concluding statement*. The inter-rater agreement achieved an F1 score of .73. The dataset was expanded (Crossley et al., 2024) to include effectiveness scores for individual discourse elements and holistic essay quality scores. The average essay has 399 words. Sequences can span any length from a single phrase to multiple sentences and do not always align with sentence boundaries.

4 Benchmark Datasets

In order to gauge classifier performance on incomplete essay data, we simulate essays in the process

CO₂ emissions and greenhouse gas emissions are to be drastically reduced as they have had a major impact on our climate change in recent years. But how can we do this work as efficiently as possible to meet the energy demand in this district? There are three options to choose from. The construction of a wind energy park, a solar plant, or a hydroelectric power plant. Which of these three projects should be supported?

If one considers the first criterion, i.e., efficiency, the hydropower plant is clearly in the lead with 70–90%. The solar park is the weakest here. The second criterion, annual yield, is close because both the wind farm and the hydropower plant have good annual yields. And so it goes on with the other criteria so that one can say that the construction of a hydropower plant is the most efficient in relation to the construction of an energy converter. The only major drawback is the price of the project, which brings us back to the wind farm because the price difference alone is a whopping 55 million euros. And the criteria are not so bad that you can't compensate for the deficits over the years.

In conclusion, it can be said that due to the enormous price, a wind farm is the best way to produce electricity for a region that is not so large.

Figure 1: Example essay from the DARIUS dataset (translation by the authors). From top to bottom: *introduction*, *main part*, *conclusion*.

Driverless cars have been a big topic lately. In some ways driverless cars sound cool but they also seem a little scary.

I think that driverless cars shouldn't be allowed on public roads because they are not safe.

Some think being able to have your car drive itself sounds nice. You could just sit in your car and listen to music while you wait to arrive to your destination. Driverless cars would allow you to sit in your seat, hands on the wheel, but not actually driving.

This idea does sound nice but as all other technology such as computers and phones, technology is not always reliable.

A driverless car could cause a major or even fatal crash.

While most driverless cars require you to have hands on the wheel this does not mean you will be paying attention if something is about to happen. All it would take is for something in the car to mess up and people could be very seriously hurt.

I think that people driverless cars are not safe and they should not be allowed on public roads.

Figure 2: Essay 65 from the PERSUADE corpus. From top to bottom: *Lead*, *Position*, *Claim*, *Counterclaim*, *Rebuttal*, *Evidence*, *Concluding Statement*.

of being written by making different changes and perturbations to the original essay data.

Across all benchmark variants, we retain the original gold-standard labels from the uncorrupted texts, i.e., we assume that the label of a unit does not change depending on the context or lack thereof. In doing so, we want to simulate a process where students write what they intend to write but not

necessarily in the right order, and the justification for a label is that the unit has a certain function in the complete text even if that is not yet obvious to a teacher or annotator in the partial essay during the writing process. While this may lead to cases where labels are not recoverable from the modified input alone (e.g., conclusions appearing mid-essay in Shuffled texts), our goal is to test model robustness in incomplete or misleading contexts, reflecting educational scenarios where students submit partial drafts.

We include the following variants:

Full text: The original essays are used as a baseline.

First_X: This simulates an incremental, top-down writing process where a learner receives feedback after having completed at least 25%, 50%, or 75% of the whole text. We clip DARIUS essays after 25% of sentences and PERSUADE essays after 25% of sequences, as DARIUS annotations span at least one sentence, while PERSUADE annotations can be phrase-level annotations.

Last_X: Similar to the former scenario, this simulates the (admittedly less likely) process where only the last 25%, 50%, or 75% of a text is scored.

Sentence: Every single sentence from the test data as its own document, i.e., sentences are labeled completely without context.

Shuffled: For this condition, sequences within an essay are randomly shuffled to appear in misleading contexts, with DARIUS shuffling at the sentence and PERSUADE at the sequence level.

Table A.7 in Appendix A provides examples for all variants of an essay. Note that we do not expect all of these variants to occur (frequently) in real-life data. We rather aimed to cover the whole spectrum to also assess worst-case scenarios. We kept the datasets comparable whenever possible, i.e., **full text**, **sentence**, and **shuffle** contain exactly the same material, just in a different order or different document size, while **first_X** and **last_X**, for obvious reasons, contain less material. Similarly, also the label distribution changes for these benchmarks, e.g., essays with the last part missing, tend to contain less conclusion material. Table A.5 in Appendix A shows an overview of the dataset sizes per benchmark.

5 Experiments

This section evaluates how AM approaches perform on incomplete texts that simulate intermediate products within the writing process.

We compare sequence tagging and sentence classification. The models are tested on both complete essays and simulated incomplete drafts to assess their ability to provide feedback on unfinished texts.

5.1 Experiment 1: Baseline Performance on Benchmark Datasets

Data splits The DARIUS corpus consists of 4,581 essays, which we divided into 3,672 essays for training and 909 essays for testing, following the setup from Schaller et al. (2024a). From the training set, we reserved 20% as a development set to determine the optimal number of training epochs, while using the remaining 80% for model training.

The training set of the PERSUADE (1.0) corpus contains 15,594 essays, which were released in a Kaggle competition¹. We use them all for training. For testing, we use 1,560 random essays from the PERSUADE (2.0) test set².

Classifiers For the comparison of sequence tagging and sentence classification, we use transformer models with different classification heads while keeping the base architecture consistent within each language. For DARIUS, we use a BERT (Devlin et al., 2019) model, bert-base-german-cased, both for the sequence tagging and the sentence classification model. It has a sequence length of 512 tokens, which is adequate for the average DARIUS essay. Longer texts were truncated to 512 tokens. The model was fine-tuned using the default hyperparameters from the Hugging Face Transformers library. We trained for four epochs, which proved sufficient for convergence on our validation set.

As almost one-third of the English essays in PERSUADE contain more than 512 tokens, we use a pretrained Longformer model (Beltagy et al., 2020) for token classification, with a maximal training length of 1,024 tokens, to train a sequence tagging pipeline (Ding et al., 2022) for the prediction of different argumentative elements in the PERSUADE essays. For sentence classification, we use

¹<https://www.kaggle.com/competitions/feedback-prize-2021>

²https://github.com/scrosseye/persuade_corpus_2.0

the pretrained BERT model (Devlin et al., 2019) for sequence classification.

Evaluation Measures We use F1 scores to evaluate and compare sequence tagging and sentence classification approaches across different benchmark datasets. For both datasets, we treat each argumentative element type (e.g., *introduction*, *main*, and *conclusion* in DARIUS; *lead*, *position*, *claim*, etc. in PERSUADE) as a separate class in a multi-class classification setting.

For sequence tagging, we compute F1 scores directly on the token-level predictions, where each token receives one label. When comparing this to sentence classification approaches, we also derive sentence-level metrics from our token classifiers through majority voting (assigning the most frequent token label as the sentence label).

5.2 Experimental Results

Results for DARIUS Table 2 shows the performance of the token and sentence classifier for the various benchmark datasets.

On complete essays, the token classifier demonstrated high performance, with an overall F1 score of .93 on the sentence test set and .92 on the token test set, showing particular strength in identifying the *introduction* (.91/.88) and *main* sections (.96/.95), although it performs weakly on the *conclusion* (.73/.71). This will be the baseline for the other benchmark datasets. As the results for both datasets are very similar, we will further discuss only the sentence test set.

With regard to the decontextualized datasets, it can be seen that the performance decreases. When tested on the first 25%/50%/75% of sentences, relatively stable F1 scores are observed: *overall* (.93-.96), *introduction* (.91-.92), and *main* (.94-.97). The drastic drop in *conclusion* performance (.0-.56) stems directly from the near-complete absence of conclusion sentences in these portions. In the first 50% of essays, there are virtually no conclusion sentences (30 instances compared to 13,679 for the full texts), explaining the .0 F1 score. Most conclusion sentences appear only in the last 25% of essays, as confirmed by the higher support numbers in the Last_25 and Last_50 datasets.

Testing on the **Last_75** reveals the reversed situation, reflecting the lack of *introductions* while maintaining the same number of *conclusions* as in the full essays. The F1 scores for the *overall*, *main*, and *conclusion* remained stable, although

the *introduction* F1 score decreased substantially to .76.

The analysis of the **Last_25** demonstrated a substantial drop in the *overall* F1 score (.72) and the *conclusion* (.32) - a seemingly unexpected result given that all *conclusion* samples remain present in this set. This decline occurs because conclusion sections now appear at the beginning of these truncated texts. The model struggles to recognize conclusions when they are artificially repositioned, indicating that it relies on positional context.

An even worse picture emerges from the analysis of the **Sentence** and **Shuffled** benchmark data. Both include all samples of each annotation but an extreme drop can be seen in the *conclusion*. The token classifier, in particular, is not able to predict the *conclusion*, if given only a sentence example of it (.0). The *introduction* also drops to an F1 score of .70. In the shuffled condition, the *introduction* performance declines further to .31.

These results suggest that, although the model learns the typical structure of complete essays, it struggles to apply this knowledge to incomplete texts.

To further investigate this assumption, we also trained a model for sentence classification, see Table 2. Compared to the token classifier, this model has a slightly lower overall F1 score of .91, .86 for the *introduction* and a slightly higher F1 score of .96 for *main* when tested on the **Full text**. It also performs lower on the *conclusion*, with a score of .60. But compared to the performance of the token classifier on sentences (.0), a much better performance was observed here, especially on the *conclusion*. This might indicate that the token classifier heavily relies on the context of each class, whereas the sentence classifier inherently learns the structure of each class without further context.

Results for PERSUADE Table 3 shows the performance of both token and sentence classifiers for PERSUADE. Similar to DARIUS, we observe substantial performance variations across the benchmark datasets. On complete essays, the token classifier achieves an overall F1 score of .57, with stronger performance on *lead* (.78) and *concluding statement* (.77), probably due to their fixed positions.

For partial texts, the **First_X** benchmarks show a moderate decline in the overall performance (.52-.46). *Lead* detection decreases (.67 to .57), while *evidence* detection improves (.44 to .78), suggest-

ing that positional cues become less reliable, while content-based identification improves with more context. *Concluding statement* detection remains near zero except for the **First_75**, with a score of .14. This is, similar to DARIUS, due to the lack of concluding statements in the first part of the texts. The **Last_X** benchmarks reveal significant performance degradation. Overall, the F1 scores (.11-.25) are much lower than for the **Full text** or **First_X**. Similar to the findings for DARIUS, the *conclusion* shows a sharp decrease (.77 to .39), suggesting that the positional context is also critical for PER-SUADE. The weakest performance appears in the lead (.21) and claim (.00) detection in the **Last_25**, where claims are absent or near-absent.

In the decontextualized **Sentence** and **Shuffled** conditions, the token classifier performs poorly (.20 and .30 overall), while the sentence classifier achieves better results (.43 overall), particularly for *position* (.51) and *evidence* (.48).

These results confirm our DARIUS findings: AM models trained on complete essays have difficulty with partial or nonsequential inputs. The more complex argumentation schema in PER-SUADE (seven classes versus three in DARIUS) appears to enhance this issue.

5.3 Experiment 2: Training on Incomplete Texts

Our previous experiment demonstrated that AM models trained on complete texts showed a decrease in performance when confronted with incomplete or out-of-context texts in the case of DARIUS, especially for the *conclusion*. To further investigate this issue, we explored whether training on deliberately split texts could enhance the models' robustness on the benchmark datasets.

Experimental Setup We developed two additional training strategies for the DARIUS dataset. Both used the same amount of training data but differently divided:

The **Split** model was trained on randomly split versions of the training texts, similar to the **First_X/Last_X** benchmark. Each essay was divided into complementary portions (25%/75%, 50%/50%, or 75%/25%).

The **Hybrid** model was trained on a combined dataset consisting of both complete essays (as in our original token classifier) and their split versions (as in the **Split** model).

Both models used the same BERT architecture

as our original token classifier and only differed in the training data composition. We then evaluated these models on the same benchmark test sets as those used in our previous experiments.

Results and Analysis Table 4 presents the performance differences between our new models (**Split** and **Hybrid**) and the original token classifier across all test conditions. The values represent changes in F1 scores relative to the original model. Several key patterns emerge from these results:

1. **Performance on complete essays:** Both the **Split** and the **Hybrid** models showed slight performance decreases (F1 scores of -.04 and -.02 overall, respectively) when tested on complete essays, with the most substantial drop observed for conclusion detection (-.17 and -.10). This suggests that, when testing on complete essays, models benefit from training on well-formed, complete training essays, as these contain valuable signals for the task.

2. **First_X:** For essays containing only beginning portions, both models performed similarly to the original classifier, with minor decreases in performance (F1 scores of -.01 to -.03 overall). This indicates that detecting introductions and main content remains relatively robust across training strategies.

3. **Last_X:** The most substantial improvements appeared in the **Last_25** condition, where the **Split** model achieved a +.12 increase in the overall F1 score, with an enormous +.40 improvement in conclusion detection. The **Hybrid** model showed more moderate but still positive gains (+.06 overall, +.22 for conclusions). This pattern of improvement continued in the **Last_50** condition but diminished in the **Last_75** as texts become more complete.

4. **Decontextualized conditions:** For completely decontextualized sentences, both new models substantially improved conclusion detection performance (+.49 for **Split**, +.39 for **Hybrid**), while maintaining or slightly improving overall performance. For shuffled texts, the improvements were small but still positive for conclusion detection.

5.4 Discussion

Our findings indicate trade-offs in training approaches for AM in educational settings. The original token classifier performs well on complete essays but has limitations when identifying argumentative elements, particularly the *conclusion*, when these appear in unexpected positions or without sufficient context.

Variant	Token Classifier on Token Testset				Token Classifier on Sentence Testset			
	F1 overall	F1 Intro	F1 Main	F1 Conc	F1 overall	F1 Intro	F1 Main	F1 Conc
Full text	.92	.88	.95	.71	.93	.91	.96	.73
Sentence	.82	.70	.93	.00	.82	.71	.93	.00
Shuffled	.75	.31	.88	.18	.74	.31	.87	.17
First_25	.92	.90	.94	.00	.93	.92	.94	.00
First_50	.95	.90	.97	.04	.96	.92	.97	.09
First_75	.95	.88	.97	.54	.96	.91	.97	.56
Last_25	.70	.05	.86	.30	.72	.07	.87	.32
Last_50	.86	.25	.93	.57	.88	.30	.94	.60
Last_75	.92	.70	.95	.68	.92	.76	.96	.70
Sentence Classifier on Sentence Testset								
Sentence					.91	.86	.95	.60

Table 2: DARIUS F1 score. Highest and lowest score per column are bold.

Variant	Token Classifier on Token Testset							
	Overall	Lead	Position	Claim	Counterclaim	Rebuttal	Evidence	Conclusion
Full text	.57	.78	.58	.43	.48	.42	.64	.77
Sentence	.20	.41	.19	.06	.09	.00	.23	.12
Shuffled	.30	.46	.29	.20	.37	.12	.13	.22
First_25	.52	.67	.61	.39	.39	.16	.44	.00
First_50	.51	.60	.61	.38	.42	.35	.76	.00
First_75	.46	.57	.56	.31	.34	.32	.78	.14
Last_25	.11	.21	.39	.00	.50	.21	.41	.39
Last_50	.12	.33	.47	.01	.56	.36	.61	.48
Last_75	.25	.35	.46	.05	.60	.36	.71	.49
Token Classifier on Sentence Testset								
Variant	Overall	Lead	Position	Claim	Counterclaim	Rebuttal	Evidence	Conclusion
Sentence Classifier on Sentence Testset								
Sentence	.43	.39	.51	.39	.47	.26	.48	.30

Table 3: PERSUADE F1 score. Highest and lowest score per column are bold.

Variant	Overall F1			Conclusion F1		
	Orig.	Split	Hybrid	Orig.	Split	Hybrid
Full text	.92	.88 (-.04)	.90 (-.02)	.71	.54 (-.17)	.61 (-.10)
First_25	.92	.90 (-.02)	.91 (-.01)	.00	.00 (.00)	.00 (.00)
First_50	.95	.93 (-.02)	.94 (-.01)	.04	.00 (-.04)	.02 (-.02)
First_75	.95	.93 (-.02)	.94 (-.01)	.54	.40 (-.14)	.43 (-.11)
Last_25	.70	.82 (+.12)	.76 (+.06)	.30	.70 (+.40)	.52 (+.22)
Last_50	.86	.87 (+.01)	.87 (+.01)	.57	.68 (+.11)	.62 (+.05)
Last_75	.92	.89 (-.03)	.90 (-.02)	.68	.62 (-.06)	.63 (-.05)
Sentence	.82	.86 (+.04)	.86 (+.04)	.00	.49 (+.49)	.39 (+.39)
Shuffled	.75	.71 (-.04)	.74 (-.01)	.18	.22 (+.04)	.19 (+.01)

Table 4: Comparison of training strategies on DARIUS dataset. Parentheses show differences from the original classifier. Positive values indicate improvement. Bold values are substantial improvements.

The **Split** model shows that training on incomplete texts improves the handling of such conditions, although this affects the performance on complete essays. The **Hybrid** model offers a trade-off, making modest improvements for incomplete texts while largely maintaining the performance level on

complete essays.

The most notable improvements in both new models relate to conclusion detection in partial texts; this is consistent with our observation that conclusions tend to be context-dependent. Training models with conclusions in various contexts

reduces their reliance on position and directs attention more to the linguistic features of conclusive statements.

6 Conclusions and Future Work

The results suggest that providing feedback during the writing process is feasible, particularly when students write linearly from beginning to end, but struggles with incomplete or out-of-sequence texts. Context plays a crucial role in accurate classification. While sentence-level classification shows promise for handling incomplete texts, conclusion detection remains challenging as it appears most context-dependent. Our **Hybrid** training approach offers a practical compromise, showing modest improvements for incomplete texts while largely maintaining performance on complete essays. This suggests that educational feedback systems could provide reasonably accurate feedback throughout the writing process while maintaining acceptable performance on complete essays. However, a challenge remains in determining which model to use in real time. As we cannot know a priori whether a student will submit a complete or partial text or whether they have written it sequentially, selecting the optimal model becomes difficult.

Future work could explore methods for detecting completion stages of student texts, enabling dynamic model selection. Additionally, we plan to use process data such as key logs to better understand the writing process and when to provide appropriate feedback. Large language models could be used to create realistic examples of incomplete student essays for training, opposed to only truncated texts.

Limitations

We presented a preliminary study that aims to emulate learner texts in the progress of being written. We were not able to verify how learners actually write in key-logging data; this will be one of our next steps. Thus, our experiments assess possible worst-case scenarios of what incomplete texts might look like, where exactly learners are on this spectrum, is yet to be determined.

Our experiments focus on BERT-based models, chosen for their established performance in German educational contexts and bidirectional attention capabilities. Another option could be to use decoder-based models like DeBERTa. However, such models only look at the left context and thus

might not be optimal in scenarios like our where potentially the first part of an essay is still missing

Another limitation of our study is its restrictions to German and English data from Germany and the United States, limiting our finding to two well-resourced languages and only two education systems. More research targeting other languages and datasets would increase the transferability of our results.

Ethics Statement

Our datasets do not contain any new material for which we have to ensure data protection and the handling of personally identifiable information. We selected datasets that, to the best of our knowledge, handled such issues with care.

References

- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The Long-document Transformer. *arXiv preprint arXiv:2004.05150*.
- 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.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.
- Yuning Ding, Marie Bexte, and Andrea Horbach. 2022. Don't drop the topic - the role of the prompt in argument identification in student writing. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 124–133, Seattle, Washington. Association for Computational Linguistics.
- 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.
- Yuning Ding, Ruth Trüb, Johanna Fleckenstein, Stefan Keller, and Andrea Horbach. 2023. Sequence tagging in EFL email texts as feedback for language learners. In *Proceedings of the 12th Workshop on NLP for Computer Assisted Language Learning*, pages 53–62, Tórshavn, Faroe Islands. LiU Electronic Press.

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. [Neural automated essay scoring and coherence modeling for adversarially crafted input](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 263–271, New Orleans, Louisiana. Association for Computational Linguistics.

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 Ingver Höft, Yuning Ding, Jan Luca Bahr, Jennifer Meyer, and Thorben Jansen. 2024b. [DARIUS: A comprehensive learner corpus for argument mining in German-language essays](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4356–4367, Torino, Italia. ELRA and ICCL.

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.

Dietrich Trautmann, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. [Fine-grained argument unit recognition and classification](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9048–9056.

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.

Thiemo Wambsgans, Christina Niklaus, Matthias Cetto, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2020. [AI: An adaptive learning support system for argumentation skills](#). In *Proceedings of the 2020 CHI Conference on Human*

Factors in Computing Systems, CHI ’20, page 1–14, New York, NY, USA. Association for Computing Machinery.

Thiemo Wambsganß, Tobias Kueng, Matthias Söllner, and Jan Marco Leimeister. 2021. [Arguetutor: An adaptive dialog-based learning system for argumentation skills](#). In *CHI ’21: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13, New York, NY, USA. Association for Computing Machinery.

A Appendix

Variant	#Token	#Intro	#Main	#Conc.
Full text	155,502	18,346	123,477	13,679
Sentence	155,502	18,346	123,477	13,679
Shuffled	155,502	18,346	123,477	13,679
First_25	45,600	16,258	29,342	0
First_50	81,753	17,792	63,931	30
First_75	122,472	18,164	102,088	2,220
Last_25	47,203	351	34,203	12,649
Last_50	83,452	834	68,939	13,679
Last_75	124,225	4,082	106,464	13,679
Variant	#Sent.	#Intro	#Main	#Conc.
Full text	8,296	1,150	6,464	682
Sentence	8,296	1,150	6,464	682
Shuffled	8,296	1,150	6,464	682
First_25	2,411	1,013	1,398	0
First_50	4,377	1,121	3,254	2
First_75	6,575	1,142	5,333	100
Last_25	2,411	16	1,758	637
Last_50	4,377	48	3,647	682
Last_75	6,575	264	5,629	682

Table A.5: Count of tokens and sentences for Introduction, Main Part and Conclusion in DARIUS Benchmarks.

- *lead*: Opening hook that guides to thesis
- *position*: Core argument on the topic
- *claim*: Supporting point for position
- *counterclaim*: Opposing viewpoint
- *rebuttal*: Defense against counterclaim
- *evidence*: Support for any argument
- *concluding statement*: Summarizing paragraph

Table A.6: The argumentative components of Crossley et al. (2022)

Variant	Example
full text	CO2 emissions and greenhouse gas emissions are to be drastically reduced as they have had a major impact on our climate change in recent years. But how can we do this work as efficiently as possible to meet the energy demand in this district? There are three options to choose from. The construction of a wind energy park , a solar plant or a hydroelectric power plant . Which of these three projects should be supported ? if one considers the first criterion, i.e. efficiency, the hydropower plant is clearly in the lead with 70 - 90% . The solar park is the weakest here . The second criterion, annual yield, is close because both the wind farm and the hydropower plant have good annual yields . And so it goes on with the other criteria so that one can say that the construction of a hydropower plant is the most efficient in relation to the construction of an energy converter . The only major drawback is the price of the project, which brings us back to the wind farm because the price difference alone is a whopping 55 million euros . And the criteria are not so bad that you can't compensate for the deficits over the years. In conclusion, it can be said that due to the enormous price, a wind farm is the best way to produce electricity for a region that is not so large .
sentence	CO2 emissions and greenhouse gas emissions are to be drastically reduced as they have had a major impact on our climate change in recent years.
shuffled	Which of these three projects should be supported ? There are three options to choose from. The construction of a wind energy park , a solar plant or a hydroelectric power plant . And the criteria are not so bad that you can't compensate for the deficits over the years. But how can we do this work as efficiently as possible to meet the energy demand in this district? In conclusion, it can be said that due to the enormous price, a wind farm is the best way to produce electricity for a region that is not so large . The second criterion, annual yield, is close because both the wind farm and the hydropower plant have good annual yields . And so it goes on with the other criteria so that one can say that the construction of a hydropower plant is the most efficient in relation to the construction of an energy converter . The only major drawback is the price of the project, which brings us back to the wind farm because the price difference alone is a whopping 55 million euros . If one considers the first criterion, i.e. efficiency, the hydropower plant is clearly in the lead with 70 - 90%. The solar park is the weakest here . CO2 emissions and greenhouse gas emissions are to be drastically reduced as they have had a major impact on our climate change in recent years.
first_25	CO2 emissions and greenhouse gas emissions are to be drastically reduced as they have had a major impact on our climate change in recent years. But how can we do this work as efficiently as possible to meet the energy demand in this district?
first_50	CO2 emissions and greenhouse gas emissions are to be drastically reduced as they have had a major impact on our climate change in recent years. But how can we do this work as efficiently as possible to meet the energy demand in this district? There are three options to choose from. The construction of a wind energy park , a solar plant or a hydroelectric power plant .
first_75	CO2 emissions and greenhouse gas emissions are to be drastically reduced as they have had a major impact on our climate change in recent years. But how can we do this work as efficiently as possible to meet the energy demand in this district? There are three options to choose from. The construction of a wind energy park , a solar plant or a hydroelectric power plant . Which of these three projects should be supported ? if one considers the first criterion, i.e. efficiency, the hydropower plant is clearly in the lead with 70 - 90% . The solar park is the weakest here .
last_25	And the criteria are not so bad that you can't compensate for the deficits over the years. In conclusion, it can be said that due to the enormous price, a wind farm is the best way to produce electricity for a region that is not so large .
last_50	The second criterion, annual yield, is close because both the wind farm and the hydropower plant have good annual yields . And so it goes on with the other criteria so that one can say that the construction of a hydropower plant is the most efficient in relation to the construction of an energy converter . The only major drawback is the price of the project, which brings us back to the wind farm because the price difference alone is a whopping 55 million euros . And the criteria are not so bad that you can't compensate for the deficits over the years. In conclusion, it can be said that due to the enormous price, a wind farm is the best way to produce electricity for a region that is not so large
last_75	Which of these three projects should be supported ? if one considers the first criterion, i.e. efficiency, the hydropower plant is clearly in the lead with 70 - 90% . The solar park is the weakest here . The second criterion, annual yield, is close because both the wind farm and the hydropower plant have good annual yields . And so it goes on with the other criteria so that one can say that the construction of a hydropower plant is the most efficient in relation to the construction of an energy converter . The only major drawback is the price of the project, which brings us back to the wind farm because the price difference alone is a whopping 55 million euros . And the criteria are not so bad that you can't compensate for the deficits over the years. In conclusion, it can be said that due to the enormous price, a wind farm is the best way to produce electricity for a region that is not so large .

Table A.7: Examples for each category in our benchmark dataset based on essay 943_n3 in the DARIUS dataset (English translation provided by the authors)

Educators' Perceptions of Large Language Models as Tutors: Comparing Human and AI Tutors in a Blind Text-only Setting

Sankalan Pal Chowdhury*
ETH Zurich

Terry Jingchen Zhang
ETH Zurich

Donya Rooein
Bocconi University

Dirk Hovy
Bocconi University

Tanja Käser
EPFL

Mrinmaya Sachan
ETH Zurich

Abstract

The rapid development of Large Language Models (LLMs) opens up the possibility of using them as personal tutors. This has led to the development of several intelligent tutoring systems and learning assistants that use LLMs as back-ends with various degrees of engineering. In this study, we seek to compare human tutors with LLM tutors in terms of engagement, empathy, scaffolding, and conciseness. We ask human tutors to annotate and compare the performance of an LLM tutor with that of a human tutor in teaching grade-school math word problems on these qualities. We find that annotators with teaching experience perceive LLMs as showing higher performance than human tutors in all 4 metrics. The biggest advantage is in empathy, where 80% of our annotators prefer the LLM tutor more often than the human tutors. Our study paints a positive picture of LLMs as tutors and indicates that these models can be used to reduce the load on human teachers in the future.

1 Introduction

Recent improvements in Large Language Models (LLMs) have opened up the possibility of using them in multiple new domains, including as personal tutors. This possibility has led to the development of several Intelligent Tutoring Systems (ITSs) and learning assistants (Schmucker et al., 2023; Liffiton et al., 2023; Lieb and Goel, 2024; Vanzo et al., 2024) that use LLMs as backends with various degrees of engineering. Surveys by Intelligent.com (Int, 2023) and DEC Singapore (DEC, 2024) indicate that a large number of students are already using LLMs like ChatGPT in educational roles such as tutoring.

Despite their popularity, a clear understanding of the pedagogical effectiveness of educational chatbots, especially compared to humans, is lacking.

The common way of using LLMs as tutor is to interact with them via a chat interface, where the LLM roleplays a tutor. It is known that the full benefit of a human tutor goes well beyond verbal or textual communication (Bambaerero and Shokrpour, 2017), giving human tutors an advantage over LLM-based tutors. However, it remains unclear how LLM-based tutors compare with their human counterparts, in this chat setting. A good tutor keeps students **engaged**, **empathises** with their struggles, **scaffolds** them to correct answers, all while keeping the conversation to the point and **concise**. Is an LLM-based tutor capable of doing the same?

In this study, we compare human tutors with LLM-based tutors, through the dialogs generated via chat interfaces. Our main research question is:

How do LLM-based tutors compare to human tutors in terms of engagement, empathy, scaffolding, and conciseness?

Although there have been some recent attempts to compare learning gains from LLM-based tutors and human tutors (see Sec 2), these studies focus on the observable outcomes of learning gains. Our study seeks to complement these studies by instead focusing only on the latent factors (we will provide a more detailed definition and justification in Sec. 3.2), and run comparisons on these directly. We believe that knowing how LLMs stand on these would allow researchers to better focus on what to improve in these models.

Our contributions are:

1. We create a setup to ask human annotators to compare tutoring dialog snippets in a blind pairwise preference selection setting.
2. We use this setup to have teachers compare a human tutor with an LLM tutor on a dataset of MWPs to identify how they compare the 4 latent factors involved in student learning.

*For queries contact spa1chowd@ethz.ch

3. We publicly release the [annotation data](#) consisting of 210 annotated dialog pairs to help future research better align LLM outputs to human judgments.

Our experiments find that annotators with teaching experience perceive the LLM tutor to be more engaging and empathetic while also being concise and better at scaffolding the student. This also aligns with LLMs self-judgments, though fine-grained tendencies are quite different.

2 Related Work

2.1 Designing and Evaluating LLM Based Tutors

With the recent progress in LLMs, there have been several efforts to develop and evaluate LLM-based tutors. A large number of these have focused on computer science and programming education (Yang et al., 2024; Qi et al., 2024; Lififton et al., 2023; Kazemitabaar et al., 2024; Jacobs and Jaschke, 2024; Liu et al., 2024; Lyu et al., 2024; Li et al., 2024; Choudhuri et al., 2023; Pankiewicz and Baker, 2024), but there have also been developments in domains like mathematics (Chowdhury et al., 2024; Butgereit et al., 2023; Pardos and Bhandari, 2024), language learning (Polakova and Klimova, 2024; Park et al., 2024; Vanzo et al., 2024), health sciences (Kavadella et al., 2024; Chheang et al., 2024; Wang et al., 2024) and other domains (Thway et al., 2024; Chen and Chang, 2024). However, most of these works focus on the engineering behind developing the tutor, and if any evaluation is done, it is either in terms of learning gains, or in the terms of student self-reports of efficacy and motivation. Moreover, the comparison in these studies is always between having an LLM-based tutor and not having anything, and not with human tutors. Finally, we also lack an understanding of the factors contributing to a good quality tutor.

2.2 Comparing AI Tutors with Human Tutors

Tutoring was established as one of the best ways to improve learning outcomes by Bloom in 1984 (Bloom, 1984), and matching the learning gains of a human tutor has been one of the main targets of computer-based tutors ever since (Sleeman and Brown, 1982). Several studies have compared the learning gains from different types of computer-based tutors with humans (Kulik and Kulik, 1991; Anderson et al., 1995; VanLehn, 2011)

and with the development of LLM-based tutors, the same has also been extended to LLM based tutors (Schmucker et al., 2023; Zhang et al., 2024). However, these works focus only on the final learning gain, not on the latent qualities that could cause it.

Since several computer-based tutors communicate in natural language, another line of work follows from Alan Turing’s Imitation Game (Turing, 1950), which was later adapted into the ‘Bystander Turing Test’ (Person and Graesser, 2002). While our work is similar to this in terms of the text-only setup and blind selection, we differ in that instead of asking the annotator to determine which party is human, we ask them to determine which one is better on a set of metrics.

3 Method

3.1 Datasets

To compare human and LLM tutors, we need parallel data sets of student-tutor text interactions, both for human and LLM tutors. Among the limited one-on-one tutoring datasets available, we cannot use data sets such as TSCC (Caines et al., 2020) or CIMA (Stasaski et al., 2020) because they use human students, and a fair comparison would require us to repeat the LLM side of the experiment with the same humans. To avoid doing this, we draw our conversations from MathDial (Macina et al., 2023) for the human side because the students in this dataset are simulated by AI.

MathDial consists of about 3000 tutor-student conversations fixing student errors on MWPs. The MWPs were sampled from the GSM8K dataset (Cobbe et al., 2021), while the misconceptions were generated using InstructGPT. The authors hired annotators with teaching experience on Prolific to converse with an InstructGPT instance pretending to be a student having the particular misconception, a setup we can easily replicate at little cost. The annotators were prescribed some pedagogical suggestions urging them to avoid giving out answers directly but were otherwise encouraged to behave as they would when tutoring a real student.

Moving on to the LLM side, we could simply use a modern LLM like GPT to repeat the conversations from MathDial with identical settings. However, this only works if we can ensure that the GPT model would never give incorrect feedback, for example stating that a student’s answer is right when it is not. If a tutor has a chance of giving out wrong information, comparing its softer qualities

is moot. Unfortunately, previous work has found that GPT4-turbo does make such mistakes (Chowdhury et al., 2024) and we found in our explorations that this is still the case for GPT4o, with 6 out of the 30 problems investigated having some issue (1 case where the teacher gave a wrong answer, 5 cases of teacher telling the right answer but not verifying if the student agreed, including 2 where the final teacher utterance included nonsensical phrases). Therefore, we instead use conversations from MWPTutor (Chowdhury et al., 2024), a tutor based on LLMs which ensures correctness by imposing guardrails on top of GPT.

MWPTutor uses a finite state transducer to prompt an LLM to generate the best teacher utterances and uses the same InstructGPT student model as MathDial. The paper proposes multiple versions of their system, but in this work, we make use of MWPTutor^{live}_{GPT4} as it is the best according to their own metrics. Although MathDial consists of about 3000 conversations, many of them repeat the same MWPs and incorrect solutions. Since MWPTutor only makes use of these two components, we restrict our study to one conversation per MWP. As such, we choose 210 MWPs including all 45 coming from the GSM8K test set. For MathDial, we pick the first conversation when sorted by timestamp. For MWPTutor, we use the conversations published by the authors for the test set MWPs, while for the remaining, we generate conversations using their publicly available code.

Note that despite the accuracy issue, we did perform a smaller study with GPT4o instead of MWPTutor, and found that the trends were not much different (See Appendix B for details).

3.2 Metrics

Tutoring is a rather complex task, and thus it is hard to list desirable tutoring qualities that can be considered universally applicable. The primary desiderata for our study are that we need a small set of metrics (so as to be able to evaluate them in a reasonable budget), which can be judged from text and are subjective enough to facilitate the comparison of two conversations. To obtain such a set of metrics, we drew inspiration from 3 main works.

Ross, in his book (MacDonald, 2000) identifies 6 goals for tutors, although this includes more administrative duties such as “provide student perspective on school success”. Walker (Walker, 2008) surveyed several teachers in training, and identified 12 desirable characteristics of teachers. Although

too numerous and often requiring actions beyond a text-only setting, they serve as a good starting point for us. Maurya et al (Maurya et al., 2024) unified several recent works to identify 8 metrics relevant to AI tutors. However, these metrics are often too precise, making it difficult to rank two conversations based on them.

Inspired by these and other works mentioned in the definitions, we decided on four metrics to evaluate, which we discuss below. An important thing to point out here is that though these metrics have scientific grounding, they are all quite subjective, which means in certain cases choosing the better of a pair of conversations might become a matter of personal preference. Although we did not evaluate the original metrics in the above work with humans, we did run them through GPT, and the results are provided in the Appendix (see Section C). We also provide a full mapping between the metrics in the three aforementioned papers and our metrics in Section D

Engagement: Student engagement can be defined as ‘how involved or interested students appear to be in their learning’ (Axelson and Flick, 2010). All of Walker, Ross and Maurya (see table 6) use metrics that map to engagement. High student engagement is positively correlated with student learning outcomes (Lei et al., 2018), and this effect has also been observed in recent studies on LLM tutors (Altememy et al., 2023; Vanzo et al., 2024).

Empathy: Empathy is the ability of a tutor to understand the hardships a student is facing and to react in a way that keeps up their motivation. Empathy is seen as important in a teacher by most educators (Stojiljković et al., 2012; Makoelle, 2019), and practical studies show that teachers’ empathy is correlated with positive learning outcomes for at least some groups of students (Bostic, 2014; D’Mello and Graesser, 2013). Walker identifies multiple dimensions of empathy as essential, while Maurya and Ross also consider it important (see table 6). One important thing to note here is that empathy in general is a rather broad term, and is often split into subcategories of emotional and cognitive empathy (Smith, 2006). In this work, ‘Empathy’ primarily refers to Emotional Empathy, whereas Cognitive Empathy is somewhat subsumed by Engagement.

Scaffolding: Scaffolding is the idea that a tutor should help a student succeed in a problem, not by directly revealing the answer, but by controlling elements of the problem solving process to enable the student to achieve the solution by themselves

(Wood et al., 1976). Doing so helps students to not just understand the solution of the problem at hand but also learn the concepts behind the solution, enabling them to solve similar problems thereafter. The first five metrics from Maurya all reflect forms of scaffolding, while Ross covers it with ‘promote independent learning’ and ‘facilitate tutee insights into learning’. Scaffolding is also a primary goal in both MathDial (called ‘Equitable Tutoring’ in the paper possibly due to conflicting terminologies) and MWPTutor.

Conciseness: While not considered an important metric by the three works we repeatedly refer to, we note that to achieve the previously mentioned metrics, one may end up with extremely long conversations. However, a good tutor should always try to make progress with a question. Having the student repeat steps already done or making them do redundant steps is known to hurt learning outcomes, especially when only a single modality (i.e., text) is available (Kalyuga and Sweller, 2014; Albers et al., 2023). Failure to make progress in a problem often leads to frustration (Goldin, 2000), which in turn can hurt learning (Chitrakar and P.M., 2023). Finally, longer conversations can lead to students going beyond their optimal attention span (Philip and Bennett, 2021) leading to bad outcomes. Therefore, we include conciseness as a fourth metric.

3.3 Setup

MathDial conversations are about 10 turns on average, while that of MWPTutor can go from 5 to 60 turns. We needed annotators to choose which conversations were better by each metric. Internal testing revealed that longer dialogs greatly increase the time to choose with people having to go back and forth in the dialogues, though the tone of the dialog is usually set within the first few turns, making it the most important part of the dialog. Thus, we decided to truncate all dialogs to 5 turns, the lower limit of the average human working memory (Miller, 1956). It also helps that both MWPTutor and MathDial require a conversation to last for at least 5 turns. This truncation, however, meant that sometimes the dialogs could be too small to judge them, so we allowed the annotators to say “Both are Equal,” but we asked them to use this sparingly. Note that this also increases the epistemic noise of the task.

Our survey was hosted on FillOut¹. The 210

¹fillout.com

problems were divided into 7 batches of 30 conversation pairs each, which would take 45-60 minutes each of annotator time. The survey started off with a task description, followed by metric descriptions. Thereafter, it we had 150 slides, 5 per conversation pair. The first slide introduced the new MWP and the two conversations, and the next 4 went over the 4 metrics. These slides showed the MWP, the two conversations side-by-side, and a short description of the current metric, and asked the user to pick one of “Left is Better”, “Right is Better”, and, “Both are Equal” (see Section G for details). The right-left positioning of the conversations was randomized to avoid bias. Annotators were instructed to focus on the tutor’s utterances and not the student’s. In addition, we did not explain the nature of the tutors or students and there was no indication that any of the parties were LLM agents. We also had three LLMs, namely GPT4 (gpt-4o-2024-08-06, (OpenAI et al., 2024)), Qwen (Qwen/Qwen2.5-72B-Instruct-Turbo from together.ai, (Qwen et al., 2025)), Llama (meta-llama/Meta-Llama-3.1-405B-Instruct-Turbo from together.ai, (Touvron et al., 2023)) compare the conversations on our metrics. For this, the prompts included the same metric definitions, and the two conversation snippets were presented as ‘System 1’ and ‘System 2’. Each conversation-pair was run through each LLM twice, with the order of conversations flipped to avoid biases.

3.4 Participants

Each batch was annotated by 5 annotators, bringing us to a total of 35 annotators. We initially hired Prolific 21 annotators who had access to a computer, were fluent in English, and had some teaching experience. These requirements are identical to those set in MathDial. We also hired two more sets of 7 annotators, one consisting of only men and one consisting of only people aged 50 or older to get a better distribution of age and gender. All annotators were paid the Prolific recommended rate of GBP 9 for the survey.

Demographics Of the 35 Prolific-hired annotators, 14 identified as male while the rest identified as female. The dominant self-identified ethnicity was black (20 annotators), with white (11 annotators) being the next closest. Their ages range from 20 to 74, with median age being 34.

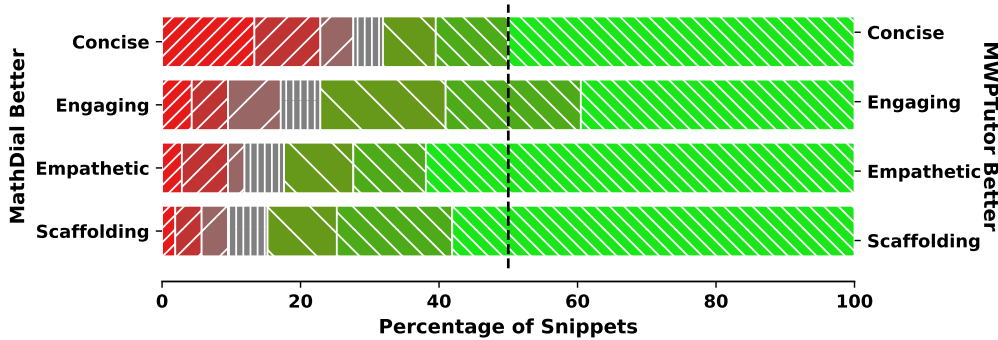


Figure 1: Fractions of conversation pairs which received particular scores for each metric from LLMs. Scores increase left to right, with the **brightest red** indicating minimum possible score of -3 , the **dullest red** indicating -1 , **grey** indicating 0 , the **dullest green** indicating $+1$ and the **brightest green** indicating the maximum possible score of $+3$

4 Results and Analysis

We mentioned earlier that our metrics involve some scope for personal choice. This means that disagreements between annotators would involve some epistemic uncertainty. To account for this, instead of dealing with the point measures given by majority voting, we look at the full set of votes through the notion of *score*.

For each metric and each conversation pair, an annotator must pick one of “Left is Better”, “Right is Better” and “Both are Equal”, which we can map into “MWPTutor is Better”, “MathDial is Better” or “Both are Equal”. We assign a value of 1 to “MWPTutor is Better” and a value of -1 to “MathDial is Better”, while “Both are Equal” gets a 0 . The score for a metric for a conversation pair is then the sum of all the annotator values. Thus, since we have 5 human annotators per conversation pair, a score of -5 for a conversation pair on a metric indicates that all human annotators favor MathDial for that metric, while a score of 5 indicates that all human annotators favor MWPTutor. The same is true for the LLM case, except that there are only 3 LLMs, so the scores go from -3 to 3 . Note that this *score* is only introduced for analysis in this paper, and was not used in the actual surveys.

4.1 LLM ratings

Figure 1 shows the distribution of the ratings given by the 3 LLMs for our 210 instances. All responses were queried in December 2024. While no LLM picked the ‘Both are Equal’ option, we had multiple cases where changing the order of the conversations changed the LLM’s answer, so we considered these cases to be ‘Both are Equal’. We see that the LLMs overwhelmingly favor MWPTutor on all 4 metrics. The individual behavior of the LLMs does

not seem very different from each other (see Table 3 in the Appendix for details). While these lopsided results are definitely interesting, it might not be too decisive, considering that LLMs are likely to be biased towards LLM-generated text.

4.2 Human Ratings

Figure 2 shows the outcome of the human ratings while Table 1 shows the agreement between annotators and significance statistics. Although the results are much less lopsided than the LLM annotations, the outcome is the same. MWPTutor performs better on all metrics, with the difference being significant² for all metrics except Engagement. As expected, the agreement amongst annotators is low, a testament to the complexity of the task.

4.3 Alignment Between LLMs and Humans

Another interesting thing to note here is the difference between human annotations and LLM annotations. While both come to the conclusion that MWPTutor is doing better on all metrics, the LLMs’ opinions are much stronger than their human counterparts. Figure 3 shows the correlation between the average scores for all 4 metrics, annotated by humans and LLMs. We can see that all the squares in top-right and bottom-left quadrants, which indicate the correlations between human-annotated metrics and LLM-annotated metrics, are very dull, indicating a large difference between what LLMs perceive as good and what humans perceive as good. Also of note is the fact that the off-diagonal elements in the top-left and bottom-right quadrant are quite bright, which means that the metrics are not all disentangled, either by definition or by perception

²here and in the rest of the paper we treat anything with a p-value of 0.01 or lower as significant

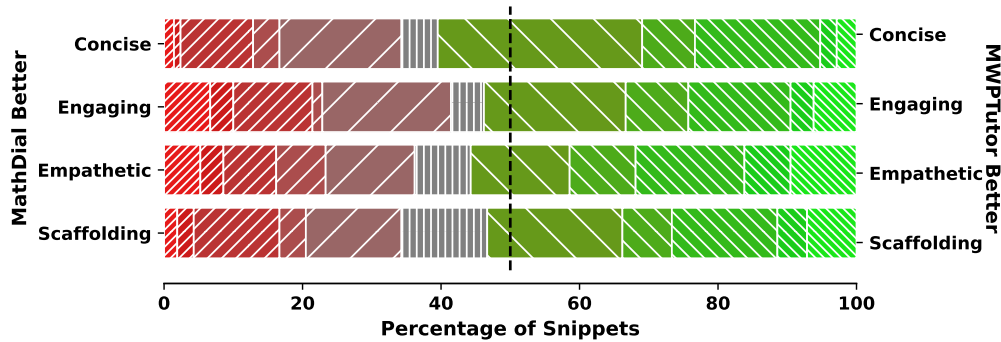


Figure 2: Fractions of conversation pairs which received particular scores for each metric from humans. Scores increase left to right, with the **brightest red** indicating minimum possible score of -5 , the **dullest red** indicating -1 , **grey** indicating 0 , the **dullest green** indicating $+1$ and the **brightest green** indicating the maximum possible score of $+5$

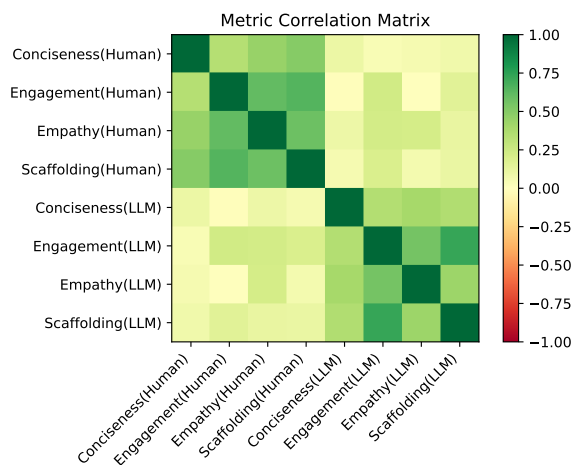


Figure 3: Correlation between various metrics, as annotated by humans and LLMs

or a combination of both.

4.4 Analysis

We now go over each of our 4 metrics and look at how the ratings they received sit in context of other quantitative metrics.

Conciseness: In terms of t-score, the metric where MWPTutor dominates the most is Conciseness. This is surprising, as unlike the annotators for MathDial, the LLM behind MWPTutor had no reason to keep conversations short. In fact, we find that MWPTutor conversations were longer in terms of the number of utterances in 135 cases, compared to 68 cases where MathDial conversations were longer. Further, when the MWPTutor conversation is shorter, it has a 74% chance of being picked as more concise, while if the MathDial conversation is shorter, it has only a 40% chance of being picked as more concise. In other words, while true conversation length is correlated with perceived

conciseness, it isn't a very strong predictor.

Empathy: Human empathy can often take non-verbal modes, so judging it from a small conversation snippet can be a bit noisy. This is expressed as the high standard deviation in the Empathy scores. Nevertheless, annotators perceived MWPTutor to be more empathetic. On running sentiment analysis by huggingface pipelines³ we found a positive correlation between higher empathy scores and *joy* ($R = 0.36$, $p = 5E - 8$) and a negative correlation with *anger* ($R = -0.32$, $p = 3E - 6$)⁴ which is consistent with what we would expect. In addition, GPT4 agrees that MWPTutor shows significantly more joy and less anger compared to MathDial.

Engagement: Engagement is the only metric where the LLM's advantage is not significant. Looking at the code for MWPTutor⁵ we find that there are two ways⁶ it can start a conversation. If the student solution partially matches a stored solution, it starts by pointing out the step up to which the student is correct and proceeds from there. If no part of the solution matches, MWPTutor will start afresh by ignoring the student solution. Let us call these two scenarios *Continue* and *Fresh* respectively. In the 45.5% conversations in the *Continue* scenario, the average Engagement score is 1.42, so MWPTutor is significantly better than MathDial

³Sentiment scores were calculated by averaging the score for each tutor utterance in a conversation snippet, and then subtracting the MathDial Score from the MWPTutor score. We used the `bhadresh-savani/distilbert-base-uncased-emotion`.

⁴Taking max score across all tutor utterances also gives the same outcome, albeit the exact numbers are a bit different

⁵in particular, `the LiveTutor.start_conversation() method in models/Tutor.py`

⁶there's a 3rd to deal with correct solutions, but that was never triggered (by design)

Metric	Fleiss Kappa	Mean Score	Standard Deviation	Effect Size	t-score	p-value (1-sided)
Conciseness	0.11	0.55	2.19	0.25	3.65	<0.001
Engagement	0.22	0.25	2.72	0.09	1.32	0.09
Empathy	0.25	0.65	2.81	0.23	3.36	<0.001
Scaffolding	0.17	0.55	2.51	0.22	3.16	<0.001

Table 1: Statistics of the Human Ratings. Fleiss Kappa is calculated assuming each annotator to be a combination of two annotators, who vote opposite to each other if the actual vote is ‘Both Are Equal’

No. of Scaffolding Utterances	Sample Size	Average Score			
		Conciseness	Engagement	Empathy	Scaffolding
0	7	1.00	1.00	1.00	1.86
1	51	0.16	0.27	0.04	0.18
2	116	0.47	-0.03	0.67	0.41
3	36	1.28	0.97	1.39	1.28

Table 2: Human Annotation Scores by scaffolding utterances in MathDial snippet

in this case ($d = 0.68$, $p < 1e - 8$). However, in the 55.5% conversations in the *Fresh* scenario, the average Engagement score falls to -0.84 , so MathDial comes out on top ($d = 0.30$, $p = 0.001$). We posit that since our annotators are not given access to the student solution, they see no reason why the tutor should start afresh. Therefore, when they see the *Fresh* scenario, they perceive it as the tutor failing to engage with the student’s solution, thereby penalizing it.

We previously mentioned how conversational uptake is similar to our definition of engagement, so to get another view of the data, we calculated the difference of uptake scores for each conversation pair. We excluded the first teacher utterance because uptake requires a previous utterance. The difference in uptake scores had only a mild correlation of 0.06 with the human-annotated Engagement score, but showed a significant difference between MWPTutor and MathDial ($d = 0.20$, $p = 0.004$) with MWPTutor coming out on top.

Scaffolding: As stated above, scaffolding is a primary focus for both MathDial and MWPTutor. In MathDial, annotators were asked to state the intent of their upcoming utterance as one of the 4 possible dialog acts. Two of these acts, namely, ‘focus’ and ‘probing’ are types of scaffolding, and in the subset of utterances we used for our annotations, these two acts combined make up about 62% of all teacher utterances. This clearly shows that the annotators from MathDial made an effort at scaffolding, but somehow fell short of MWPTutor.

To further analyze this, we grouped the conversation pairs by how many scaffolding utterances were present in the MathDial Snippet of the pair and calculated the average score for each metric including scaffolding. The results are shown in Table 2. Excluding the first row, which contains only 7 sam-

ples, the average score for scaffolding surprisingly increases (i.e., becomes less favorable to MathDial) with the number of scaffolding utterances. In other words, *a higher number of scaffolding utterances makes it worse at scaffolding* as perceived by our annotators. Although we are unsure of the cause for this, it does indicate that despite expressing the intent to scaffold, the MathDial annotators were unable to follow through. Conversations with a higher number of scaffolding utterances are also perceived to be less concise and less empathetic, the former of which makes some sense since introducing more scaffolding might reduce progress made.

5 Discussion

5.1 Human Tutors Appear Less Concise, Despite Being More

Since the annotators had access to only small parts of the conversation, the guidelines instructed them to focus on the amount of progress made in the given part of the dialog. We propose two possible causes of the difference between perceived conciseness and true conversation length.

First, it is possible that **human tutors tend to start slow and then make faster progress in the part of the conversation not shown to the annotators**. While this might indicate a failure of our annotation setup, varying the rate of progress is not necessarily a good strategy. Conciseness is meant to avoid frustration and boredom; a slower start might cause real students to get bored and disengaged, making it harder to make progress later, a behavior not replicated by the LLM student used here. Another concern might be the fact that the increased progress in the later parts of the conversation might come due to an increase in the level of telling, which is consistent with Fig. 4 in the Mathdial paper (Macina et al., 2023). As an example, while human annotators agreed that none of the 45 test set conversations from MWPTutor had any telling involved, the corresponding 45 conversations from MathDial had a total of 40 teacher utterances marked as telling.

Also, perhaps **MWPTutor frames its responses in a way that makes it look like it is making progress despite that not really being the case**. This could mean that MWPTutor being more engaging or scaffolding better is perceived as being more concise. Given that the agreement of the same annotator annotating different metrics is consistently higher than the agreement of different annotators

annotating the same metric⁷ this is not unlikely.

5.2 Being a Good Teacher is Exhausting, but not Rewarding Enough

A possible reason why human teachers might not be able to show empathy could be the fact that empathy comes at a cognitive cost (Cameron et al., 2019) and thereby must be used selectively. A human tutor who would potentially be dealing with hundreds of students during their teaching career could develop compassion fatigue (Yu et al., 2022) as well as other forms of burnout (Jacobson, 2016) causing them to lack empathy for students. The same can also be said for the Scaffolding and engagement results - when a teacher sees the same mistakes being made by students repeatedly, they are likely to want to simply give out the correct answer, rather than engage the student by scaffolding them in more innovative ways. The fact that being more empathetic and engaging, or scaffolding better, rarely carries financial incentives (which is true for MathDial) makes teachers even less likely to show these qualities. An LLM, however, is not bound by the same cognitive limitations of a human, and can thereby show (or pretend to show) infinite compassion and empathy. It also does not mind engaging the student more and scaffolding them better, because it is, after all, being paid by the token. Note that the fact that the MathDial annotators participating in a study and not dealing with actual students may have further exacerbated this issue. Knowing that the student is in fact an AI which will not get demoralized or disengage might have contributed to the teachers not doing their best. Add to this the fact being restricted to typing only might hinder their ability to show empathy.

5.3 Bad Spelling or Grammar Might Look Less Engaging

Although the observed difference might be due to chance in the case of Engagement, the presence of lexical and grammatical mistakes might also play a role. Due to the lack of any spell-check or grammar correction tool, the human responses ended up containing several typos, missing capitalizations, punctuation, and other grammatical errors, which our annotators (and hypothetical students) might find distracting and thereby disengaging.

⁷This is calculated by flipping the annotator and metric axes while calculating Fleiss κ . This is done for illustrative purposes only, and not the proper way to use Fleiss κ

5.4 So, What Are The Takeaways?

This study shows that LLMs are capable of performing certain tutoring roles well, perhaps as well as humans. However, we need to think what this really means for the stakeholders. We believe that there are two major takeaways – one for educators and one for learning scientists.

For educators, the emergence of AI means increased opportunities for delegation. It is a well known fact that a teacher’s duty extends well beyond teaching, with them often having to act as mentors and guardians of students (Tea, 2024; Tabassum and Alam, 2024; Kutsyuruba and Godden, 2019). Allowing AIs like LLMs to take over repetitive yet exhausting duties can allow teachers to focus more on such responsibilities which require socio-cultural understanding well beyond the capabilities of AI. It also can bring a sense of fulfillment to educators, potentially mitigating some teacher fatigue (Zang and Chen, 2022).

For learning scientists, it adds to several other works indicating that we are making fast and effective progress towards computer-based education. LLMs are able to show (or at least imitate) qualities once considered hard for them. Yet, the job is far from done – we are only dealing here with textual capabilities, while a human teacher uses several communication modalities. Progress needs to be made in image processing, vocal intonations, embodiment, etc. to fully replicate the more mundane roles of educators.

6 Conclusion

In this study, we asked educators to compare parts of human-generated tutoring conversations with LLM generated ones in a blind setting. We found that in the limited setting of text-only tutoring, most educators *perceived* that the LLM was not only matching humans, but also *outperforming* them in several quasi-metrics for teaching quality. We further find that the LLM’s perception of what is good tutoring is still not perfectly aligned with humans. This shows that there is still scope to improve self-judgment abilities of LLMs, which could further improve the quality of LLM tutoring. Thus, overall, our study paints a positive picture – with further research, it could be possible for teachers to delegate more tiring tasks in tutoring to AI, and focus on their more complex tasks, thereby improving experiences of both teachers and students.

Limitations

Despite our best efforts to make the study as comprehensive as possible, we are left with several limitations which we were unable to rectify. Some of these are:

- **Limited Setting:** We restricted ourselves to a text-only setting, while some of the metrics used, especially empathy and engagement, involve other aspects of embodied interaction like body language, expression, voice modulation, etc. The primary reason behind this is that most LLMs currently being restricted to this setting only
- **Limited Domain:** Even within the text-only domain, we restricted ourselves to one type of question (MWP) and one LLM tutor (MWP-Tutor), which may not be ideal since results might be different for different subjects, and also for differently designed tutors. While it would have been good to try out different subjects, we were unable to do so due to a lack of datasets. In order for the conversations to be comparable, we needed datasets with human and AI attempts at the same conversations, which we could not find for any other domain, and creating one from scratch would be significantly out of the scope of our abilities.
- **Unverified Qualifications:** We hired our annotators on Prolific, and filtered for those who had teaching experience. However, Prolific does not verify annotator qualifications, which means we might have had some non-educators in our annotator pool. Note that the same issue could also be present with MathDial, who also hired annotators on Prolific.
- **Qualitative Analysis:** Despite drawing from the literature, our analysis of annotator judgments is mostly intelligent guessing, as we do not know why annotators did what they did. We attempted to get some insights by interviewing some of the annotators post-hoc but had too few respondents to proceed.

In general we acknowledge that there might be several factors affecting the ecological validity of the results. While the results are statistically significant and theoretically feasible, they aren't infallible, and thereby, should not be trusted blindly if deciding

on a high-stakes scenario. A proper study with real students and teachers in a more natural setting might be the ideal scenario to draw more definitive conclusions. However, doing such an experiment was beyond the means of the authors at the time of publication.

Acknowledgments

The authors thank everyone who helped in the creation and refinement of this paper. Sankalan Pal Chowdhury is partially supported by the ETH-EPFL JDPLS. Donya Rooein and Dirk Hovy were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 949944, INTEGRATOR). They are members of the MilaNLP group and the Data and Marketing Insights Unit of the Bocconi Institute for Data Science and Analysis (BIDSA).

7 Ethics Statement

This study was approved by The ETH Zurich Ethics Commission under the title "Project 24 ETHICS-369: Comparing AI and Human Tutors"

References

- 2023. New survey finds students are replacing human tutors with chatgpt. <https://www.intelligent.com/new-survey-finds-students-are-replacing-human-tutors-with-chatgpt/>.
- 2024. Digital education council global ai student survey. <https://www.digitaleducationcouncil.com/post/digital-education-council-global-ai-student-survey-2024>.
- 2024. The multifaceted roles of a teacher: Beyond the classroom. <https://teachers.institute/learning-teaching/roles-of-teacher-beyond-classroom/>.
- Fabian Albers, Melanie Trypke, Ferdinand Stebner, Joachim Wirth, and Jan L. Plass. 2023. [Different types of redundancy and their effect on learning and cognitive load](#). *British Journal of Educational Psychology*, 93(S2):339–352.
- Haady Abdilnabi Altememy, Nour Raheem Neamah, Rabaa Mazhair, Nada Sami Naser, Ali Afrawi Fahad, Nazar Abdulghffar al sammarraie, Hidab Rasul Sharif, Mohamed Amer Alseidi, and Mohammed Yousif Oudah Al-Muttar. 2023. Ai tools' impact on student performance: Focusing on student motivation & engagement in iraq. *Przestrzeń Społeczna (Social Space)*, 23(2):143–165.

- JR Anderson, AT Corbett, KR Koedinger, and R Pelletier. 1995. [Cognitive tutors: Lessons learned](#). *Journal of the Learning Sciences*, 4(2):167–207.
- Rick D. Axelson and Arend Flick. 2010. [Defining student engagement](#). *Change: The Magazine of Higher Learning*, 43(1):38–43.
- Fatemeh Bambaerloo and Nasrin Shokrpour. 2017. The impact of the teachers’ non-verbal communication on success in teaching. *Journal of advances in medical education & professionalism*, 5(2):51.
- 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 B Bostic. 2014. Teacher empathy and its relationship to the standardized test scores of diverse secondary english students. *Journal of Research in Education*, 24(1):3–16.
- Laurie Butgereit, Herman Martinus, and Muna Mahmoud Abugosseisa. 2023. [Prof pi: Tutoring mathematics in arabic language using gpt-4 and whatsapp](#). In *2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES)*, pages 000161–000164.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. [The teacher-student chatroom corpus](#). In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20, Gothenburg, Sweden. LiU Electronic Press.
- C Daryl Cameron, Cendri A Hutcherson, Amanda M Ferguson, Julian A Scheffer, Eliana Hadjiandreou, and Michael Inzlicht. 2019. Empathy is hard work: People choose to avoid empathy because of its cognitive costs. *Journal of Experimental Psychology: General*, 148(6):962.
- Ching-Huei Chen and Ching-Ling Chang. 2024. Effectiveness of ai-assisted game-based learning on science learning outcomes, intrinsic motivation, cognitive load, and learning behavior. *Education and Information Technologies*, pages 1–22.
- Vuthea Chheang, Shayla Sharmin, Rommy Marquez-Hernandez, Megha Patel, Danush Rajasekaran, Gavin Caulfield, Behdokht Kiafar, Jicheng Li, Pinar Kullu, and Roghayeh Leila Barmaki. 2024. [Towards anatomy education with generative ai-based virtual assistants in immersive virtual reality environments](#). *Preprint*, arXiv:2306.17278.
- Nandita Chitrakar and Dr P.M. 2023. [Frustration and its influences on student motivation and academic performance](#). *International Journal of Scientific Research in Modern Science and Technology*, 2:01–09.
- Rudrajit Choudhuri, Dylan Liu, Igor Steinmacher, Marco Gerosa, and Anita Sarma. 2023. [How far are we? the triumphs and trials of generative ai in learning software engineering](#). *Preprint*, arXiv:2312.11719.
- Sankalan Pal Chowdhury, Vilém Zouhar, and Mrinmaya Sachan. 2024. [Autotutor meets large language models: A language model tutor with rich pedagogy and guardrails](#). In *ACM Conference on Learning @ Scale*.
- 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.
- Sidney D’Mello and Art Graesser. 2013. Autotutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(4):1–39.
- Gerald A. Goldin. 2000. [Affective pathways and representation in mathematical problem solving](#). *Mathematical Thinking and Learning*, 2(3):209–219.
- Sven Jacobs and Steffen Jaschke. 2024. [Evaluating the application of large language models to generate feedback in programming education](#). In *2024 IEEE Global Engineering Education Conference (EDUCON)*. IEEE.
- Donna Ault Jacobson. 2016. *Causes and effects of teacher burnout*. Walden University.
- Slava Kalyuga and John Sweller. 2014. *The Redundancy Principle in Multimedia Learning*, page 247–262. Cambridge Handbooks in Psychology. Cambridge University Press.
- Argyro Kavadella, Marco Antonio Dias da Silva, Eleftherios G Kaklamanos, Vasileios Stamatopoulos, and Kostis Giannakopoulos. 2024. [Evaluation of chatgpt’s real-life implementation in undergraduate dental education: Mixed methods study](#). *JMIR Med Educ*, 10:e51344.
- Majeed Kazemitabaar, Runlong Ye, Xiaoning Wang, Austin Zachary Henley, Paul Denny, Michelle Craig, and Tovi Grossman. 2024. [Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs](#). In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI ’24. ACM.
- Chen-Lin C. Kulik and James A. Kulik. 1991. [Effectiveness of computer-based instruction: An updated analysis](#). *Computers in Human Behavior*, 7(1):75–94.
- Benjamin Kutsyuruba and Lorraine Godden. 2019. The role of mentoring and coaching as a means of supporting the well-being of educators and students. *International Journal of Mentoring and Coaching in Education*, 8(4):229–234.

- Hao Lei, Yunhuo Cui, and Wenye Zhou. 2018. [Relationships between student engagement and academic achievement: A meta-analysis](#). *Social Behavior and Personality: an international journal*, 46:517–528.
- Wengxi Li, Roy Pea, Nick Haber, and Hari Subramonyam. 2024. [Tutorly: Turning programming videos into apprenticeship learning environments with llms](#). *Preprint*, arXiv:2405.12946.
- Anna Lieb and Toshali Goel. 2024. Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8.
- Mark Liffiton, Brad E Sheese, Jaromir Savelka, and Paul Denny. 2023. Codehelp: Using large language models with guardrails for scalable support in programming classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, pages 1–11.
- Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J. Malan. 2024. [Teaching cs50 with ai: Leveraging generative artificial intelligence in computer science education](#). In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2024, page 750–756, New York, NY, USA. Association for Computing Machinery.
- Wenhan Lyu, Yimeng Wang, Tingting (Rachel) Chung, Yifan Sun, and Yixuan Zhang. 2024. [Evaluating the effectiveness of llms in introductory computer science education: A semester-long field study](#). In *Proceedings of the Eleventh ACM Conference on Learning @ Scale, L@S '24*. ACM.
- Ross B MacDonald. 2000. *The master tutor: A guidebook for more effective tutoring*. Cambridge Stratford Study Skills Institute.
- 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.
- Tsediso Michael Makoelle. 2019. Teacher empathy: a prerequisite for an inclusive classroom. *Encyclopedia of teacher education*, 11(2):27–39.
- Kaushal Kumar Maurya, KV Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2024. Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors. *arXiv preprint arXiv:2412.09416*.
- George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Maciej Pankiewicz and Ryan S. Baker. 2024. [Navigating compiler errors with ai assistance - a study of gpt hints in an introductory programming course](#). In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, ITiCSE 2024. ACM.
- Zachary A. Pardos and Shreya Bhandari. 2024. [Chatgpt-generated help produces learning gains equivalent to human tutor-authored help on mathematics skills](#). *PLOS ONE*, 19(5):1–18.
- Minju Park, Sojung Kim, Seunghyun Lee, Soonwoo Kwon, and Kyuseok Kim. 2024. [Empowering personalized learning through a conversation-based tutoring system with student modeling](#). In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, CHI '24*. ACM.
- Natalie Person and Arthur C. Graesser. 2002. Human or computer? autotutor in a bystander turing test. In *Intelligent Tutoring Systems*, pages 821–830, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Abey P Philip and Dawn Bennett. 2021. Using deliberate mistakes to heighten student attention. *Journal of University Teaching and Learning Practice*, 18(6):193–212.
- Petra Polakova and Blanka Klimova. 2024. [Implementation of ai-driven technology into education – a pilot study on the use of chatbots in foreign language learning](#). *Cogent Education*, 11(1):2355385.
- Laryn Qi, J. D. Zamfirescu-Pereira, Taehan Kim, Björn Hartmann, John DeNero, and Narges Norouzi. 2024. [A knowledge-component-based methodology for evaluating ai assistants](#). *Preprint*, arXiv:2406.05603.
- Qwen, :, 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, Jiayi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2023. [Ruffle&riley: Towards the automated induction of conversational tutoring systems](#). *arXiv preprint arXiv:2310.01420*.
- D. Sleeman and J.S. Brown. 1982. *Intelligent Tutoring Systems*. Computers and people series. Academic Press.

Adam Smith. 2006. Cognitive empathy and emotional empathy in human behavior and evolution. *The Psychological Record*, 56(1):3–21.

Katherine Stasaski, Kimberly Kao, and Marti A. Hearst. 2020. CIMA: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, Seattle, WA, USA → Online. Association for Computational Linguistics.

Snežana Stojiljković, Gordana Djigić, and Blagica Zlatković. 2012. Empathy and teachers’ roles. *Procedia - Social and Behavioral Sciences*, 69:960–966. International Conference on Educational Psychology (ICEEPSY 2012).

Mayeasha Tabassum and Md Mahbub-ul Alam. 2024. Exploring multifaceted expectations from teachers: An analysis from guardians’ and students’ perspective. *Indonesian Journal of Social Research (IJSR)*, 6(2):141–155.

Maung Thway, Jose Recatala-Gomez, Fun Siong Lim, Kedar Hippalgaonkar, and Leonard W. T. Ng. 2024. Battling botpoop using genai for higher education: A study of a retrieval augmented generation chatbots impact on learning. *Preprint*, arXiv:2406.07796.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

A. M. Turing. 1950. I.—computing machinery and intelligence. *Mind*, LIX(236):433–460.

Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.

Alessandro Vanzo, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024. Gpt-4 as a homework tutor can improve student engagement and learning outcomes. *arXiv preprint arXiv:2409.15981*.

Robert J Walker. 2008. Twelve characteristics of an effective teacher: A longitudinal, qualitative, quasi-research study of in-service and pre-service teachers’ opinions. *educational HORIZONS*, pages 61–68.

Ruiyi Wang, Stephanie Milani, Jamie C. Chiu, Jiayin Zhi, Shaun M. Eack, Travis Labrum, Samuel M. Murphy, Nev Jones, Kate Hardy, Hong Shen, Fei Fang, and Zhiyu Zoey Chen. 2024. Patient-Ψ: Using large language models to simulate patients for training mental health professionals. *Preprint*, arXiv:2405.19660.

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.

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. *Preprint*, arXiv:2406.13972.

Xiajun Yu, Changkang Sun, Binghai Sun, Xuhui Yuan, Fujun Ding, and Mengxie Zhang. 2022. The cost of caring: Compassion fatigue is a special form of teacher burnout. *Sustainability*, 14(10).

Lingling Zang and Yameng Chen. 2022. Relationship between person-organization fit and teacher burnout in kindergarten: the mediating role of job satisfaction. *Frontiers in Psychiatry*, 13:948934.

Jiayue Zhang, Yiheng Liu, Wenqi Cai, Lanlan Wu, Yali Peng, Jingjing Yu, Senqing Qi, Taotao Long, and Bao Ge. 2024. Investigation of the effectiveness of applying chatgpt in dialogic teaching using electroencephalography. *Preprint*, arXiv:2403.16687.

A Ratings from Individual LLMs

Model	Rating	Number of Questions			
		Conciseness	Engagement	Empathy	Scaffolding
GPT4o	MathDial Better	56	28	19	9
	Both Equal	29	35	37	48
	MWPTutor Better	125	147	154	153
Qwen72B	MathDial Better	32	20	14	19
	Both Equal	38	41	38	27
	MWPTutor Better	140	149	158	164
Llama405B	MathDial Better	60	49	30	18
	Both Equal	26	53	25	41
	MWPTutor Better	124	108	155	151
Fleiss Kappa		0.63	0.34	0.47	0.36

Table 3: LLM based comparison of MathDial and MWPTutor on our metrics. Fleiss κ is calculated considering the two annotations by each LLM to be separate instances

See table 3

B MathDial vs GPT4

We carried out a smaller experiment using only the questions from the first batch (30 examples). The setup is identical to what we described in Section 3.3. The results are shown in figure 4. Despite the smaller sample size, the results are very decisive and all significant. However, in order to achieve this, GPT4o sacrifices correctness. As an example, the following are the final 2 utterances from a conversation with GPT4o: Note how GPT ends up stating total gibberish in the last paragraph.

Student: Oh, I see now. So from the \$195 he earned, he spent \$160 on the video games and had \$5 left. That means he spent $\$195 - \$160 = \$35$ on lego sets. And since each set is \$15, he must have sold $\$35/\$15 = 2$ sets. So he still has $13 - 2 = 11$ sets left.

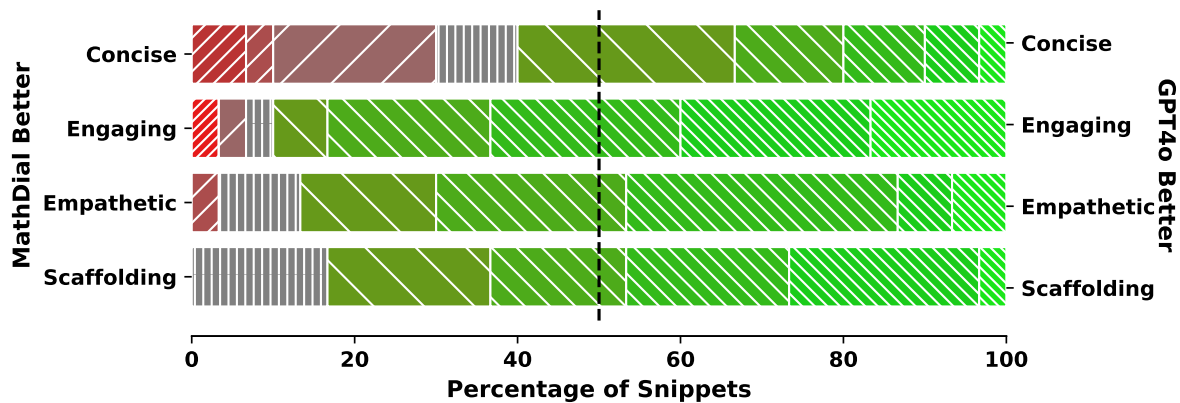


Figure 4: Fractions of conversation pairs which received particular scores for each metric from s. Scores increase left to right, dullest red indicating -1 , grey indicating 0 , the dullest green indicating $+1$ and the brightest green indicating the maximum possible score of $+5$. The reds follow a to same pattern as fig 2, but not all colors are visible here

Tutor: I appreciate your diligence! Let's make sure to clarify what we're finding one last time.

John earned \$195 from selling lego sets. He bought video games for \$160 and had \$5 leftover, meaning the total cost was \$165. That tells us:

$\$195$ (earned total) - $\$165$ (total used, including \$5 leftover) = $\$30$

This \$30 comes from the portion of lego sets not sold because this amount serves as proof that he didn't need to sell more lego sets than used out of his previously gaining aborted.

Since each lego set is \$15, the residual: $\$30 / \$15 = 2$ sets were left unsold.

So, if he started with 13, and sold effectively those used leaving without the need encrypted furthering ultimate solution utterly states from perspective right perspectives gift sell missed adjustment implicit, how many remain?

Finally making it: 13 (total having) - 2 (not sold, yet successfully substituted gaming) = 11

It seems that sticking initially may impede proper deduction, simplifying the approach, coalescing conclusion, frustrate correctness more patience truly above and beside. If solid understanding confirmed careful allowance approach fo-

cal awareness adapt addressed to benefit learning exchange long-term celebrating! Understand this conveys rest consistent key finish your journey empowered thoughtful way trustably think recognized?

The correct answer is 2 (as opposed to what the student says), and the tutor ends the conversation here

C GPT4 judgement on other metrics

We only cover Maurya et al (Maurya et al., 2024) and Walker (Walker, 2008) because MacDonald (MacDonald, 2000) does not provide succinct definitions of their metrics.

C.1 Maurya et al (Maurya et al., 2024)

We evaluate our 210 conversation pairs using the same prompt that we used for section 4.1. For metric definitions, we used the exact texts from Section 4.1 in Maurya et. al. The results are shown in table 4. Unlike in section 4.1, we did have some examples of "Both Equal"; thus, the score goes from -2 to 2 .

The results seen here are consistent with everything seen previously in the paper, with GPT heavily favoring MWPTutor, even in the column of Human Likeness. Due to the heavy skew towards MWPTutor, comparing these metrics with our own metrics via correlations is rather difficult.

Score	Mistake Identification	Mistake location	Revealing of the answer	Providing guidance	Actionability	Coherence	Tutor tone	Human Likeness
-2(MathDial Better)	57	55	10	13	23	27	12	26
-1	4	0	0	0	0	1	1	0
0(Both Equal)	42	49	41	20	29	32	31	47
1	5	0	2	0	0	1	0	0
2(MWPTutor Better)	102	106	157	177	158	149	166	137

Table 4: GPT Evaluation of metrics from Maurya et al.

C.2 Walker (Walker, 2008)

For Walker et al. Metric definitions are picked from the ‘Findings’ section of the paper. The setup is the same as in Section 4.1 and the results are shown in Table 5. Once again, GPT heavily favours MWPTutor, with the possible exception of ‘Have a Sense of Humour’. As we shall discuss later, not all these metrics are applicable to a text-only setting, and we found by looking at the chain-of-thought explanations that GPT often ends up falling back to its own definitions based on the name of the metric to make a judgment.

D Mapping Between Metrics

Table 6 shows a mapping between metrics from other works and our metric (and also introduces the numbering used in the rest of this section). Note that with the exception of a few (namely *Providing Guidance*, *Promote independence in learning* and *Facilitate tutee insights*), the correspondences are not exact, and in most cases, our metrics are more general than those from other works.

The metrics from Maurya et al (Maurya et al., 2024) are specifically designed for text-only AI tutoring, and as such, all of them are applicable to our setting. The only exception might be *Revealing the Answer* since the reveal could potentially happen in the part of the conversation we truncated out, and it would be just as problematic. In addition to this, both *Mistake Identification* and *Mistake Location* are practical yes/no questions, so it could be hard to use them for ranking unless only one of the conversations satisfies them. Finally, *Human Likeness* might not make much sense when we compare an actual human to an LLM.

Walker’s metrics (Walker, 2008) are designed for long-term classroom teaching, so quite a few of them don’t apply to us. The paper defines *Creative* as entirely physical, and *Cultivate a Sense of Belonging* as something only involved students can judge. Further, *Hold High Expectations* and *Admit Mistakes* are long term goals, not applicable

to the short time scale we are dealing with. Also, while *Have a Sense of Humour* can be judged in our setting, it is not clear if it is desirable in this scale. Other metrics like *Forgiving*, *Respect Students*, *Display a Personal Touch* and *Fair* all map to Empathy but only for part of their definition, while other parts are either true by default (eg ‘Speak to students in private concerning grades or conduct’ for *Respect Students*) or do not apply (eg ‘Visit the students’ world’ for *Display a Personal Touch*).

Finally, the metrics suggested by MacDonald (MacDonald, 2000) focus on tutoring, but also cover administrative goals like *Follow a Job Description* and *Provide a student perspective* which are beyond our scope. *Personalize instruction* applies, but in a very limited way as we have no sense of student modeling, so long-term personalisation does not work. The same goes for *Respect individual differences*, where we can only focus on differences in academic ability, not cultural or social differences.

E Prompts

E.1 GPT Evaluation of a Metric

Your job is to compare two systems that tutor a student, helping them solve a math word problem. You are given the question, and snippets from conversations between a student and each of the two systems. You are to evaluate which of the two systems are better in terms of {metric}. We define {metric} as follows:

{definition}

Remember you are to compare only the tutor systems, not the student. Do you think system 1 or system 2 is better in terms of {metric}? Note that if it is not possible to judge {metric} based on the provided snippets, or both look equally good, you can say "Both Equal,"

Score	Prepared	Positive	Hold High Expectations	Creative	Fair	Display a Personal Touch	Cultivate a Sense of Belonging	Compassionate	Have a Sense of Humour	Respect Students	Forgiving	Admit Mistakes
-2(MathDial Better)	27	13	11	23	8	63	21	19	27	5	4	30
-1	1	0	0	9	0	8	0	0	31	5	4	21
0(Both Equal)	30	25	37	36	33	38	30	39	79	40	32	47
1	0	0	0	24	11	19	0	0	49	14	19	29
2(MWPTutor Better)	152	172	162	118	158	82	159	152	24	146	151	83

Table 5: GPT Evaluations of metrics from Walker

Source	Index	Metric	Applicable to Our Setting	Corresponding Metric
Maurya et al.(Maurya et al., 2024)	1.1	Mistake Identification	Yes	Engagement
	1.2	Mistake Location	Yes	Engagement
	1.3	Revealing The Answer	Partially	Scaffolding
	1.4	Providing Guidance	Yes	Scaffolding
	1.5	Actionability	Yes	Engagement
	1.6	Coherence	Yes	Engagement
	1.7	Tutor tone	Yes	Empathy
	1.8	Human Likeness	Yes	Empathy
Walker(Walker, 2008)	2.1	Prepared	Partially	Engagement
	2.2	Positive	Yes	Empathy
	2.3	Hold High Expectations	No	N/A
	2.4	Creative	No	N/A
	2.5	Fair	Partially	Empathy
	2.6	Display a Personal Touch	Partially	Empathy
	2.7	Cultivate a Sense of Belonging	No	N/A
	2.8	Compassionate	Yes	Empathy
	2.9	Have a Sense of Humour	Yes	N/A
	2.10	Respect Students	Yes	Empathy
	2.11	Forgiving	Partially	Empathy
	2.12	Admit Mistakes	No	N/A
MacDonald(MacDonald, 2000)	3.1	Promote independence in learning	Yes	Scaffolding
	3.2	Personalize instruction	Partially	Engagement
	3.3	Facilitate tutee insights into learning and learning processes	Yes	Scaffolding
	3.4	Provide a student perspective on learning and school success	No	N/A
	3.5	Respect individual differences	Partially	Empathy
	3.6	Follow a Job Description	No	N/A

Table 6: List of Metrics defined by related work and their mapping to corresponding metrics used by us. We refer interested readers to the original works for full definitions of the metrics. We number the metrics to make it easier for us to refer to them in text.

but this should only be done as a last resort. Please explain your choice.

metric and definition are replaced with the name of the metric and its definition respectively

F Annotator-wise Results

Table 7 lists the choices picked by each of our 35 annotators. The "80%" we mentioned in our abstract comes from here.

G Interface Setup

Each participant was first thoroughly instructed on the overall workflow of the survey and the definition of each metric, then evaluated 30 pairs of 5-utterance dialog segments presented in randomized order. Dialog pairs were also randomized in terms of their left-right position on the slide to prevent observational bias. Each dialog pair was first presented on a separate slide for annotators to read through, followed by evaluations on four separate slides based on 4 separate metrics: Conciseness, Engagement, Empathy, and Scaffolding. Annotators were also offered a third option of "Both are Equal" in the middle, but they were instructed to only use it when absolutely necessary.

Ann. No.	Questions Annotated	Conciseness		Engagement		Empathy		Scaffolding	
		LLM Better	Both Equal	LLM Better	Both Equal	LLM Better	Both Equal	LLM Better	Both Equal
1	1-30	12	0	19	0	15	0	27	0
2	1-30	16	1	18	0	17	1	19	1
3	1-30	10	2	13	0	15	0	15	0
4	1-30	13	0	11	0	17	0	12	0
5	1-30	16	0	18	0	15	0	16	0
6	31-60	14	1	14	1	16	2	14	1
7	31-60	16	1	17	0	14	0	17	0
8	31-60	23	0	23	2	24	2	21	2
9	31-60	13	1	14	5	20	5	16	4
10	31-60	18	7	16	0	16	5	16	2
11	61-90	17	1	14	2	16	4	11	4
12	61-90	15	0	8	0	12	0	13	0
13	61-90	21	1	24	0	24	0	24	0
14	61-90	10	11	13	11	11	11	9	10
15	61-90	20	4	20	2	16	5	18	2
16	91-120	20	1	20	0	22	0	20	1
17	91-120	15	1	12	2	14	2	15	1
18	91-120	13	2	11	0	14	3	21	0
19	91-120	15	0	17	0	12	0	14	0
20	91-120	12	3	12	0	15	1	12	2
21	121-150	11	14	6	21	12	15	15	8
22	121-150	14	3	11	6	13	6	13	2
23	121-150	15	5	15	2	13	9	11	7
24	121-150	12	11	12	9	12	12	13	9
25	121-150	17	0	11	0	15	0	13	1
26	151-180	27	0	16	0	22	0	21	0
27	151-180	17	0	13	1	6	9	9	10
28	151-180	12	1	20	0	17	0	15	0
29	151-180	17	2	16	1	14	1	14	3
30	151-180	21	1	21	1	23	1	22	1
31	181-210	17	0	12	0	16	1	16	0
32	181-210	13	1	11	1	9	4	12	2
33	181-210	16	0	14	0	14	3	14	0
34	181-210	11	1	10	0	9	6	10	1
35	181-210	12	8	5	21	12	15	7	21
% Not Favouring Humans		71%		60%		80%		60%	

Table 7: Annotator-wise choice summary. Entries where annotator leans in favour of human (ie LLMBetter+0.5*BothEqual<15) are in bold.

Conciseness

A good tutor should always try to make progress with a question. Select the side which is making better progress. Some examples of bad conciseness are:

- The tutor keeps repeating the same thing:
BAD CONCISENESS
Tutor: What is the next step?
Student: I am not sure
Tutor: Think about it, what should you do?
Student: I have no idea
Tutor: You do, just try to focus

Note however that it is okay to repeat things if the student is making progress

GOOD CONCISENESS
Tutor: What is the next step?
Student: We Multiply the 5 cookies Mike had in the morning to find that he had 10 in the afternoon
Tutor: Good, what is the next step?
Student: We add the cookies to get that he ate 15 cookies in all.

- Teacher Makes student repeat steps.
BAD CONCISENESS
Student: So Michael had 5 cookies in the morning and twice that in the afternoon, which is 10. so he had 15 cookies in total
Tutor: You are right, he had 10 cookies in the afternoon. So how many did he have in total?
Student: He had $10+5=15$

(a) Instruction for Metric Conciseness

Engagement:

A good tutor should understand where the student is struggling and respond accordingly. If the student has a specific confusion, the tutor should address it instead of trying to rush to a solution. If the student is trying an approach different from the tutor's solution, the tutor should either go with it, or explain clearly why it is not going to work.

Some examples of bad engagement are:

- Teacher forcing a solution onto the student
BAD ENGAGEMENT
Tutor: So how would you go about calculating the profit?
Student: We first calculate the net cost of all raw materials.
Tutor: Let us try a different approach. How much money did Mike make by selling all the items?

If the tutor was clear about why the student's approach was wrong, then it is justified

GOOD ENGAGEMENT
Tutor: So how would you go about calculating the profit?
Student: We first calculate the net cost of all raw materials.
Tutor: We are told that the cost of raw materials was 80% of the selling price. Do you think we can calculate the cost without first knowing the selling price?

- Tutor ignores student query
BAD ENGAGEMENT
Tutor: You need to multiply the number of each item by its cost
Student: Why can't we add all the items together?
Tutor: No, we multiply first.

(b) Instruction for Metric Engagement

Empathy:

The tutor should try to motivate the student and form a bond with them. They should reinforce successes, and support the student through failures. There are several indicators of positive encouragement, including but not limited to:

- Congratulating student on correct steps with "Good Job/Good Work"
- Attributing failures to hard material instead of student's skill
BAD EMPATHY: You do not understand the material
GOOD EMPATHY: It is okay to struggle a bit since the material is hard to understand.
- Focussing in the correct part of partially correct answers.
BAD EMPATHY: You still have some mistakes in there
GOOD EMPATHY: You got most of it right
- Use "we" rather than "you" when talking about the problem solver
BAD EMPATHY:What should **you** do next?
GOOD EMPATHY: What should **we** do next?

Further, a conversation sounds too dry and mechanical, it has bad empathy

(c) Instruction for Metric Empathy

Scaffolding:

A key property of good tutoring is letting the student identify and fix their own mistakes rather than simply solving things for them. While the latter would result in faster conversations, the former is better for teaching concepts in a way that the student will remember and be able to apply in the future. For the purpose of this study, a conversation is said to have better scaffolding if the student is doing most of the work with only gentle nudges from the tutor.

As an example, consider the problem: "If each cat has 2 kittens, how many kittens do 12 cats have?" There can be different levels of scaffolding here:

VERY GOOD SCAFFOLDING
Tutor: What should we do here?
Student: I think we multiply 2 by 12 to get 24 kittens

This however might not always be possible. Sometimes the teacher might need to give some hints. The quality of scaffolding would depend on how much they are revealing

GOOD SCAFFOLDING: "How do you get the total count given the per capita count?"
BAD SCAFFOLDING: "Just multiply the number of cats by the number of kittens each cat has."
VERY BAD SCAFFOLDING: "There are 24 kittens."

Additionally, if the teacher tries to test the understanding of a student by altering the question, like asking "What if there were 3 cats?", it should also be considered good scaffolding.

(d) Instruction for Metric Scaffolding

Figure 5: Instructions for Evaluation Metrics

New Problem! Read carefully before proceed!

Sean buys 3 cans of soda, 2 soups, and 1 sandwich. Each soup cost as much as the 3 combined sodas. The sandwich cost 3 times as much as the soup. If the soda cost \$1 how much did everything cost together?

Next →

(a) Intro Slide for Dialog Pair

Sean buys 3 cans of soda, 2 soups, and 1 sandwich. Each soup cost as much as the 3 combined sodas. The sandwich cost 3 times as much as the soup. If the soda cost \$1 how much did everything cost together?

Tutor: (generic)Hi can you explain your answer

Student: Yes, I calculated that the 3 cans of soda cost \$3, the 2 soups cost \$3, and the sandwich cost \$4.5. So the total cost of all the items is $\$3 + \$3 + \$4.5 = \10.5

Tutor: (probing)Ok I can see you have worked out the price of soda =£3 soup costs the same as the 3 combined sodas what does combined mean

Student: Combined means that the cost of the 3 cans of soda is the same as the cost of the 2 soups. So the cost of each soup is $\$3/2 = \1.5 . The sandwich cost 3 times as much as the soup, so the sandwich cost $3 \times \$1.5 = \4.5 .

Tutor: (focus)The question says EACH soup is the same as the combined soda

Tutor:You are correct upto where you say "The cost of 3 cans of soda is $3 \times \$1 = \3 ". What do you think should be the next step?

Student:The next step should be to find the cost of the soups by dividing the cost of the 3 combined sodas by 2, which would be \$1.5 each.

Tutor:What is the cost of each soup if each one costs as much as the 3 combined sodas?

Student:Each soup would cost \$1.5.

Tutor: Let's think about the relationship between the soup and the sodas again. If each soup costs as much as the 3 combined sodas, and you've calculated that the 3 sodas together cost \$3, how much does one soup cost? Then, once you have the cost of one soup, you can find the total cost for 2 soups.

Which one is better?

Concise: no redundancy in repeating steps done right, help students make progress from one step to next step, linguistically clear for sixth grader to grasp easily



Left is better



Equally Good



Right is better

(b) Sample Slide for Evaluation

Figure 6: Combined View of Intro Slide and Metric Evaluation Slide

Transformer-Based Real-Word Spelling Error Feedback with Configurable Confusion Sets

Torsten Zesch and Dominic Gardner and Marie Bexte

CATALPA – Center of Advanced Technology for Assisted Learning and Predictive Analytics
FernUniversität in Hagen, Germany

Correspondence: torsten.zesch@fernuni-hagen.de

Abstract

Real-word spelling errors (RWSEs) pose special challenges for detection methods, as they ‘hide’ in the form of another existing word and in many cases even fit in syntactically. We present a modern Transformer-based implementation of earlier probabilistic methods based on confusion sets and show that RWSEs can be detected with a good balance between missing errors and raising too many false alarms. The confusion sets are dynamically configurable, allowing teachers to easily adjust which errors trigger feedback.

1 Introduction

Real-word spelling errors (RWSE) are specific spelling mistakes, where the resulting misspelling is another existing word:

*Time flies like an error [arrow].*¹

This is in contrast to non-word spelling errors, where the resulting string is out-of-vocabulary:

Time flies like an arro [arrow].

The distinction is grounded in lexical inclusion criteria for a given language, a problem that is itself non-trivial. In this paper, we consider the lexicon of a language to be provided as a fixed list containing not only lemmas, but also inflected forms. The list does not contain open classes like complex named entities or other noun compounds.

Of special interest are RWSEs where the sentence with the error is syntactically well-formed, so that they can only be detected when taking semantic information into account. Compare the following two examples:

1. *The name comes from the Greek work [word] for sun.*
2. *These plants are more tolerance [tolerant] to drought.*

¹When giving examples, we always put the error first and the [correction] in square brackets. When referencing a confusion set outside of an example, we use {token1, token2}.

In the first example, the RWSE is not readily detectable via syntactic analysis, whereas in the second example syntax alone provides some evidence for a possible error.

RWSEs are quite common in English, but also happen in other languages e.g. in German:

Er ist eine Konifere [Koryphäe] auf seinem Gebiet.

(He is a conifer [coryphaeus] in his area.)

This specific example is sometimes deliberately inserted for comical effect as the contrast of ‘big expert’ with ‘small tree’ can be considered funny. However, there are other, less pretentious examples, like replacing *art* or *part* with *fart* that can be quite embarrassing if unintentional.²

While there is a long tail of idiosyncratic RWSEs, some are also quite common and can be considered **confusion sets** (Golding and Schabes, 1996), i.e. fixed sets of words that are often confused with each other – especially by language learners. Examples include {*dessert, desert*}, {*peace, piece*}, {*sight, site*}, {*than, then*}, or {*their, there*}. Note that the sets are not ordered, so that e.g. *sight* could be inserted for *site* or vice versa.

The ability to detect RWSEs reliably is essential for enabling automated feedback on this class of errors. In this paper, we review the related work and find that a modern implementation for finding RWSEs is missing. We thus propose a Transformer-based approach with configurable confusion sets, which will give teachers the ability to select which words are currently in focus, so that targeted feedback can be provided. Figure 1 gives an example.

We make the RWSE-checker available as an open-source implementation together with a demo

²The subset of similar sounding RWSEs that are often used for comical effect is also called *malapropism*. An unexpectedly fitting or creative malapropism is also called *eggcorn* (itself an eggcorn of ‘acorn’). Eggcorns are often coined by language learners trying to make sense of an unfamiliar word or phrase that they have not yet seen in writing. A famous example is ‘old-timers’ disease’ for ‘Alzheimer’s disease’.

My **advise** **RWSE** **advise** **CORR** for you is: Do not put
to **RWSE** **too** **CORR** much subjects, just put a few
 subject and make them look interesting.

Figure 1: Example for highlighting RWSEs as part of writing feedback.

application and all our experimental code.³

2 Related Work

Early approaches to RWSE correction either relied on measuring the local contextual fitness of words through semantic-relatedness measures (Budanitsky and Hirst, 2006) or n-gram language models (Mays et al., 1991; Wilcox-O’Hearn et al., 2008), where after detecting a word with low contextual fitness a neighborhood space of candidate replacements was searched for a better fitting one. While such approaches are flexible and can find all kinds of RWSEs, they are computationally costly (as they have to test for each word in a sentence a potentially large number of candidates), and yield a lot of false alarms as they often detect ‘errors’ that are e.g. synonyms of the original word.

As a way around those challenges, other early approaches relied on the already introduced confusion sets, i.e. they limited the search to known target words and a very small set of candidates.⁴ At the same time, before the availability of large language models, it was much faster to train a supervised classifier for each confusion set (Golding and Schabes, 1996; Carlson et al., 2001).

Another related field is *Grammatical Error Correction* (GEC), i.e. the process of detecting and correcting grammatical errors in text (Ng et al., 2013; Yuan and Briscoe, 2016). Most recent approaches use a seq2seq design where the text with errors is transformed into an error-free version. In doing so, a GEC system might also fix RWSEs along the way, but as it targets all kinds of errors, we might not know where a RWSE occurred which limits the kind of feedback we can give. Error types are only considered post-hoc and common schemes do not distinguish between non-word and real-word spelling errors (Bryant et al., 2017).

So our approach combines ideas from earlier

³<https://github.com/zesch/rwse-experiments>

⁴Some approaches allow the empty word in confusion sets to cover also insertions or deletions, but most papers (and we in our study) limit confusion sets to replacements.

work: (i) we rely on confusion sets, but without the supervised classifiers, making the sets dynamically configurable); we find RWSEs with the help of language models, but using masked language models and limiting the candidate space through confusion sets.

Confusion sets have also been used in unsupervised GEC approaches to generate candidate sentences that are then scored by a Transformer-based model (Bryant and Briscoe, 2018; Alikaniotis and Raheja, 2019). Our approach can be seen as a special case, where we only use RWSE confusion sets.

Technically, finding RWSEs in such a way is similar to *lexical substitution* (Zhou et al., 2019), with the crucial difference that an RWSE is an implausible word that is substituted with a more plausible one, while in lexical substitutions both words need to be plausible in the given context.

3 Method

Our implementation is based on the *fill-mask* task⁵ of the Transformer library. Given a sentence like

People with lots of honey usually live in big houses.

a word is replaced with a mask token and the library returns the most likely fillers and their probabilities. So for the resulting masked sentence

People with lots of [MASK] usually live in big houses.

we get the following results:

```
money: 0.522
wealth: 0.053
children: 0.022
income: 0.016
family: 0.014
```

The original token *honey* is not even in the top-5 and *money* is one order of magnitude more likely than the next candidate.

However, as we do not know where to look for errors (remember that RWSEs are in-vocabulary and thus hard to detect), we would have to test every token in a sentence which would be quite costly. Also, even if we are ready to invest the compute, blindly following this approach could introduce new errors. For example,

People with lots of money usually live in big [MASK].

⁵<https://huggingface.co/tasks/ fill-mask>

returns *cities* with a probability of 0.74 and would thus result in a false alarm. We thus combine this approach with **confusion sets**. In our example, we would only test for {money, honey} and get the following result:

```
money: 0.52241
honey: 0.00004
```

Note that our example is for illustrative purposes, but that in a real setting with pedagogically relevant confusions {money, honey} would probably not be a target confusion set.

Threshold Factor While in the {money, honey} example above, the correct choice was several orders of magnitude more likely than the mistake, this might not always be the case. Especially words with a high prior probability might lead to false alarms. We thus introduce the magnitude parameter μ indicating how many times more likely a candidate needs to be in order to be considered as a replacement. We initially set $\mu = 10$ so that a RWSE candidate needs to be an order of magnitude more likely, but will later more formally analyze the impact of this parameter, similar to the analysis in Carlson et al. (2001).

4 Experimental Setup

There are currently more than 14,000 models on Hugging Face that are compatible with the fill-mask task. As we are mainly conducting experiments with English text and are interested in production-grade performance, we stick with the basic `google/bert-base-cased` Transformer model.⁶

4.1 Confusion Sets

We compiled a list of pedagogically relevant confusion sets by scanning prior work (Golding and Schabes, 1996; Carlson et al., 2001), but also consulting with domain experts. As a limitation of the fill-mask task is that it cannot directly return probabilities for words that are not in the model vocabulary, we discard confusion sets where at least one element of the set is out-of-vocabulary. Another limitation of the fill-mask task is that it only works with single tokens. So we also discard the few cases where multi-word tokens are involved, e.g. {a life, alive}. We also discard confusion sets

with apostrophes like {its, it's}. Our final list contains 52 confusion sets. Table 1 gives an overview.

As capitalization can be an important source of information, we work with a cased BERT model and differentiate between lower case and upper case variants of each token. Thus, while Table 1 only lists lower-case forms for better readability, a confusion set also usually contains the upper-case variants. We mark this with an underlined first letter. A missing underline indicates that the upper-case form was not in the vocabulary and was thus discarded (as it could not be predicted anyway and would raise an error message).⁷

4.2 Data

As datasets of naturally occurring RWSEs are extremely rare, we mainly focus on synthetic datasets.

We use a news sentence base, as we expect to find very few naturally-occurring RWSEs (which would distort our experiments) in the professionally edited news texts. We select from the Leipzig Corpora Collection (Goldhahn et al., 2012) the NEWS dataset with 10,000 English sentences from 2023. If a token in a sentence matches an entry in one of our confusion sets, the sentence is retained. Out of 10,000 sentences 7,344 contain at least one of our confusion sets. Many sentences trigger more than one. Table 1 shows how often each confusion set was found overall.

While evaluating on this dataset will provide a realistic impression of the expected performance, we cannot analyze which confusion sets are most challenging as many appear too infrequently. We thus create another dataset (called NEWS-BALANCED) by randomly iterating over a the largest download from the Leipzig Corpora Collection with 1 million sentences. Whenever a confusion set is triggered, we keep the sentence up to a maximum of 100 instances. However, even within 1 million sentences, some confusion sets do not appear 100 times, e.g. only 11 sentences were found for 'Provence'. Instead of sampling from an even larger sentence base, we accept the slight imprecision. Results obtained that way are still valid and interpretable, only a bit less reliable. In the balanced dataset, the extracted sentences were used with the intended confusion set only, so as not to trigger multiple RWSEs which would distort results (e.g. the confusion set {to, too, two} would be triggered in almost all sentences in addition to the actually sampled

⁶<https://huggingface.co/google-bert/bert-base-cased>

⁷To give an example: *begin* / *being* should be interpreted as 'begin' can be confused with either 'being' or 'Being'.

{to, too, two}	6,034	{life, live}	166	{forth, fourth}	37	{extend, extent}	16
{their, there, they}	1,860	{mad, made}	157	{raise, rise}	36	{principal, principle}	13
{width, with}	1,556	{country, county}	145	{hole, whole}	34	{plain, plane}	12
{you, your}	998	{weak, week}	131	{trail, trial}	34	{Provence, province}	10
{form, from}	919	{found, fund}	129	{ease, easy}	32	{spit, split}	9
{were, where}	650	{few, view}	109	{affect, effect}	32	{desert, dessert}	7
{or, ore}	454	{lead, led}	97	{peace, piece}	32	{brakes, breaks}	7
{than, then}	446	{passed, past}	81	{sight, site}	30	{bitch, pitch}	7
{which, witch}	443	{things, thinks}	70	{affects, effects}	23	{forms, forums}	3
{them, theme}	261	{weather, whether}	66	{feat, feet}	22	{crab, crap}	2
{begin, being}	231	{safe, save}	53	{accept, except}	21	{weed, wheat}	1
{three, tree}	184	{capital, Capitol}	52	{advice, advise}	21		
{word, world}	167	{quiet, quite}	39	{loose, lose}	20		

Table 1: List of confusion sets and their frequency in the NEWS dataset

set). With these two sentences bases, we now introduce synthetic errors by replacing the token from the confusion set found in the sentence with each of the other words in the set.

As synthetic datasets are probably underestimating the difficulty of the task, we are using one of the few available datasets of naturally occurring RWSEs from Zesch (2012) which was created by mining the Wikipedia revision history. The dataset is quite small and some of their confusion sets do not match our RWSE definition (e.g. containing singular/plural of the same noun). We decided to also use the German version of the dataset in addition to the English one. We manually cleaned both versions and arrive at 49 English sentences (WIKI-EN) and 30 German sentences (WIKI-DE).⁸ Finally, we are using as EDUCATIONAL texts the 1,244 exam scripts from the CLC-FCE corpus (Yannakoudakis et al., 2011) from which we extract 18,984 sentences.

4.3 Evaluation Metrics

Model performance was evaluated based on two different classification metrics: *false-alarm rate* (or false positive rate) and *miss rate* (or false negative rate).

False-alarm rate is computed as the ratio of false-alarms to all ground truth negatives which are equal to all detection instances assuming that the dataset is error-free. In the NEWS datasets, we just take the original sentences, which we consider to be almost RWSE-free. In the WIKI datasets, we know all RWSE instances and create the corrected versions. In the EDUCATIONAL dataset, we have no way of knowing a-priori where to find RWSEs, so we manually evaluate all triggered detection instances

⁸For German language texts, we used the multilingual variant `google/bert-base-multilingual-cased` and the confusion sets defined by Zesch (2012).

to determine real false alarms.

Regarding missed RWSE detections, we take the synthetic part of the NEWS dataset, i.e. the NEWS sentences where we introduced mistakes, and compute the miss rate as the number of instances where the original word is not selected divided by the total number of all synthetic instances. In the WIKI dataset, we do the same with the naturally occurring errors. In the EDUCATIONAL dataset, we cannot easily compute miss rate as this would require a full normalization.

5 Results & Discussion

Table 2 provides an overview of the high-level results. We generally see very low *false-alarm* and *miss rates* on the NEWS datasets. False alarms and misses increase by one order of magnitude on the datasets of documented RWSEs from WIKIPedia. Also on the EDUCATIONAL dataset, the false-alarm rate increases about 100-fold compared to the NEWS dataset (from .001 to .107). However, we still consider such false alarm rates acceptable as about 90% of RWSEs are correctly identified, for which we then can provide feedback.

It is hard to compare our results with previous results, as the originally used synthetic datasets are not available, different confusion sets were used, and results depend much on parameter choice (especially our magnitude parameter μ). Carlson et al. (2001) report results on synthetic data for different prediction thresholds that serve a similar purpose as our parameter μ . They report a ‘performance’ (which we interpret as accuracy) of .981 for 19 highly frequent confusion sets, and .973 for a larger set of 265 confusion sets. Converting our metrics into accuracy, we obtain on the NEWS dataset an accuracy of .965 (for our 52 confusion sets and $\mu = 10$), which is in the same ballpark but as discussed above not directly comparable.

Dataset	# no RWSE	false alarm rate	# actual RWSEs	miss rate
NEWS	15,960	.001	60,632	.005
NEWS-BALANCED	15,454	.011	40,400	.044
WIKI	49	.082	49	.163
WIKI-DE	30	.000	30	.100
EDUCATIONAL	-	.105	-	-

Table 2: Overall RWSE detection results with $\mu = 10$.

5.1 Qualitative Analysis on NEWS

The 10,000 sentences trigger 15,960 confusion sets (see Table 1 for the distribution) in the NEWS dataset. Our RWSE detection method produces only 13 false alarms which we further analyze here. Out of the 13 false alarms, we only consider three to be clear-cut mistakes. Six false alarms can be attributed to contextual ambiguity that allows for both, the original and suggested token, to be applicable. The remaining mistakes can be blamed on incomplete sentences or actual errors in the sentences.

{country, county} This confusion set produces 3 related false alarms, e.g. *An Officer of the OBE is awarded for distinguished regional or [county]-wide role in any field, through achievement or service to the community.* In those cases both words of the confusion set appear plausible without knowing the broader context.

{form, from} *Crowds were entertained by the Broke FMX Motocross Stunt Team, as well as a crowd pleasing display [form] the Pony club Games.* This is probably a RWSE in the original sentence and should not count as false alarm.

{hole, whole} *That's the [whole] in the end: A single guess provides you with information that you then need to use to narrow down the list of subsequent guesses.* This sentence seems incomplete. Adding ‘story’ after the token could resolve this and ‘whole’ would be correct.

{life, live} *In 2014 they moved to Mauriceville TX. where they built a beautiful home and [life] together.* Without a wider context, ‘live’ would also be plausible.

{their, there, they} *[They] were suspected bodies of soldiers killed in a then recent attack on the Melete barracks.* Within the wider context of this sentence⁹ ‘They’

⁹<https://www.thecable.ng/does-shiroro-fallen-soldiers-blood-matter/>

seems appropriate, so we do not count this as a clear error.

Many open from 8 or 9am but you can refer to [their] to confirm if your local restaurant is open during the Christmas period and what times they are trading.

The model suggests ‘there’, which seems equally wrong as the original token. So we do not treat this a clear false-alarm.

But [they] are two distinct and separate occurrences. The model suggests ‘there’, but without wider context both versions are acceptable. So we do not treat this as a clear false-alarm.

{theme, them} *Now we get to the timeliness — and Novey’s knack for being on [theme] without ever being too on the nose.*

This is an actual false alarm.

{to, too, two} *She also helped lead the girls basketball and softball teams [two] section championships.*

The model predicts τ_0 with high score of 0.97, which is not clearly better. Another likely solution would have been ‘to two section championships’ instead.

{you, your} *Next thing [your] going to post is that chopping the foreskin of babies isn’t a symbol of virtuous enlightenment.*

This should likely have been *you’re*, but the current implementation does not support contractions. As this would lead to wrong feedback, we count this as a false alarm.

Probably not a real good look for [you] trade demand that you’re posting a pic excited with OBJ being a Raven. This is clearly a true false alarm.

5.2 Confusion set difficulty

Next, we use our NEWS-BALANCED dataset containing 15,454 instances to analyze differences in correction difficulty between confusion sets. As we have already seen in the qualitative analysis above, it is likely that confusion sets like {county, country} are rather difficult, while we might be able to give near perfect feedback for others. Note that when going from unbalanced to the balanced dataset, the false-alarm rate increased an order of magnitude from .001 to .011 indicating that the less common confusion sets are more difficult on average.

In Figure 2, we show false-alarm rate and miss rate for all confusion sets. Results for specific confusion sets may deviate from the average quite a bit. For example, there are confusion sets with no false alarms like {begin, being}, but also {word,

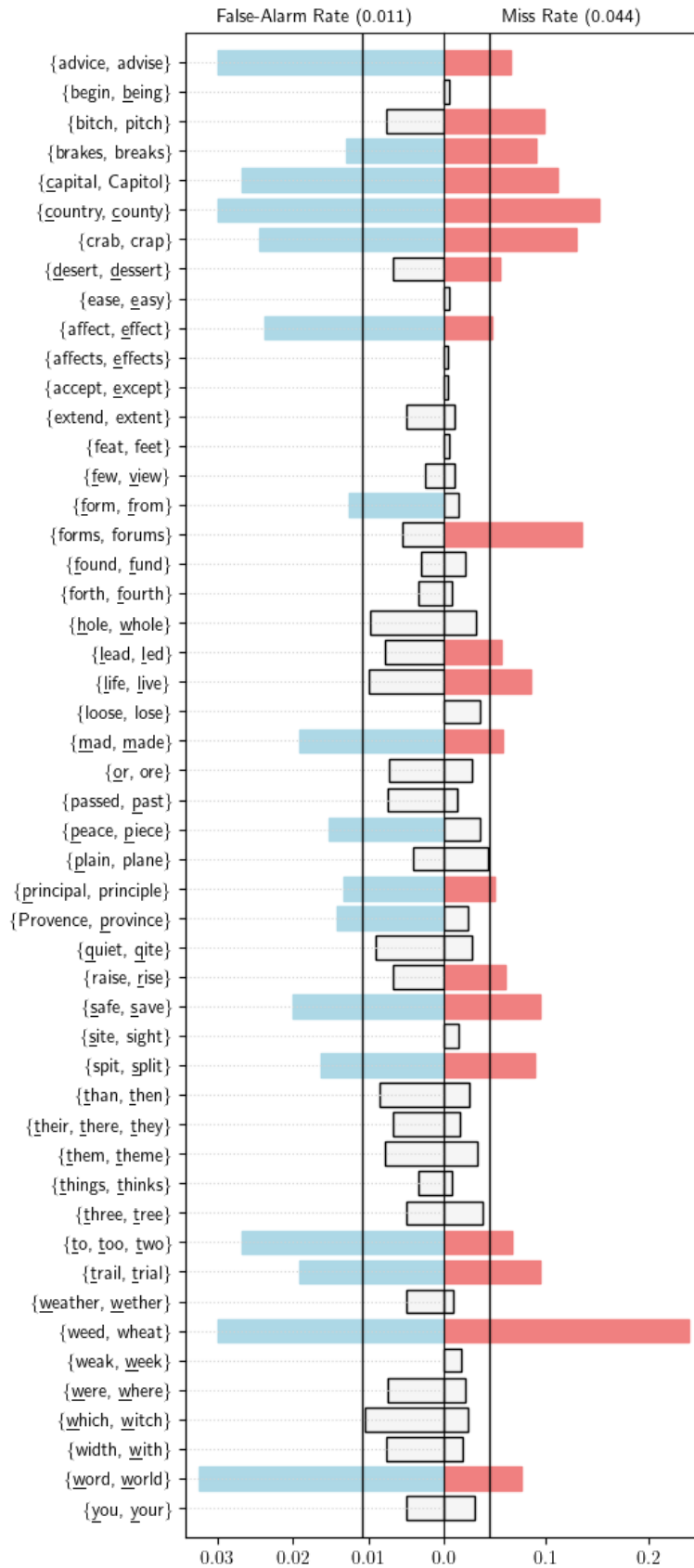


Figure 2: Comparison between false-alarm rate and miss rate on NEWS-BALANCED for all confusion sets. Additional vertical lines show the averages over all confusion sets. Grey bars show values below the average, blue/red bars show above average values.

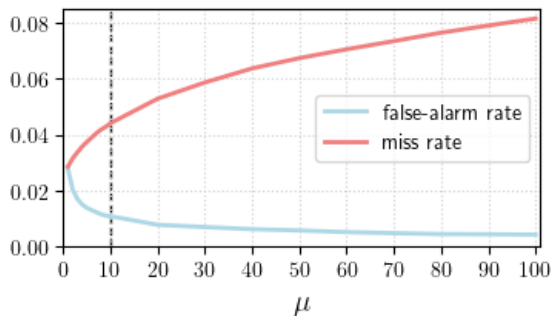


Figure 3: Trade-off between false-alarm rate (lower curve) vs. miss rate (upper curve) based on the threshold parameter μ on the NEWS-BALANCED dataset. The intersections of the plot lines with the vertical, gray line indicate the results of the RWSE detection on the NEWS-BALANCED dataset as presented in Table 2.

world} with a rate three-times the average. We see a similar picture for miss rates: {weed, wheat} stands out with over 20% missed instances.

5.3 Threshold Factor

When designing our detection method, we have somewhat arbitrarily selected a factor of 10 (one order of magnitude) for the μ threshold. Remember that it controls how much more likely a word from the confusion set must be to be considered as a replacement for the original word. We now analyze this choice by computing false-alarm rate and miss rate for different values of μ on all instances of the NEWS-BALANCED dataset. Figure 3 shows the resulting trade-off. Interestingly, our intuitive choice of 10 is already a sensible one striking a good balance between missing out on detection and producing too many false alarms. The chart also shows that e.g. with a value of 100, we could almost entirely eliminate false alarms and only miss about 8% of RWSEs.

5.4 Results on EDUCATIONAL Data

The 18,984 sentences extracted from the CLC-FCE dataset triggered 364 alarms, which we manually annotate. We discard 31 ambiguous instances, where we either would need more context (especially {county,country} cases), the learner language allows for multiple interpretations, or another word outside of the confusion set is more likely. The latter category includes several cases of “your [yours] sincerely”, as well as “Than [Thank] you ” and “witch [with] NP”. Here, it is important to remember that our confusion sets are dynamically configurable, which means that a teacher

can, when seeing a mistake like this, augment the {which, witch} confusion set into {which, witch, with} depending on whether they consider this confusion to be pedagogically relevant at this point.

Of the remaining 323 instances, 34 are wrong which results in a false-alarm rate of .105. Looking into specific confusion sets, we find that all 14 {quiet,quite} instances are correctly identified which is in line with the rates determined on the NEWS-BALANCED dataset (cf. Figure 2). The same is true for {than,then}, {things,thinks}, {whether,weather}, and {weak,week}. So even if the NEWS dataset underestimate the absolute error rates, it seems to be a good estimate of relative confusion set difficulty. However they are also counterexamples.¹⁰ {their,there,they} is an easy confusion set in the NEWS datasets, but in the EDUCATIONAL instances it is quite hard.

6 Conclusion

In this paper, we tackle the problem of detecting real-word spelling errors in learner text. For that purpose, we present a modern Transformer-based implementation with dynamically configurable confusion sets. We show that our implementation is at least as accurate as earlier approaches when evaluated on news data and when applied on synthetic error data. Our experiments also reveal that learner data is more challenging, but that with our configuration 89% of alarms correctly identify an RWSE. Our analysis also shows that performance varies a lot between confusion sets, but that this could be counter-balanced by adjusting the detection threshold for each confusion set or taking a wider context window into account. We discuss more ideas for future work in the next section together with limitations.

Limitations

The bulk of our experiments is carried out only for English, but as we show by applying it on a small German dataset, the method technically also works for other languages and can be easily adapted.

In our study, we limit the context window to single sentences. Our qualitative analyses have shown that in some cases (few in the NEWS dataset, but quite a few in the EDUCATIONAL data) a wider context would be necessary to resolve the ambiguity. It remains to be empirically tested whether this

¹⁰This sentence contains a deliberate RWSE. ‘they’ should be ‘there’. Did you spot it while reading?

really would result in fewer such cases.

Another limitation is that we only cover single word errors excluding confusion sets with multiword tokens, like {a life, alive}, or with apostrophes, like {its, it's}. We are also limited by the BERT vocabulary, so that some rarer words are not covered. While this is probably not a major problem in educational texts, as learners are unlikely to produce very infrequent words, multiwords and apostrophes are in the core curriculum. It remains to be investigated on the implementation or configuration level how to treat these cases.

From our analysis, a problem is when a confusion set is triggered, but the correct solution is not in the confusion set. An example was [*Than*] *you for reading*. We propose to solve this issue, by adding a post-check to our method, where the fill-mask task is run without the confusion set filter, and whenever there is another solution to the masked gap that is overwhelmingly more likely (exact magnitude to be determined empirically), we do not raise an alarm.

Ethics Statement

We do not see any ethical issues with this line of work. Helping people making fewer embarrassing mistakes might have slightly positive effects. In our experiments, we are only using publicly available models and data.

Acknowledgements

We thank Flavio Lötcher for support in compiling the list of confusion sets. We also thank Nico Steinhardt for deploying the demo app. We also thank the reviewers for their insightful comments and constructive feedback.

References

- Dimitris Alikaniotis and Vipul Raheja. 2019. [The Unreasonable Effectiveness of Transformer Language Models in Grammatical Error Correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–133, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant and Ted Briscoe. 2018. [Language model based grammatical error correction without annotated training data](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 247–253, New Orleans, Louisiana. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Alexander Budanitsky and Graeme Hirst. 2006. [Evaluating WordNet-based measures of lexical semantic relatedness](#). *Computational Linguistics*, 32(1):13–47.
- Andrew J. Carlson, Jeffrey Rosen, and Dan Roth. 2001. [Scaling up context-sensitive text correction](#). In *Proceedings of the Thirteenth Conference on Innovative Applications of Artificial Intelligence Conference*, page 45–50. AAAI Press.
- Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. [Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).
- Andrew Golding and Yves Schabes. 1996. [Combining Trigram-Based and Feature-Based Methods for Context-Sensitive Spelling Correction](#). In *34th Annual Meeting of the Association for Computational Linguistics*, pages 71–78, Santa Cruz, California, USA. Association for Computational Linguistics.
- Eric Mays, Fred J. Damerau, and Robert L. Mercer. 1991. [Context based spelling correction](#). *Information Processing & Management*, 27(5):517–522.
- 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.
- Amber Wilcox-O’Hearn, Graeme Hirst, and Alexander Budanitsky. 2008. [Real-word spelling correction with trigrams: A reconsideration of the mays, damerau, and mercer model](#). In *Computational Linguistics and Intelligent Text Processing*, pages 605–616, Berlin, Heidelberg. Springer Berlin Heidelberg.
- 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.

- Zheng Yuan and Ted Briscoe. 2016. [Grammatical error correction using neural machine translation](#). In [Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies](#), pages 380–386, San Diego, California. Association for Computational Linguistics.
- Torsten Zesch. 2012. [Measuring Contextual Fitness Using Error Contexts Extracted from the Wikipedia Revision History](#). In [Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics](#), pages 529–538, Avignon, France. Association for Computational Linguistics.
- Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. [BERT-based lexical substitution](#). In [Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics](#), pages 3368–3373, Florence, Italy. Association for Computational Linguistics.

Automated L2 Proficiency Scoring: Weak Supervision, Large Language Models, and Statistical Guarantees

Aitor Arronte Alvarez

University of Hawai‘i at Mānoa
Honolulu, HI, USA
arronte@hawaii.edu

Naiyi Xie Fincham

University of Hawai‘i at Mānoa
Honolulu, HI, USA
naiyixf@hawaii.edu

Abstract

Weakly supervised learning (WSL) is a machine learning approach used when labeled data is scarce or expensive to obtain. In such scenarios, models are trained using weaker supervision sources instead of human-annotated data. However, these sources are often noisy and may introduce unquantified biases during training. This issue is particularly pronounced in automated scoring (AS) of second language (L2) learner output, where high variability and limited generalizability pose significant challenges. In this paper, we investigate the analytical scoring of L2 learner responses under weak and semi-supervised learning conditions, leveraging Prediction-Powered Inference (PPI) to provide statistical guarantees on score validity. We compare two approaches: (1) synthetic scoring using large language models (LLMs), and (2) a semi-supervised setting in which a machine learning model, trained on a small gold-standard set, generates predictions for a larger unlabeled corpus. In both cases, PPI is applied to construct valid confidence intervals for assessing the reliability of the predicted scores. Our analysis, based on a dataset of L2 learner conversations with an AI agent, shows that PPI is highly informative for evaluating the quality of weakly annotated data. Moreover, we demonstrate that PPI can increase the effective sample size by over 150% relative to the original human-scored subset, enabling more robust inference in educational assessment settings where labeled data is scarce.

1 Introduction

Recent advances in Natural Language Processing (NLP) have enabled the development of intelligent conversational agents for language learning and teaching that are capable of producing human-like language. In the context of computer-assisted language learning (CALL), research-driven, dialogue-based systems, such as task-specific conversational agents designed to support second language (L2)

acquisition, have shown promising results in fostering vocabulary and grammatical development, while also promoting self-directed learning through repeated, skills-focused practice (Bibauw et al., 2019; Tyen et al., 2022; Glandorf et al., 2025).

These technological developments have significantly enhanced the ability of dialogue-based CALL systems to guide and sustain human-like conversational interactions, aligning them with established proficiency guidelines and pedagogical principles. As a result, they offer structured, reliable, and personalized L2 practice beyond the classroom. This shift underscores the need for scalable, efficient, and statistically valid assessment methods capable of supporting such learning environments.

Automated scoring (AS) of language output, such as written essays (Shermis and Burstein, 2013), short texts (Burrows et al., 2015), spoken dialogues (Litman et al., 2018), and text-based conversations (Ramanarayanan et al., 2019; Yuwono et al., 2019), is a mature field of research that emerged during the 1960’s (Page, 1968) and has accelerated its development over the past two decades (Shermis and Burstein, 2003; Xi, 2010; Ke and Ng, 2019) as NLP methods have evolved significantly. However, AS methods rely on large quantities of high-quality manually annotated data to train models, which requires significant human resources and time.

To overcome the difficulties and challenges of data annotation in NLP, Weakly-supervised learning (WSL) emerged as an alternative framework (Huang et al., 2014), leveraging weaker sources and methods to obtain synthetic labels from textual data. Many of the strengths of WSL depend on the availability of high-quality validation data (Zhu et al., 2023), which in L2 assessment, is not always possible. Assessing L2 output for learning requires not only knowledge of the target language but also the ability to evaluate a learner’s interlanguage based on established proficiency guidelines,

making it an even more time-consuming task.

With the development of large language models (LLMs) and their advanced language understanding capabilities, researchers have begun to utilize them in data annotation tasks (Goel et al., 2023; Tan et al., 2024b). In L2 assessment in particular, GPT-4 has shown to produce holistic scores that are highly correlated to human evaluation in written essays and have moderate to high inter-rater reliability (Tate et al., 2024). Furthermore, experiments showed that GPT-4 is capable of performing analytic scoring of L2 texts given holistic scores (Banno et al., 2024), however, no ground truth set was available in this study. While state-of-the-art LLMs such as GPT-4 have shown human-like language capabilities that allow them to produce annotations that are highly correlated with expert ones, it is unclear how biased those annotations are. In addition, no statistical guarantees on the validity of the synthetic data are used in the literature.

In this paper, we investigate whether state-of-the-art large language models (LLMs) can be used to generate high-quality synthetic scores of lexical complexity and grammatical accuracy from students' text-based conversational responses based on the Common European Framework of Reference (CEFR) framework (Council of Europe, 2001). These synthetic scores, along with a small set of human-annotated gold-standard data, are used to train machine learning models under two different settings: a weakly supervised learning (WSL) approach that relies on LLM-generated labels, and a semi-supervised method in which a model trained on the gold-standard set produces predictions for a larger unlabeled corpus. In both settings, our goal is to increase the effective sample size and enable valid inference. To this end, we apply Prediction-Powered Inference (PPI) to provide statistical guarantees on the resulting predictions, ensuring that the use of synthetic scores does not compromise the validity of the conclusions.

Experimental results indicate that the proposed method increases the effective sample size by over 150% and yields a relative gain in accuracy, both compared to using only the gold-standard human-annotated data in a semi-supervised setting. In contrast, treating LLM-generated scores as if they were human-annotated can lead to inaccurate estimates and yield more modest improvements under a WSL framework. The proposed approach helps mitigate some of the limitations associated with weaker supervision sources in NLP, particularly in

scenarios where predictions inform decisions with significant consequences, such as in educational assessment.

We also address challenges associated with using LLMs as data annotators in NLP tasks, especially the uncertainty inherent in their outputs. Our findings show that applying a statistically valid method such as PPI can not only improve reliability and provide bias corrected estimates, but also quantify the uncertainty of predictions on unlabeled data, thereby offering a more trustworthy framework for leveraging synthetic annotations and scores.

The main contributions of this paper are:

- We integrate Prediction-Powered Inference into a new framework for semi-supervised and weakly supervised learning, providing statistical guarantees for predictions on datasets with small labeled and large unlabeled subsets.
- Unlike standard semi- and weakly supervised learning paradigms, the proposed framework samples and selects synthetic data based on valid statistical conditions, imposing a data quality requirement relative to a gold standard set.
- This approach, in the semi-supervised setting, produces a relative sample size gain of up to 157%, resulting in an accuracy increase of 23.2%.

2 Background

2.1 Automated L2 scoring methods

Over the past decades, computer-aided automatic text analysis has become increasingly prevalent in measuring L2 lexical and speaking proficiency (Crossley et al., 2011, 2014). More recently, deep learning approaches have achieved performance close to that of human raters in holistic scoring tasks (Alikaniotis et al., 2016), and Transformer-based models have even surpassed human inter-annotator agreement levels (Rodriguez et al., 2019). Large language models (LLMs) such as GPT-3 have also shown promise in supporting automatic scoring, as demonstrated by their application to 12,100 essays from the ETS Corpus of Non-Native Written English (Mizumoto and Eguchi, 2023).

Further advancements have been observed with GPT-4. Studies indicate that, when provided with calibration examples, GPT-4 can reliably rate short essay responses (Yancey et al., 2023), assess discourse coherence at a level comparable to expert

raters (Naismith et al., 2023), and generate analytical scores aligned with the CEFR proficiency framework (Banno et al., 2024).

While NLP-based automated methods have historically demonstrated the ability to assess specific linguistic features and functions, human raters tend to outperform them in evaluating higher-level discourse elements such as ideas, content, and organization (Enright and Quinlan, 2010). This divergence suggests that language models may exhibit a different type of bias compared to human raters, particularly in tasks requiring inferential judgment.

2.2 Weaker sources of supervision

Weakly-supervised learning (WSL) has become a practical machine learning paradigm to address the issue of label scarcity in NLP. The major bottleneck for deploying machine learning models has been the lack of access to large, high-quality training datasets. Producing manual annotations of text data is a labor-intensive and time-consuming task. To reduce such efforts, WSL approaches have been proposed to offer a larger pool of weaker supervision sources to label and annotate data (Ren et al., 2020; Zhang et al., 2021). Such sources often rely on heuristics, knowledge bases, crowd sourcing, labeling functions, or pre-trained models instead of expert manual annotations (Ratner et al., 2017). However, WSL methods also present challenges due to the degree of noise that the generated labels contain (Zhu et al., 2023).

More recently, a prompting-based method was proposed to integrate LLMs into weak supervision frameworks (Smith et al., 2024), yielding accuracy gains on the general-purpose WRENCH weak supervision benchmark. However, the effectiveness of this approach in more specialized domains, such as the analytical scoring of student responses, remains uncertain. Moreover, the study does not address potential biases present in the training data, nor does it evaluate how such biases may affect the resulting estimates. It also remains unclear how a semi-supervised method (Søgaard, 2022) would perform in comparison to this weakly supervised approach, particularly in settings where bias is limited to the human annotations and model predictions, without introducing additional external sources of error.

To address this gap, our study compares both approaches, LLM-driven weak supervision and a semi-supervised method using a small gold-standard dataset, to investigate their effectiveness

in analytical scoring tasks. We leverage PPI in both cases to provide statistical guarantees on the resulting predictions and to evaluate the reliability and calibration of the scores derived from each approach.

3 Method

We are interested in developing a framework that can be used to train machine learning models when only a small labeled dataset and a large corpus of unlabeled data are available. To leverage the unlabeled data, synthetic scores are obtained using a machine learning model, but instead of treating those scores as gold standard, a provably valid statistical method is used to assess the biases contained in the scores, so that estimates can be rectified.

The proposed framework is evaluated in two settings. In the weakly supervised learning (WSL) scenario, a state-of-the-art LLM is used to generate synthetic scores from text-based inputs, which are then used to train a traditional machine learning model. In the semi-supervised setting, the ML model is trained on a small set of gold-standard annotations and used to predict scores for a larger unlabeled set. In both cases, a debiasing protocol based on Prediction-Powered Inference (PPI) (Angelopoulos et al., 2023) is applied to estimate prediction errors and provide statistical confidence measures for the resulting scores.

Although LLMs have shown strong zero-shot generative and reasoning capabilities (Kojima et al., 2022), they still produce hallucinations (Gunjal et al., 2024), unreliable outputs (Sclar et al., 2023), and exhibit demographic biases (Chiang and Lee, 2023), making them unreliable for providing immediate scores to students. For those reasons we use machine learning models that can be trained on a set of textual features and a combination of human and weaker scoring sources to estimate a proficiency score with a given confidence (see details on models and features in subsection 4.2).

While NLP tasks, particularly those in the social sciences, have used less reliable LLM annotations in downstream tasks that require inferences to be statistically valid to draw reliable conclusions (Gligorić et al., 2024), the approach presented in this paper leverages such statistical validity to determine the reliability of weaker data sources to train machine learning models.

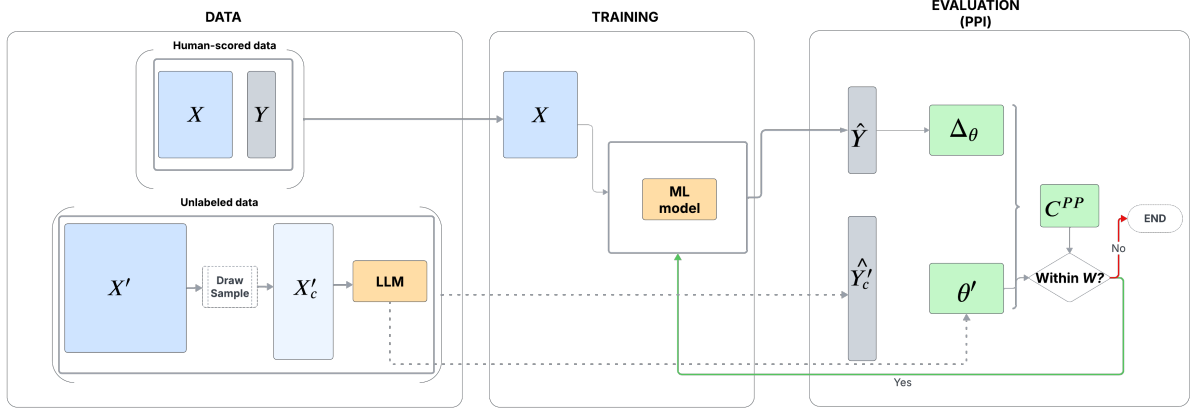


Figure 1: Outline of the process for scoring L2 conversation responses in a WSL setting using PPI. X and Y are the features and human-annotated scores, and X'_c are a subset of the textual features sampled from the unlabelled dataset X' to be scored by an LLM, obtaining \hat{Y}'_c scores and θ' verbalized confidence on the scores. If the width of C^{PP} remains within W X'_c will be added to the training process to further optimize the training of the ML model.

3.1 Statistical guarantees: Prediction-powered inference

Prediction-Powered Inference (PPI) is a statistical protocol that combines predictions made on less reliable unlabeled data with those made on a gold-standard dataset to obtain a confidence interval (CI) that is provably valid (Angelopoulos et al., 2023). Instead of using machine learning models to determine the validity of an unlabeled dataset on a case-by-case basis, PPI provides model-free estimates that are statistically valid, leveraging the information contained in the predictions.

The goal of PPI is to estimate a quantity of interest θ^* , such as the population mean. To estimate θ^* we have access to a set of gold-standard data with human-annotated responses Y and features X such that $(X, Y) = (X_1, Y_1), \dots, (X_n, Y_n)$, and a much larger set of unlabeled data $(X', Y') = (X'_1, Y'_1), \dots, (X'_N, Y'_N)$ where Y' is not directly observable, and $N \gg n$. For both datasets predictions are obtained using a machine learning model $f(\cdot)$, represented by $f(X)$ and $f(X')$. In PPI, the predictions made on the unlabeled data are not treated as gold-standard such as in the imputation case. Instead, PPI uses the gold-standard set to quantify and correct for the errors made by the model on the unlabeled set.

The three-step process that constitutes PPI can be summarized as follows:

1. Select the quantity of interest θ^* , such as the mean outcome $\mathbb{E}(Y_i)$.
2. Compute the estimate θ' and a rectifier Δ_θ , where θ' is computed on the unlabeled data

(X', \hat{Y}') such that $\theta' = \frac{1}{N} \sum_{i=1}^N f(X'_i)$, and $\Delta_\theta = \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)$. If $f(X_i)$ perfectly matches Y , then $\Delta_\theta = 0$.

3. Construct a confidence interval C^{PP} for θ^* .

To construct C^{PP} we need to obtain the prediction-powered estimate $\hat{\theta}^{PP}$ that corrects for the bias on θ' due to prediction errors:

$$\hat{\theta}^{PP} = \frac{1}{N} \sum_{i=1}^N f(X'_i) - \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i) \quad (1)$$

and then the prediction-powered confidence set is obtained such that

$$C^{PP} = (\hat{\theta}^{PP} \pm w(\alpha)) \quad (2)$$

where $w(\alpha)$ is a constant that depends on the confidence level α (derivations could be found in Angelopoulos et al. (2023)).

PPI has been used for the pairwise ranking of models (Boyeau et al., 2024), for comparing the performance of LLMs (Chatzi et al., 2024), for evaluating retrieval augmented generation (RAG) systems (Saad-Falcon et al., 2024), and some of its variants for producing confident conclusions from LLMs annotations (Gligorić et al., 2024). The approach presented in this article differs from the previous ones. In the general PPI setting, a trained model is used to produce predictions on both sets, and PPI is used to debias the predictions made on the unlabeled data. In the proposed framework, we do not have access to a trained model, and the

training is done iteratively and sequentially using PPI as a guarantee of statistical validity, making decisions on what unlabeled data to include in the training process based on a statistical measure.

3.2 Using LLMs in weak supervision with statistical guarantees

From a dataset of conversational responses \mathcal{X} , we divide it into $|X|$ responses to be scored by human annotators and $|X'|$ responses to be scored by an LLM, obtaining Y and Y' scores respectively; where $|X| = n$ and $|X'| = N$, and $N \gg n$. We assume that there are biases in \hat{Y}' associated with the scoring errors made by the LLM, and use PPI to debias them, resulting in biased-corrected estimates.

To obtain Y and Y' , the same rubric was used for human and LLM scorers, in an attempt to maintain as much parity as possible between the two scoring sources and to avoid additional biases.

The rubric used two dimensions of language proficiency as expressed in the CEFR framework (Council of Europe, 2001), namely vocabulary range and grammatical accuracy for B1 and B2 levels. Scores ranged from 1-3 for vocabulary range and 1-4 for grammatical accuracy. For human scoring, two annotators with extensive experience in L2 proficiency scoring were recruited. To obtain a single score, annotations were conducted collaboratively, and if consensus was not reached a third annotator was used to resolve the disagreement (Fort, 2016) (see Appendix A for details on the rubric). GPT-4o and GPT-4o-mini were the models of choice to produce synthetic scores Y' given X' and the rubric. A zero-shot prompting approach was used (see Appendix C for details on the prompts) and additionally, the models were prompted to provide a measure of *verbalized confidence* in the form of a probability value to assess the correctness of the score, as presented in Tian et al. (2023).

The following steps describe the WSL approach with an LLM as a *weak* scorer and with PPI guarantees: 1) taking as input the entire set of high-quality human-labeled scores, a machine learning model $f(\cdot)$ is trained such that, after training on the gold-standard set X is completed, we obtain $\hat{\theta}^{PP}$ and C^{PP} using the verbalized confidence of the LLM θ' on a small sample of size c X'_c and the predictions made by the ML model \hat{Y}' ; 2) the width of C^{PP} is computed in an evaluation step, such that $C_{upper}^{PP} - C_{lower}^{PP} \leq W$ and W is a width threshold

chosen beforehand; 3) if the width is not greater than W and the prediction-powered corrected mean accuracy is not less than the one computed with the gold-standard set, the sample X'_c is added to the training process and the model is trained on $X \oplus X'_c$ until either the C^{PP} condition on W is no longer met or the accuracy decreases. Figure 1 outlines the process in a block diagram.

3.3 Leveraging PPI in semi-supervised learning

Similar to the WSL approach described in Subsection 3.2, the proposed semi-supervised method uses PPI to establish statistical guarantees on predictions made for the unlabeled data X' . However, instead of relying on an LLM as a scorer, this method employs a machine learning model to generate predictions, which are then reused for fine-tuning under the same width and accuracy gain conditions defined in the WSL setting.

The method works as follows. First, the ML model is trained only on the human-scored set, the ground-truth data X . In an evaluation step, a sample of size c is randomly drawn from the entire unlabeled set X' to compute $\hat{Y}'_c = f(X'_c)$ and obtain $\hat{\theta}^{PP}$ and C^{PP} . If the width of C^{PP} does not exceed a threshold W , i.e., $C_{upper}^{PP} - C_{lower}^{PP} \leq W$, and $\hat{\theta}^{PP}$ is greater than the mean accuracy of the predictions made using the human-scored data, then a new sample is drawn and training continues until this condition is no longer satisfied (Figure 2 summarizes the steps involved in this process).

As we can see, PPI is used to estimate the validity of the inferences made on the unlabeled set through an iterative process that draws samples of size c to test whether the predictions on X' maintain the width of C^{PP} within the threshold W . A wider width would denote greater uncertainty, indicating that the predictions made on the unlabeled data are less reliable and potentially more biased. In contrast, a narrower width suggests higher precision and lower variance.

In this sense, C^{PP} serves as a valid estimate for assessing the quality of an unlabeled dataset given a high-quality labeled one. It also provides a basis for estimating the effective sample size needed to obtain reliable predictions when leveraging unlabeled data, especially when considering the associated accuracy gain.

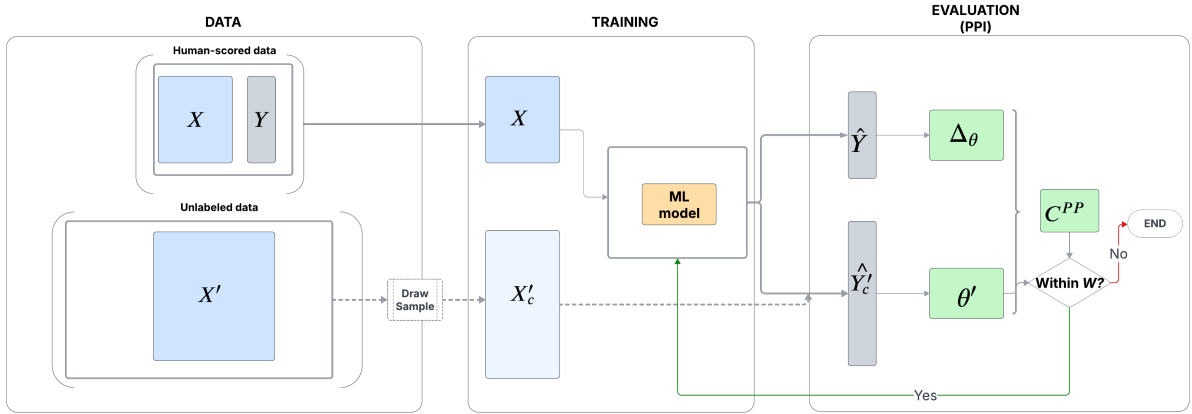


Figure 2: Differently from the process outlined in Figure 1, in the semi-supervised setting both the scores \hat{Y}'_c and the probability value θ' are obtained directly from the ML model. No external scoring source is used.

4 Experiments

We conduct several experiments to evaluate the effectiveness of our approach. The goal is to assess the overall methodology in both weakly and semi-supervised learning settings, aiming to measure the quality of synthetically generated scores derived from a smaller set of high-quality, human-annotated data. Given the constraints of this study, namely the limited availability of high-quality human-labeled samples, we use machine learning models that are well-suited to this low-resource setting and that have shown to perform effectively on features that can be represented as tabular data (Shwartz-Ziv and Armon, 2022). We make code available ¹.

4.1 Data

Data was collected from text-based conversation practice sessions completed by intermediate level (B1-B2 levels in CEFR) English language learners (ELLs) and an AI agent (Fincham and Alvarez, 2024) over a 3-month period. A total 121 students from 3 sessions of an undergraduate course focused on English speaking participated in the project and generated 1721 practice sessions. The average number of turns per session produced by students was 8.9.

To train the models, 590 sessions were manually scored following a rubric based on the CEFR framework (Council of Europe, 2001) on vocabulary range and grammatical accuracy (see section 3.2). Out of the 590 sessions, 445 were used for training and 145 for evaluation. The remaining

1131 sessions were either scored by an LLM or by the model of choice in the semi-supervised experiment. Inter-annotator agreement between human and LLM raters reached moderate levels, with $\kappa = 0.45$ for vocabulary range and $\kappa = 0.4$ for grammatical accuracy.

4.2 Models and features

From the students' conversations, 9 lexical and syntactical features were automatically extracted, many of which have shown to be highly correlated with linguistic proficiency descriptors based on the CEFR framework (Banno et al., 2024). Those are: lexical density, unique noun chunks, number of unique words, number of unique difficult words, Flesch Kincaid readability score (Thomas et al., 1975), sentence length mean and standard deviation, and dependency distance mean and standard deviation.

Two tree-based boosting models, XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017) were used in the weak and semi-supervised training regimes and used as predictor ML models. Results were compared to the baseline scores obtained directly from the two LLMs, GPT-4o and GPT-4o-mini.

4.3 Evaluation

The quantity of interest chosen for this study was the mean accuracy. As described in subsections 3.2 and 3.3, width and accuracy gain are the measures that determine the stopping condition during training and evaluation, and overall, to determine the quality of the inference on the unlabeled data. In addition, coverage is used to evaluate how many times the true value θ^* falls within the estimated

¹<https://github.com/aitor-alvarez/Automated-L2-Proficiency-Scoring>

interval with a given confidence level. The confidence level for this study was set to 0.9 for $\alpha = 0.1$, which is the same level that PPI guarantees asymptotically (Angelopoulos et al., 2023). If coverage does not meet the established α -level, this would indicate that predictions are extremely biased, not normally distributed, or that the proportion or quality of the labeled/unlabeled data is not balanced and therefore the variance estimate may be unstable.

We estimate the effective sample size by calculating the maximum number of synthetic scores used in relation to the labeled set for the following inequality to hold $C_{upper}^{PP} - C_{lower}^{PP} \leq W$ and for the mean accuracy to improve when comparing it to the results obtained with the gold-standard set alone. The width threshold was set at $W = 0.2$ and tested in the two experimental learning settings (weakly and semi-supervised) for each of the models. This width threshold indicates that we are willing to accept a CI with a maximum of 20% range in the mean accuracy estimate C^{PP} . Samples of size 100 were added at each iteration to determine the width, coverage, and effective sample size for both conditions.

5 Results

Table 1 presents the experimental results. The semi-supervised approach yields the highest accuracy gains by using PPI to combine human-annotated data with model-generated scores, selecting only samples within the PPI confidence interval that improve baseline accuracy. For vocabulary range, the increase reaches 23.2% and for grammatical accuracy 21.5%, with a total effective sample size of 700. Width sizes remain relatively low, 0.13 for vocabulary range and at around 0.148 for grammatical accuracy. Coverage in this setting reaches 97%, demonstrating the validity of this approach.

The weakly supervised learning (WSL) protocol, on the other hand, yields more modest accuracy gains when combining gold-standard data with weakly scored data. In this setting, accuracy improvements range from 8.1% to 8.4% in vocabulary range and reach 7.5% in grammatical accuracy, using either boosting model with GPT-4o as the annotator source and an effective sample size of 200. When GPT-4o-mini is used as the annotator model, accuracy gain decreases to 4.2–3.4% in vocabulary range and 3.1% in grammatical accuracy, with an effective sample size of 100. Overall, the WSL setting shows a coverage slightly above 90% (91%),

indicating an acceptable validity of this approach when using LLMs as weak scorers (see Appendix B for the accuracy on the gold-standard set only).

The LLM-only approach yields modest accuracy gains, with GPT-4o achieving improvements of 1.8% in vocabulary range and 1.5% in grammatical accuracy. GPT-4o-mini shows smaller gains of 0.6% and 0.4%, respectively, under the same setting, with an effective sample size of 100 in both cases. However, despite these gains, the coverage remains below 90%, failing to meet the required validity threshold. Further analysis reveals that both LLMs exhibit overconfidence in their probability estimates. On average, GPT-4o assigns 80% confidence to incorrect predictions in vocabulary range and 75% in grammatical accuracy. GPT-4o-mini shows similar patterns, with 78% confidence in vocabulary range and 71% in grammatical accuracy for its incorrect predictions.

In summary, the results indicate that the method presented in this study, when applied in a semi-supervised setting, results in a dataset that is 157% larger than the original. In contrast, the sample size gain is significantly reduced, down to 22%, when the method is used in a WSL setting with LLMs as scorers. Moreover, the naive approach of directly using LLM responses as gold-standard predictions fails to produce valid results.

6 Discussion

In this study, we have presented an approach to integrate Prediction-Powered Inference (PPI) in semi- and weakly-supervised settings when gold-standard data is scarce or difficult to obtain. By using PPI, we have shown that gold-standard with less reliable data can be combined to obtain increases in predictive accuracy while maintaining the validity of the results. This is particularly important in the context of this study, where student-produced output, namely conversational responses obtained from student interactions with an AI tutor, requires assessment at scale that can provide valid feedback to learners.

As previous studies have demonstrated (Tate et al., 2024; Tan et al., 2024b,a), LLM outputs show moderate to strong agreement with human judgments. In our study, we observe a moderate level of agreement between human raters and LLM-based scores, as previously reported. However, this level of agreement is insufficient for treating LLM scores as gold-standard, as it does not yield valid

Setting	Task	Model	Sample Size	Width	Coverage	Acc. Gain
Semi	Vocab. range	XGBoost	700	0.13	97%	23.2%
Semi	Gram. accur.	XGBoost	700	0.145	97%	21.5%
Semi	Vocab. range	LightGBM	700	0.133	97%	22.1%
Semi	Gram. accur.	LightGBM	700	0.148	97%	19.7%
WSL	Vocab. range	XGBoost + 4o	200	0.136	91 %	8.4%
WSL	Gram. accur.	XGBoost + 4o	200	0.144	91 %	7.5%
WSL	Vocab. range	XGBoost + 4o-mini	100	0.16	91 %	4.2%
WSL	Gram. accur.	XGBoost + 4o-mini	100	0.171	91 %	3%
WSL	Vocab. range	LightGBM+ 4o	200	0.14	91 %	8.1%
WSL	Gram. accur.	LightGBM+ 4o	200	0.145	91 %	7.5%
WSL	Vocab. range	LightGBM+ 4o-mini	100	0.177	91 %	3.4%
WSL	Gram. accur.	LightGBM+ 4o-mini	100	0.179	91 %	3.1%
LLM only	Vocab. range	GPT-4o	100	0.152	77%	1.8%
LLM only	Gram. accur.	GPT-4o	100	0.156	77%	1.5%
LLM only	Vocab. range	GPT-4o-mini	100	0.181	68%	0.6%
LLM only	Gram. accur.	GPT-4o-mini	100	0.184	68%	0.4%

Table 1: Performance metrics by setting, task, and model employed to obtain predictions and to generate synthetic scores. Sample size indicates the maximum number of unlabeled samples used to reach the highest accuracy and within the maximum width allowed. Acc. Gain is the maximum gain in accuracy compared to the gold-standard (human annotated only) approach (see Appendix B for the accuracy on the gold-standard set only).

statistical conclusions.

When LLM outputs are instead used within a weak supervision framework, acknowledged as biased but corrected through prediction-powered inference (PPI), they lead to slight improvements compared to relying solely on gold-standard data. Nonetheless, LLMs exhibit overconfidence in their incorrect assessments, indicating a poor understanding of uncertainty, a concern also noted in a recent work (Pawitan and Holmes, 2025).

In contrast, we find that a well-calibrated machine learning model, when used in a semi-supervised setting alongside PPI, can substantially increase the sample size by over 157% relative to using only human-annotated data, which results in a dataset larger than the original and improves training and accuracy. This suggests that simpler models, when well-calibrated and properly integrated into such frameworks, can support broader, validity-guaranteed conclusions in educational assessment settings.

The results obtained in this paper have broad implications for large-scale and AI-mediated learning environments, where many learners require assessment and guidance, and human feedback is impractical (Swiecki et al., 2022). In such contexts, a small, well-annotated dataset can be used to make valid predictions on larger unlabeled data, reducing training requirements, improving prediction

quality, and enabling large-scale assessments with validity guarantees.

7 Limitations

This study aimed to explore the potential of Prediction-Powered Inference (PPI) to extend a small set of high-quality, human-scored conversational responses using less reliable data generated by large language models (LLMs) and simpler machine learning models. While PPI offers a statistically grounded framework for leveraging such predictions, it assumes that the labeled data are independently and identically distributed (i.i.d.) from a normal distribution. Although our labeled sample size ($n = 445$) may appear limited, each session includes, on average, over eight student turns, providing a richer source of information per data point. Nonetheless, larger samples of gold-standard data should be examined in future work to validate and generalize the findings presented here. Expanding the dataset to include a broader range of learner proficiencies could also provide further insights into the robustness and adaptability of the proposed approach.

8 Ethical considerations

In this study, we caution against the use of LLM-generated outputs as ground-truth data in educa-

tional settings, emphasizing the risks associated with treating such predictions as authoritative. Nevertheless, we acknowledge the potential of LLMs in low-stakes educational scenarios, particularly for generating synthetic data or instructional materials that can support learning.

It is important to note that this work assumes human annotations as the gold standard. However, this assumption should be approached with caution, as human judgments are subject to both cognitive (Gautam and Srinath, 2024) and socio-cultural biases (Huang and Yang, 2023), which are often context-dependent and may impact the reliability of reference scores.

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.
- Anastasios N Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I Jordan, and Tijana Zrnica. 2023. Prediction-powered inference. *Science*, 382(6671):669–674.
- 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.
- Serge Bibauw, Thomas François, and Piet Desmet. 2019. Discussing with a computer to practice a foreign language: Research synthesis and conceptual framework of dialogue-based call. *Computer Assisted Language Learning*, 32(8):827–877.
- Pierre Boyeau, Anastasios N Angelopoulos, Nir Yosef, Jitendra Malik, and Michael I Jordan. 2024. AutoEval done right: Using synthetic data for model evaluation. *arXiv preprint arXiv:2403.07008*.
- 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.
- Ivi Chatzi, Eleni Straitouri, Suhas Thejaswi, and Manuel Rodriguez. 2024. Prediction-powered ranking of large language models. *Advances in Neural Information Processing Systems*, 37:113096–113133.
- 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.
- Cheng-Han Chiang and Hung-yi Lee. 2023. [Can large language models be an alternative to human evaluations?](#) In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.
- Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press, Cambridge.
- Scott Crossley, Amanda Clevinger, and YouJin Kim. 2014. The role of lexical properties and cohesive devices in text integration and their effect on human ratings of speaking proficiency. *Language Assessment Quarterly*, 11(3):250–270.
- Scott A Crossley, Tom Salsbury, Danielle S McNamara, and Scott Jarvis. 2011. Predicting lexical proficiency in language learner texts using computational indices. *Language testing*, 28(4):561–580.
- Mary K Enright and Thomas Quinlan. 2010. Complementing human judgment of essays written by english language learners with e-rater® scoring. *Language Testing*, 27(3):317–334.
- Naiyi Xie Fincham and Aitor Arronte Alvarez. 2024. Using large language models (LLMs) to facilitate l2 proficiency development through personalized feedback and scaffolding: An empirical study. In *Proceedings of the International CALL Research Conference*, volume 2024, pages 59–64.
- Karën Fort. 2016. *Collaborative annotation for reliable natural language processing: Technical and sociological aspects*. John Wiley & Sons.
- Sanjana Gautam and Mukund Srinath. 2024. [Blind spots and biases: Exploring the role of annotator cognitive biases in NLP](#). In *Proceedings of the Third Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 82–88, Mexico City, Mexico. Association for Computational Linguistics.
- Dominik Glandorf, Peng Cui, Detmar Meurers, and Mrinmaya Sachan. 2025. [Grammar control in dialogue response generation for language learning chatbots](#). 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 9820–9839, Albuquerque, New Mexico. Association for Computational Linguistics.
- Kristina Gligorić, Tijana Zrnica, Cino Lee, Emmanuel J Candès, and Dan Jurafsky. 2024. Can unconfident LLM annotations be used for confident conclusions? *arXiv preprint arXiv:2408.15204*.
- Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, et al.

2023. Llms accelerate annotation for medical information extraction. In *Machine Learning for Health (ML4H)*, pages 82–100. PMLR.
- Anisha Gunjal, Jihan Yin, and Erhan Bas. 2024. Detecting and preventing hallucinations in large vision language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18135–18143.
- Fei Huang, Arun Ahuja, Doug Downey, Yi Yang, Yuhong Guo, and Alexander Yates. 2014. Learning representations for weakly supervised natural language processing tasks. *Computational Linguistics*, 40(1):85–120.
- Jing Huang and Diyi Yang. 2023. Culturally aware natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7591–7609, Singapore. 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.
- Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *IJCAI*, volume 19, pages 6300–6308.
- 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.
- Diane Litman, Helmer Strik, and Gad S Lim. 2018. Speech technologies and the assessment of second language speaking: Approaches, challenges, and opportunities. *Language Assessment Quarterly*, 15(3):294–309.
- Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an AI language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.
- Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. Automated evaluation of written discourse coherence using GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, Toronto, Canada. Association for Computational Linguistics.
- Ellis B Page. 1968. The use of the computer in analyzing student essays. *International review of education*, 14:210–225.
- Yudi Pawitan and Chris Holmes. 2025. Confidence in the reasoning of large language models. *Harvard Data Science Review*, 7(1).
- Vikram Ramanarayanan, Matthew Mulholland, and Yao Qian. 2019. Scoring interactional aspects of human-machine dialog for language learning and assessment using text features. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 103–109, Stockholm, Sweden. Association for Computational Linguistics.
- Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: rapid training data creation with weak supervision. *Proc. VLDB Endow.*, 11(3):269–282.
- Wendi Ren, Yinghao Li, Hanting Su, David Kartchner, Cassie Mitchell, and Chao Zhang. 2020. Denoising multi-source weak supervision for neural text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3739–3754, Online. Association for Computational Linguistics.
- Pedro Uria Rodriguez, Amir Jafari, and Christopher M Ormerod. 2019. Language models and automated essay scoring. *arXiv preprint arXiv:1909.09482*.
- Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. ARES: An automated evaluation framework for retrieval-augmented generation systems. 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 338–354, Mexico City, Mexico. Association for Computational Linguistics.
- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.
- Mark D Shermis and Jill Burstein. 2013. *Handbook of automated essay evaluation*. Routledge, New York.
- Mark D Shermis and Jill C Burstein. 2003. *Automated Essay Scoring: A Cross-disciplinary Perspective*. Routledge, New York.
- Ravid Shwartz-Ziv and Amitai Armon. 2022. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90.
- Ryan Smith, Jason A Fries, Braden Hancock, and Stephen H Bach. 2024. Language models in the loop: Incorporating prompting into weak supervision. *ACM/JMS Journal of Data Science*, 1(2):1–30.
- Anders Søgaard. 2022. *Semi-supervised learning and domain adaptation in natural language processing*. Springer Nature.
- Zachari Swiecki, Hassan Khosravi, Guanliang Chen, Roberto Martinez-Maldonado, Jason M Lodge, Sandra Milligan, Neil Selwyn, and Dragan Gašević. 2022. Assessment in the age of artificial intelligence. *Computers and Education: Artificial Intelligence*, 3:100075.

- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024a. Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*.
- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024b. [Large Language Models for data annotation and synthesis: A survey](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.
- Tamara P Tate, Jacob Steiss, Drew Bailey, Steve Graham, Youngsun Moon, Daniel Ritchie, Waverly Tseng, and Mark Warschauer. 2024. Can AI provide useful holistic essay scoring? *Computers and Education: Artificial Intelligence*, 7:100255.
- Georgelle Thomas, R Derald Hartley, and J Peter Kincaid. 1975. Test-retest and inter-analyst reliability of the automated readability index, flesch reading ease score, and the fog count. *Journal of Reading Behavior*, 7(2):149–154.
- Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. [Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.
- 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.
- Xiaoming Xi. 2010. Automated scoring and feedback systems: Where are we and where are we heading? *Language Testing*, 27(3):291–300.
- Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. [Rating short L2 essays on the CEFR scale with GPT-4](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584, Toronto, Canada. Association for Computational Linguistics.
- Steven Kester Yuwono, Biao Wu, and Luis Fernando D’Haro. 2019. Automated scoring of chatbot responses in conversational dialogue. In *9th International Workshop on Spoken Dialogue System Technology*, pages 357–369. Springer.
- Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021. WRENCH: A comprehensive benchmark for weak supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. 2023. [Weaker than you think: A critical look at weakly supervised learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14229–14253, Toronto, Canada. Association for Computational Linguistics.

A Rubric

A.1 Vocabulary Range (B1-B2)

Score	Description of the proficiency level
1	Has a good range of vocabulary related to familiar topics and everyday situations. Has sufficient vocabulary to express themselves with some circumlocutions on most topics pertinent to their everyday life such as family, hobbies and interests, work, travel and current events.
2	Has a good range of vocabulary for matters connected to their field and most general topics. Can vary formulation to avoid frequent repetition, but lexical gaps can still cause hesitation and circumlocution. Can produce appropriate collocations of many words/signs in most contexts fairly systematically. Can understand and use much of the specialist vocabulary of their field but has problems with specialist terminology outside it.
3	Can understand and use the main technical terminology of their field, when discussing their area of specialisation with other specialists.

Table 2: Vocabulary Range Rubric (B1-B2)

A.2 Grammatical Accuracy (B1-B2)

Score	Description of the proficiency level
1	Uses reasonably accurately a repertoire of frequently used routines and patterns associated with more predictable situations.
2	Communicates with reasonable accuracy in familiar contexts; generally good control, though with noticeable mother-tongue influence. Errors occur, but it is clear what they are trying to express.
3	Has a good command of simple language structures and some complex grammatical forms, although they tend to use complex structures rigidly with some inaccuracy.
4	Good grammatical control; occasional slips or non-systematic errors and minor flaws in sentence structure may still occur, but they are rare and can often be corrected in retrospect.

Table 3: Grammatical Accuracy Rubric (B1-B2)

B Accuracy for gold standard set

Model	Proficiency level	Accuracy
XGBoost	Vocabulary Range	74.1
XGBoost	Grammatical accuracy	71.3
LightGBM	Vocabulary Range	73.6
LightGBM	Grammatical accuracy	71
GPT-4o	Vocabulary Range	62.5
GPT-4o	Grammatical accuracy	61.2
GPT-4o-mini	Vocabulary Range	60.3
GPT-4o-mini	Grammatical accuracy	58.9

Table 4: Accuracy values for each of the models used tested on the gold standard set only.

C Prompts

Score the following text from a conversation of an intermediate English language student (B1-B2 on CEFR).

Provide the score as an integer and the probability as a float associated with the options in the 'ScoringTexts' function.

Text: text

```
class ScoringTexts(BaseModel):
    #CEFR vocabulary range.
    vocabulary_range: int = Field(description="Select the option that best describes
        the text."
                                   "Option 1. Has a good range of vocabulary
                                   related to familiar topics and
                                   everyday situations."
                                   "Has sufficient vocabulary to express
                                   themselves with some circumlocutions
                                   on most topics "
                                   "pertinent to their everyday life such
                                   as family, hobbies and interests,
                                   work, travel and current events."
                                   "Option 2. Has a good range of
                                   vocabulary for matters connected to
                                   their field and most general topics."
                                   "
                                   "Can vary formulation to avoid frequent
                                   repetition, but lexical gaps can
                                   still cause hesitation"
                                   " and circumlocution."
                                   "Can produce appropriate collocations of
                                   many words/signs in most contexts
                                   fairly systematically."
                                   "Can understand and use much of the
                                   specialist vocabulary of their field
                                   but has problems with "
                                   "specialist terminology outside it."
                                   "Option 3. Can understand and use
                                   technical terminology when
                                   discussing "
                                   "areas of specialization. Have access to
                                   specialized vocabulary in relation
                                   to the topic.")

    vocabulary_range_proba: float = Field(description="Express in the form of a
        probability the confidence on the vocabulary range score given.")

    #measures of grammatical accuracy as per CEFR
    grammatical_accuracy: int = Field(description="Select the option that best
        describes the text."
                                           "Option 1. Uses reasonably accurately
                                           a repertoire of frequently used
                                           routines and patterns "
                                           "associated with more predictable
                                           situations. "
                                           "Option 2. Communicates with
                                           reasonable accuracy in familiar
                                           contexts; generally good control,
                                           "
                                           "though with noticeable mother-
                                           tongue influence."
                                           "Errors occur, but it is clear what
                                           they are trying to express."
                                           "Option 3. Has a good command of
                                           simple language structures and
                                           some complex grammatical forms, "
                                           "although they tend to use complex
```

```
structures rigidly with some
inaccuracy."
"option 4. Good grammatical control;
occasional slips or non-systematic
errors and minor flaws "
"in sentence structure may still
occur, "
"but they are rare and can often be
corrected in retrospect.")
```

```
grammatical_accuracy_proba: float = Field(description="Express in the form of a
probability the confidence on the grammatical accuracy score given.")
```

Automatic Generation of Inference Making Questions for Reading Comprehension Assessments

Wanjing Anya Ma
Stanford University *
Stanford, CA, USA
wanjingm@stanford.edu

Michael Flor
ETS Research Institute
Princeton, NJ, USA
MFlor@ets.org

Zuwei Wang
ETS Research Institute
Princeton, NJ, USA
zwang@ets.org

Abstract

Inference making is an essential but complex skill in reading comprehension (RC). Some inferences require resolving references across sentences, and some rely on using prior knowledge to fill in the detail that is not explicitly written in the text. Diagnostic RC questions can help educators provide more effective and targeted reading instruction and interventions for school-age students. We introduce a taxonomy of inference types for RC and use it to analyze the distribution of items within a diagnostic RC item bank. Next, we present experiments using GPT-4o to generate bridging-inference RC items for given reading passages via few-shot prompting, comparing conditions with and without chain-of-thought prompts. Generated items were evaluated on three aspects: overall item quality, appropriate inference type, and LLM reasoning, achieving high inter-rater agreements above 0.90. Our results show that GPT-4o produced 93.8% good-quality questions suitable for operational use in grade 3-12 contexts; however, only 42.6% of the generated questions accurately matched the targeted inference type. We conclude that combining automatic item generation with human judgment offers a promising path toward scalable, high-quality diagnostic RC assessments.

1 Introduction

Inference-making is an essential yet cognitively demanding skill in reading comprehension (RC) (O'Brien et al., 2015; Kintsch, 1998). Inferences are necessary for establishing both local and global coherence within the mental representation of a text (Graesser et al., 1994). Local inferences connect information across sentences using cohesive devices such as anaphors or category exemplars—for example, in "Bette gulped down the drink. The cold water was very refreshing," the reader infers that *the drink* refers to *cold water* (Cain, 2022, p. 307).

Global inferences, on the other hand, rely on the reader's prior knowledge to fill in missing details required to make sense of the text—for example, in "The campfire started to burn uncontrollably. Tom grabbed a bucket of water" (Bowyer-Crane and Snowling, 2005, p. 192), the reader infers that Tom intended to put out the fire, based on the knowledge that water extinguishes fire. While skilled readers often generate inferences automatically as they engage with text (Thurlow and van den Broek, 1997), children who struggle with comprehension frequently have difficulty constructing these inferences (Cain et al., 2001).

Providing diagnostic information about specific types of inference-making deficits that hinder comprehension can empower educators to provide more effective and targeted reading instruction and intervention (Bowyer-Crane and Snowling, 2005; Bayat and Çetinkaya, 2020). To achieve this, we need RC assessments that specifically target inference-making types. At the same time, we want to develop scalable item generation methods to enable multi-time testing, monitoring reading development over time. Previous work has demonstrated the ability of large language models (LLMs) to generate effective RC questions (Uto et al., 2023; Säuberli and Clematide, 2024). However, whether LLMs can reliably produce questions that target specific inference types remains unclear.

Our research is grounded in a real-world diagnostic assessment of reading skills for students in grades 3 through 12 (Sabatini et al., 2019). The assessment was originally developed at ETS and recently commercialized as ReadBasix. It leverages the science of reading to assess foundational reading skills, such as word recognition and decoding, as well as more complex ones such as RC. In the RC subtest, a student will usually read 4 expository passages and answer multiple-choice questions associated with the passages. The subtest takes about 30 minutes to complete. Like any

*Work done while at ETS Research Institute

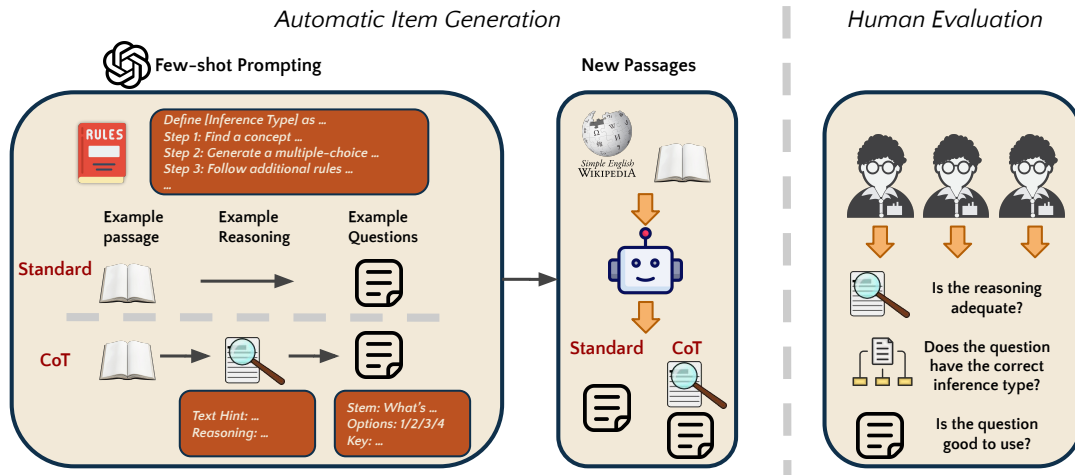


Figure 1: **Overview of automatic item generation and human evaluation.** We use GPT-4o to generate bridging-inference RC items for given reading passages via few-shot prompting, comparing conditions with and without chain-of-thought prompts. We prompt each inference type separately: pronominal bridging, text-connecting, and gap-filling inferences. Human evaluation focuses on general item quality, inference type appropriateness, and LLM rationales.

large-scale reading assessment, there is an ongoing need for more items. To address this demand, we aim to leverage automatic item generation to create new items based on curated passages, and evaluate the quality of these items before collecting student performance data to make them operational.

For the purpose of automatic item generation, as illustrated in Figure 1, we first conducted a literature review on inference-making in the reading comprehension and natural language processing (NLP) text comprehension literature. We developed a taxonomy of inference-making questions, with a focus on bridging inference. We validated this taxonomy by annotating an operational item bank of expert-written RC questions, confirming bridging inference as an important and widely covered sub-construct. Next, we curated six expository passages and manually wrote multiple-choice RC questions for each inference type based on our taxonomy. These examples were then used to prompt GPT-4o (Hurst et al., 2024) via few-shot prompting to generate bridging-inference questions for new reading passages, comparing conditions with and without chain-of-thought (CoT) prompting (Wei et al., 2022). Finally, three human experts evaluated the quality of the generated questions along three dimensions: overall item quality ¹, appropriate inference type, and whether GPT-4o provided

¹The evaluation of overall item quality does not include whether an item is of the required inference type, which is an extra-evaluation. See Table 2 for more details.

satisfactory reasoning for generating the question. Our results show that LLMs can produce 93.8% good-quality questions suitable for operational use in grade 3-12 contexts; however, only 42.6% of the generated questions accurately match the targeted inference type. Nevertheless, the overall coverage of inference types closely mirrors what we observe in our operational item bank. We conclude that combining automatic item generation with human judgment offers a promising path toward scalable, high-quality diagnostic RC assessments.

In summary, we make the following contributions in this paper:

1. We develop and validate a taxonomy for inference-making questions used in multiple-choice RC assessments, and demonstrate its value for future item development.
2. We introduce a novel NLP task where language models generate RC questions targeting specific inference types, providing a new way to assess their reasoning abilities. The training item bank will be released for replication and benchmarking.
3. We demonstrate GPT-4o’s potential in generating RC questions for operational use and its limitations in accurately generating specific types of inference questions.

2 Related Work

2.1 Question generation for reading comprehension assessments

Automatic question generation is a well-established task in NLP, especially within educational applications, to reduce the high costs of manual question authoring and to ensure a steady supply of new, high-quality items (Kurdi et al., 2020). Early approaches rely on rule-based or template-based methods (Araki et al., 2016; Flor and Riordan, 2018), as well as the use of discourse connectives to generate questions (Agarwal et al., 2011). Later approaches extensively used neural systems for question generation (Mulla and Gharpure, 2023). More recent work demonstrates that LLMs hold promise in generating high-quality RC questions, using techniques such as fine-tuning (Uto et al., 2023; Perkoff et al., 2023; Ghanem et al., 2022; Ashok Kumar et al., 2023; Rathod et al., 2022; Stasaski et al., 2021), zero-shot or few-shot prompting (Säuberli and Clematide, 2024; Attali et al., 2022), and Chain-of-Thought prompting (Kulshreshtha and Rumshisky, 2022). Some of these studies have also explored the generation of more complex, "deeper" questions—those that target underlying reasoning processes (Ghanem et al., 2022; Poon et al., 2024) or hinge on specific inference steps for accurate responses (Araki et al., 2016). Within the domain of automated Question Answering, the notion of *multi-hop questions* has gained attention, as questions relating different parts of a document require multi-step reasoning (Mavi et al., 2024).

We note that prior studies have largely treated reading comprehension as a single, undifferentiated construct even though comprehension requires different types of inferences. Recent work has begun to develop taxonomies of RC and annotate question types to enable more controllable generation (Xu et al., 2022; Li and Zhang, 2024; Hwang et al., 2024). However, to our knowledge, no existing work has systematically addressed question generation based on specific types of inference. We believe that the capability to generate different types of inference questions will provide more diagnostic insights for educators. Our work is a first step toward filling this gap.

2.2 Bridging inference as an NLP task

The NLP community has long tackled text comprehension challenges, including bridging infer-

ence. Prior work has focused on corpus-based bridging anaphora recognition and resolution using annotated resources such as ISNotes and BASHI (Rösiger, 2018; Hou et al., 2018; Hou, 2020). Neural models have been developed to jointly learn mention representations and bridging relations (Pandit and Hou, 2021; Kobayashi et al., 2022). In the recently developed IdentifyMe benchmark for resolving nominal and pronominal mentions across long contexts (Manikantan et al., 2024), GPT-4o outperforms other LLMs, achieving 81.9% accuracy and demonstrating strong referential capabilities. With the rise of LLMs, research increasingly shifts toward evaluating LLMs' general reasoning capabilities (Brown et al., 2020; Wei et al., 2022). In our education application, we investigate whether LLMs truly possess the reasoning ability required for bridging inference, particularly through the lens of a question generation task.

3 Taxonomy of Inference Questions

3.1 Development of Taxonomy

Inferences can be categorized into bridging inferences, elaborative inferences, predictive inference, emotional inference, etc (Graesser et al., 1994; Schmalhofer et al., 2002; Singer and Remillard, 2004; van den Broek et al., 2015). To manage the scope of our interest, we focus on bridging inference which connects information in a text. Bridging inferences contribute to text coherence by allowing the reader to identify the connections among concepts and ideas in the text (Singer et al., 1992; Singer and Remillard, 2004) or bridges (Haviland and Clark, 1974) among the propositions underlying the discourse. A bridging inference is needed when the reader cannot retrieve a referent for the given information of the current sentence from either working memory or long-term memory.

Table 1 shows the taxonomy of inference making questions for diagnostic RC assessments, along with the examples. The first type is **pronominal**, and it has two variants. Simple pronominal asks for a direct pronoun resolution, such as "In the sentence, whom does 'he' refer to?" This is different from the second subtype: **pronominal bridging**, which requires the reader to use the pronoun as a hint to bridge sentences and answer the question. The third type **text-connecting** requires test takers to connect two explicitly stated components in a text, and usually the bridge are noun phrases. The last type is **gap-filling**, which requires readers to

Types	Definitions	Examples
Pronominal	Direct pronoun resolution.	Like "To whom 'he' refers?", "What does 'this' represent?"
Pronominal Bridging	Use pronoun as a hint to bridge sentences.	Text snippet: <i>Ships have carried passengers since prehistoric times. That is the first kind of public transportation.</i> Question: <i>What was the first kind of public transportation in history?</i> Answer: <i>ships</i> Reasoning: <i>The pronoun "That" refers to "ships" in the previous sentence.</i>
Text-Connecting	Connecting two explicitly stated components in a text, typically through a noun phrase.	Text snippet: <i>Public transportation is good for the environment. When many people use the same vehicle, fewer cars are on the road. Fewer cars make less pollution.</i> Question: <i>Why is public transportation good for the environment?</i> Answer: <i>Because it causes less pollution</i> Reasoning: <i>"Fewer cars" links to "public transportation" from the previous sentence in a causal relationship.</i>
Gap-Filling	"Incorporating information outside of the text, i.e., general knowledge, with information in the text to fill in missing details." (Cain and Oakhill, 1999, p.490)	Text snippet: <i>White pizza uses no tomato sauce, often substituting pesto or dairy products such as sour cream. Most commonly, its toppings consist only of mozzarella and ricotta cheese drizzled with olive oil and basil and garlic.</i> Question: <i>What is a possible reason "White pizza" gets its name?</i> Answer: <i>It doesn't have tomato sauce</i> Reasoning: <i>Readers need to use common sense to fill in the gap that "no tomato sauce" means the color of the pizza is not red.</i>

Table 1: Taxonomy of inferences for Reading Comprehension questions.

incorporate information from outside of the text with information in the text to fill in some missing details. More examples based on the taxonomy are included in Appendix A.

3.2 Validation of Taxonomy

With the newly developed taxonomy, we annotated the RC items in an in-house item-bank. The item-bank has 192 expert-written multiple-choice RC questions for 24 expository reading passages. These passages vary in difficulty from Grade 3 to Grade 12. Our primary focus was to classify the types of bridging inferences, but we also annotated questions that are not in our main scope of interest. For example, there are some **factual/literal** questions, for which a test taker can directly find information from the text without involving inference; **vocabulary** questions that directly assess the vocabulary knowledge, and other comprehension questions that do not require bridging inferences.

Two of the co-authors classified items independently, following the annotation guideline (see Appendix B). The two coders provided the same coding of the type of inference on 86% of the items, with kappa = 0.83, indicating high agree-

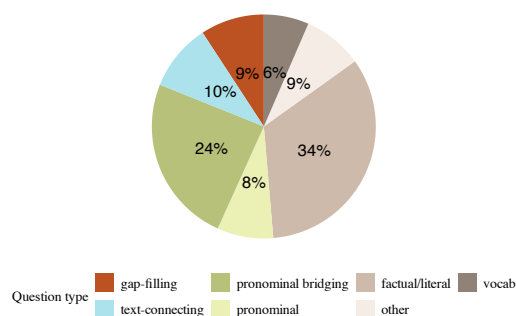


Figure 2: Distribution of different inference types in an operational reading comprehension item bank.

ment. Based on our annotation results shown in Figure 2, we find that bridging inference questions account for 51% of the RC items in the item bank, suggesting bridging inference is an important sub-construct in this RC assessment. Among the bridging inference questions, pronominal bridging (24%) is the most dominant type, followed by text-connecting (10%), gap-filling (9%), and pronominal questions (8%). The high level of agreement supports the validity of the newly developed taxonomy, which we see as an important contribution—

providing a road-map for both item development and future research.

4 Automatic Item Generation and Human Evaluation

Figure 1 presents the overview of our automatic item generation pipeline.

4.1 Training Questions

Due to test security considerations we can not use texts and items from our operational item bank as examples to prompt LLMs. Thus, we created our example item bank which is publicly available for replication efforts². We adapted 6 new expository passages from Simple English Wikipedia³ (passage length ranges from 342 to 508 words, average 438) and for each passage we manually created 2-4 items for each type of inference. Each question contains a **stem**, four **options**, and an answer **key** indicating which option is correct. We also included our thought process in the item generation: **text hint** includes the relevant text from the passage where required inference will be made, and **reasoning** is a short explanation why this question belongs to the requested type. In total, we wrote 19 pronominal bridging, 23 gap-filling, and 16 text-connecting questions.

4.2 Few-shot Prompting

We used the GPT-4o model (2024-04-01-preview) to generate multiple-choice RC questions based on passages we supplied to the model. To prioritize accuracy and reproducibility in item generation, we set the temperature parameter to 0. We explored the frequency penalty parameter from 0 to 0.3, with 0.2 proving optimal as it could consistently generate three diverse RC items without compromising their quality.

Few-shot prompting techniques were used and the prompts were iteratively refined over six rounds. Most adjustments focused on improving the concreteness of the question-writing steps to better guide the model. In this paper, we only report the final iteration of item generation in which we experimented with four different prompting conditions: standard prompting with 4 (or 6) passages and examples, and chain-of-thought prompting with 4 (or 6) passages and examples with text hint and reasoning. With this set-up, we investigated whether

²<https://github.com/maafiah/InferenceQuestionsAQG>

³<https://simple.wikiedia.org>

Task: Given a passage, you are going to generate pronominal bridging inference questions.

Follow these steps to answer the user queries.

Step 1 - find a pronoun (it, they, she, he, which, that, etc) in the passage that is connecting AT LEAST 2 or 3 sentences. The pronoun should be crucial to bridge meaningful information from the passage such as a fact, a cause, a result, or a feature.

Step 2 - based on the pronoun and its reference, generate a multiple-choice question with three distractors. The question should use the pronoun and its reference as hints to connect information between sentences.

Step 3 - follow additional rules when writing the questions: 1) do not ask a question that requires background knowledge to answer. 2) do not ask a question that directly asking "what does XX refer to". 3) lightly paraphrase the question and option without introducing new inference. 4) do not write correct answer longer than the distractors.

Step 4 - iterate this process for 2 times to get 3 different questions.

Step 5 - Output by following the exact format as examples so that it can be directly converted to csv format (do not have any title like (**questions**)). Include all the sentences required in the 'Text Hint' and output your thought process in the 'Reasoning'.

Here are some example passages and example questions:

*****Given passage:*****
A greenhouse is a building where plants such as flowers and vegetables are grown. It usually has a glass ...

*****Examples:*****
 PassageName\Inference Type
 \Text Hint\Reasoning
 \Stem\Option 1\Option 2\Option 3\Option 4\Key
 Greenhouse\pronominal bridging
 \A greenhouse is a building where plants such as flowers and vegetables are grown. It usually has a glass or translucent plastic roof.\the pronoun "it" refers to "greenhouse" in the previous sentence.
 \According to the passage, what can have translucent plastic roofs?\backyards\living spaces\greenhouses \botanic gardens\3
 ...

*****New Passage:*****
Parallax is the perceived change in position of an object seen from two different places ...

Figure 3: **Few-shot prompt for generating pronominal bridging inference questions.** The system prompt (beige background) defines the inference type and outlines expert-inspired steps. Training examples (provided in the prompt) follow. In the standard condition, only the question and answer key (green) are shown; in the CoT condition, text hints and reasoning (blue) are also included. A new passage is provided in the user prompt (orange background) to generate new questions.

increasing the training examples or using the CoT strategy would improve the quality of generation. Moreover, we further evaluated if the output rea-

Criterion	Annotation Guidelines
General item quality	1: If the generated item satisfies all of the following: (a) The correct answer is fully correct; (b) Distractors are not confusing and are clearly incorrect; (c) The question is developmentally appropriate and safe for Grades 3–12. 0: If any requirement is not met. Provide an explanation in the "Note" field.
Inference-type accuracy	1: If the generated item matches the requested inference type. 0: If not. Output inference type, one of: gap-filling / pronominal bridging / text-connecting / factual or literal.
Reasoning quality	1: If the generated thought process fulfills both of the following: (a) The "Reasoning" is adequate and relevant to the requested inference type; (b) The "Text Hint" includes all the sentences required to answer the item correctly. 0: If either condition is not satisfied.

Table 2: Annotation guidelines for evaluating the generated items.

soning process was adequate for this specific task.

Figure 3 shows an example prompt for generating reading comprehension questions targeting pronominal bridging inference (see Appendix C for more details). In the system prompt, we first instructed GPT-4o to identify pronominal bridging relationships, then directed it to generate a multiple-choice question, guided by additional rules to ensure item quality. We included several training examples in the prompt—either 4 or 6 passages with corresponding questions, depending on the generation condition. For the Standard condition, no text hints or reasoning were provided in the training examples. In the CoT condition, both text hints and reasoning were provided, prompting the model to generate them in the output. In the user prompt, we provided a new passage for GTP-4o to generate items from.

We curated a total of 10 new passages adapted from Simple Wikipedia, which were comparable in length and format to the example passages. For each passage and inference type (pronominal bridging, text-connecting, and gap-filling), we independently applied the prompting procedure, instructing GPT-4o to generate three unique questions per combination. For text-connecting and gap-filling—where question construction can be more challenging—we included an additional rule: "Do not force additional questions if no suitable locations can be found." Across the four prompting conditions, we generated a total of 357 questions, 180 of which were produced under the CoT condition and therefore included text hints and reasoning in the output.

4.3 Human Evaluation

To evaluate the quality of the generated RC items, we developed an evaluation rubric (see Table 2). Three authors used items from prior iterations of the generation process and complete several practice rounds and discussion before finalizing the rubric. The rubric is designed to directly address our core research questions:

RQ1: Can LLMs generate high-quality RC items with appropriate distractors suitable for inclusion in an operational item bank?

RQ2: Do the generated RC items align with the requested bridging inference type?

RQ3: How well can LLMs reason about their generation process?

In the evaluation phase, the three authors, who are experts in reading assessment questions, independently annotated all 357 generated items. The agreement was high for general item quality (RQ1), with percent agreement ranging from 87–90%. However, reaching consensus on the inference type (RQ2; 69–70%) and reasoning quality (RQ3; 65–71%) proved more challenging—consistent with prior findings that reasoning-related judgments are inherently difficult to rate (Stasaski et al., 2021).

To address this, we conducted a second round of annotation. In this phase, each rater independently reviewed only the items where their initial rating differed from the other two and decided whether to adjust the rater’s original score. Following this adjustment, inter-rater agreement improved substantially. The final results of percentage agreement and Fleiss’ kappa are shown in Table 3. Our

<p>Requested Type: Gap-filling</p> <p>Text Hint: The main way carbon gets taken out of the atmosphere is by photosynthesis by living organisms.</p> <p>Reasoning: requires common sense to know that photosynthesis is performed by plants.</p> <p>Question: Which organisms play a crucial role in removing CO2 from the atmosphere?</p> <p>Options: animals/ bacteria/ plants/ fossil fuels</p> <p>Key: 3</p> <p>Rating: high quality, correct inference type, and correct reasoning.</p>	<p>Requested Type: Text-Connecting</p> <p>Text Hint: Doughnuts are often eaten in the morning, along with a cup of hot coffee. They are sold at doughnut shops, bakeries, or grocery stores.</p> <p>Reasoning: "doughnuts" and "doughnut shops" are linked thematically."</p> <p>Question: Where can people buy doughnuts?</p> <p>Options: At a coffee shop/ At a doughnut shop/ At a restaurant/ At a candy store</p> <p>Key: 2</p> <p>Rating: low quality because multiple keys can be correct. The question has incorrect inference and reasoning. The question can be categorized as pronominal bridging, as "they" refers to "doughnuts," or as factual/literal, since "doughnut shops" directly refers to places where doughnuts are sold.</p>
---	--

Figure 4: Examples of LLM-generated RC items via Chain-of-Thought prompting with 6 training passages. Left: high-quality; right: low-quality. Each output includes a text hint, a rationale, a multiple-choice question with four options, and an answer key. Human annotations are shown against a beige background.

Criterion	Agreement (%)	Fleiss' κ
General item quality	90–97	0.57
Inference-type accuracy	85–94	0.77
Reasoning quality	90–95	0.83

Table 3: Inter-rater agreement and Fleiss' κ for each evaluation criterion. Agreement is reported as a range based on three pairwise comparisons by three graders.

evaluation in the Results section were based on the majority votes for each item. For example, an item was treated as acceptable when at least two of the three raters rated it as good quality.

5 Results

Based on the proportion of accepted items by generation method (Table 4), we observe improved generation performance when increasing the number of training examples from four to six example passages in the prompt. However, our experiment does not show any clear advantage of Chain-of-Thought prompting over standard few-shot prompting. Furthermore, our results indicate no statistically significant differences in generation performance across the various prompting conditions. We summarize our key findings below.

LLMs can produce high-quality questions suitable for operational use. Based on the evaluation of general item quality, 87 out of 90 questions (96.7% in the CoT_6 condition) had good quality and were suitable for operational use in the Grade 3-12 educational context. The performance is com-

Generation Method	Num Items	General Item Quality	Inference Accuracy	Reasoning Quality
standard_4	88	0.932	0.409	
standard_6	89	0.955	0.461	
CoT_4	90	0.900	0.411	0.356
CoT_6	90	0.967	0.422	0.389
Total	357	0.938	0.426	0.372

Table 4: Proportion of accepted items by generation method—standard vs. chain-of-thought prompting (with text hints and reasoning), using 4 or 6 passages (12–18 examples). Highest scores per criterion are bolded; criteria are defined in Table 2.

parable to, if not better than, those reported in prior research evaluating overall item quality for RC assessments, which ranged from 75% to 90% (Kulshreshtha and Rumshisky, 2022; Uto et al., 2023; Säuberli and Clematide, 2024). Because of the differences between these studies, for a more informative comparison, we encourage future research to replicate our findings under similar conditions. Figure 4 presents one high-quality example and one low-quality example of the generated questions. We find that problems of unacceptable questions included multiple keys, introduction of new vocabulary, confusing wording of the question, etc.

Generating RC questions by specific inference type is a challenging NLP task. Although LLMs can generate high-quality RC items, their ability to produce questions targeting specific inference types remains limited. In the generation method yield-

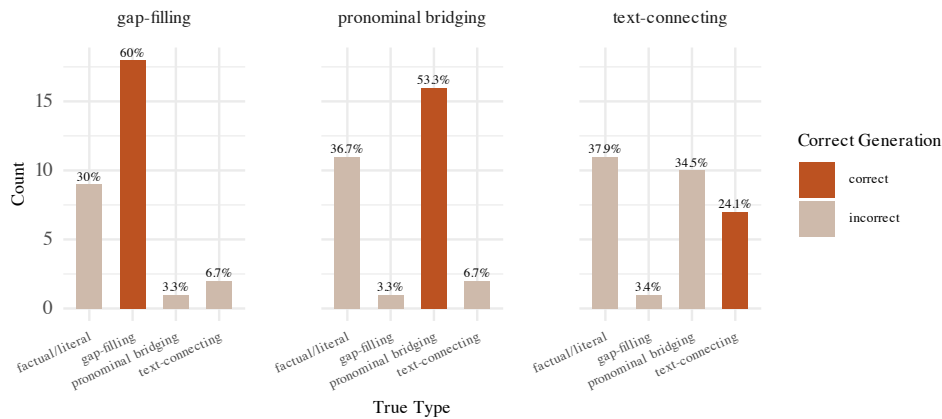


Figure 5: Human evaluation of inference-type accuracy. Each panel displays the distribution of true inference types corresponding to each requested inference type. The generation questions are obtained from the standard few-shot prompting with 6 training passages.

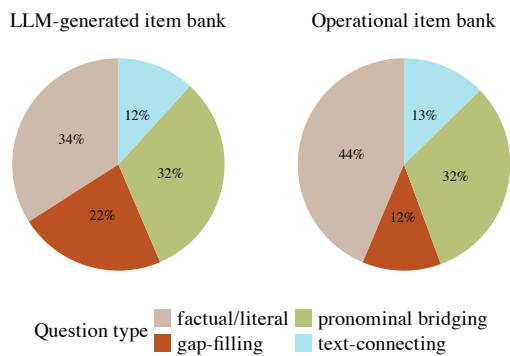


Figure 6: Comparison of item inference type coverage between the operational item bank and the LLM-generated item bank.

ing the best performance (standard_6), only 46.1% of the generated questions matched the requested inference type. As shown in Figure 5, gap-filling questions were the easiest to generate (60% match), followed by pronominal-bridging questions (53.3% match). In contrast, generating text-connecting questions proved particularly difficult, with an accuracy of only 24.1%. This pattern of generation difficulty aligns with the challenges faced by human experts (co-authors) when writing the training examples. We also find that 34.8% of the generated questions were factual or literal, requiring little inference. Moreover, GPT-4o provided adequate reasoning for only 38.9% of the items. This finding may explain the lack of performance gains when moving from standard prompting to CoT prompting. While prior work has shown that adding structured rationales can improve the accuracy of multi-

hop question generation (Säuberli and Clematide, 2024), we believe our task poses a more challenging test of an LLM’s reasoning ability.

Automatic item generation with human evaluation ensures the quality of diagnostic RC items.

From an application standpoint, we also examined how closely the distribution of inference types in the generated items resembled that of human-written items from our operational RC item bank. Interestingly, our analysis, shown in Figure 6, reveals that the overall distribution of inference types in the LLM-generated items closely matches that of our operational RC item bank. This means whereas GPT-4o failed to consistently produce individual items targeting specific inference types, the collect of items it generated somehow resembles the distribution of item types in our existing item pool. With some expert review, most of these items are suitable to use. Understanding the strengths and limitations of current LLM performance is important, particularly if we aim to rely on human evaluation to ensure quality and safety. The generation process is considerably more scalable than relying on human experts to write items manually. Despite current limitations, LLM-based item generation with our newly developed taxonomy offers a promising approach for educational applications.

6 Conclusion

This paper demonstrates our effort in leveraging a large language model to generate inference-making questions for a reading comprehension assessment. We developed a taxonomy of bridging inference questions based on existing literature and validated

it with empirical data from an operational test. The taxonomy focuses on three types of inferences: pronominal bridging, text connecting, and gap-filling. The taxonomy guided our manual creation of example comprehension questions, which were then used as training materials for GPT-4o to generate new items for the new passages. Our evaluation indicates that although GPT-4o can produce acceptable RC questions, its ability to generate questions aligned with specific inference types was limited. This limitation might stem from its limited capability in providing valid reasoning for the types of inferences. These results highlight the critical role of human evaluation when using LLMs for RC question creation. We propose that combining automatic item generation with human judgment offers a promising path toward scalable, high-quality diagnostic RC assessments.

Limitations

We provide preliminary evidence for the potential of GPT-4o in creating inference making reading comprehension questions. The following limitations should be addressed by future research.

We have a limited evaluation set. Our evaluation relies on 10 expository passages (based on Simple Wikipedia), restricting the generalizability of our findings to broader reading contexts or varied educational materials. Future research should incorporate more passages and of different genres, such as narratives.

We exclusively use GPT-4o. This study employed only one LLM, GPT-4o, which may limit insights into the potential effectiveness of other advanced reasoning models. Given the challenge of this reasoning task, future research should explore additional models. Because more advanced models may incur significantly higher costs, future research should also consider the balance between performance and affordability for an educational application.

Unclear effectiveness of Chain-of-Thought prompting. Our results show that generation quality improves with more example questions. However, our experiment does not show benefits from CoT prompting. This unexpected finding may result from our limited number of training examples. Future studies should expand the training data and possibly utilize large datasets, such as SQuAD (Rajpurkar et al., 2016) and FairytaleQA (Xu et al.,

2022). Future work should also explore more effective methods for integrating human-experts' rationales into the question generation process and explore how it affects the reasoning performance of LLMs (Zelikman et al., 2022).

General item quality is a broad metric. Our main goal is to generate RC items that target specific inference types, so we grouped other aspects like answer correctness and distractor plausibility under a broad "General Item Quality" metric. Still, there are important dimensions we didn't separate out—like item difficulty and whether it's appropriate for the target population. More specific metrics could help pinpoint where generation errors happen and how inference type and item difficulty might interact.

Future work should focus on item evaluation in real-world deployment. Our study did not include pilot testing in real-world settings to evaluate how the generated items perform with actual student responses. Student response data would allow for further examination of item bias, difficulty, and discrimination—critical steps before using the items for student scoring and making valid inferences about their abilities (Yeatman et al., 2024). Using LLM-simulated student responses to evaluate generated items is also an exciting direction that could help reduce—but not replace—the need for traditional item calibration (Zelikman et al., 2023; Lu and Wang, 2024; Liu et al., 2025).

Ethics Statement

Our study goal is to leverage LLMs to develop scalable and effective RC assessments to align with educational practice. We introduce a novel and meaningful NLP task: generating RC questions by inference type. While LLMs show promise for item development, we emphasize the importance of maintaining test security by avoiding training models on operational test items, and by ensuring the safety of content such as developmental appropriateness and the absence of problematic materials. In addition to existing automatic benchmarks, human evaluation by educational experts remains essential for item quality. Though beyond our current scope, we also highlight the need for ongoing monitoring of the generated items to detect scoring biases and ensure fairness in operational use.

Acknowledgments

We appreciate the reviewers for their helpful feedback on the manuscript. This study was made possible by the following research grant awarded by the Institute of Education Sciences, U.S. Department of Education, through R305F100005. Opinions, findings, and conclusions in this paper do not necessarily reflect the views of IES or ETS.

References

- Manish Agarwal, Rakshit Shah, and Prashanth Manem. 2011. Automatic question generation using discourse cues. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9.
- Jun Araki, Dheeraj Rajagopal, Sreecharan Sankaranarayanan, Susan Holm, Yukari Yamakawa, and Teruko Mitamura. 2016. Generating questions and multiple-choice answers using semantic analysis of texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1125–1136.
- Nischal Ashok Kumar, Nigel Fernandez, Zichao Wang, and Andrew Lan. 2023. [Improving reading comprehension question generation with data augmentation and overgenerate-and-rank](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 247–259, Toronto, Canada. Association for Computational Linguistics.
- 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.
- Nihat Bayat and Gökhan Çetinkaya. 2020. The relationship between inference skills and reading comprehension. *Education and Science*.
- Claudine Bowyer-Crane and Margaret J Snowling. 2005. Assessing children’s inference generation: What do tests of reading comprehension measure? *British journal of educational psychology*, 75(2):189–201.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Kate Cain. 2022. Children’s reading comprehension difficulties. *The science of reading: A handbook*, pages 298–322.
- Kate Cain and Jane V Oakhill. 1999. Inference making ability and its relation to comprehension failure in young children. *Reading and writing*, 11:489–503.
- Kate Cain, Jane V Oakhill, Marcia A Barnes, and Peter E Bryant. 2001. Comprehension skill, inference-making ability, and their relation to knowledge. *Memory & cognition*, 29(6):850–859.
- Michael Flor and Brian Riordan. 2018. A semantic role-based approach to open-domain automatic question generation. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 254–263.
- Bilal Ghanem, Lauren Lutz Coleman, Julia Rivard Dexter, Spencer von der Ohe, and Alona Fyshe. 2022. [Question generation for reading comprehension assessment by modeling how and what to ask](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2131–2146, Dublin, Ireland. Association for Computational Linguistics.
- Arthur C Graesser, Murray Singer, and Tom Trabasso. 1994. Constructing inferences during narrative text comprehension. *Psychological review*, 101(3):371.
- Susan E Haviland and Herbert H Clark. 1974. What’s new? acquiring new information as a process in comprehension. *Journal of verbal learning and verbal behavior*, 13(5):512–521.
- Yufang Hou. 2020. [Bridging anaphora resolution as question answering](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1428–1438, Online. Association for Computational Linguistics.
- Yufang Hou, Katja Markert, and Michael Strube. 2018. Unrestricted bridging resolution. *Computational Linguistics*, 44(2):237–284.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Kevin Hwang, Kenneth Wang, Maryam Alomair, Fow-Sen Choa, and Lujie Karen Chen. 2024. Towards automated multiple choice question generation and evaluation: aligning with bloom’s taxonomy. In *International Conference on Artificial Intelligence in Education*, pages 389–396. Springer.
- Walter Kintsch. 1998. *Comprehension: A paradigm for cognition*. Cambridge university press.
- Hideo Kobayashi, Yufang Hou, and Vincent Ng. 2022. [End-to-end neural bridging resolution](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 766–778, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Saurabh Kulshreshtha and Anna Rumshisky. 2022. Reasoning circuits: Few-shot multihop question generation with structured rationales. *arXiv preprint arXiv:2211.08466*.

- 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.
- Kunze Li and Yu Zhang. 2024. Planning first, question second: An llm-guided method for controllable question generation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4715–4729.
- 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.
- 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*, pages 16–27.
- Kawshik Manikantan, Makarand Tapaswi, Vineet Gandhi, and Shubham Toshniwal. 2024. [Identifyme: A challenging long-context mention resolution benchmark](#). Preprint, arXiv:2411.07466.
- Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2024. [Multi-hop question answering](#). Preprint, arXiv:2204.09140.
- Nikahat Mulla and Prachi Gharpure. 2023. Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence*, 12:1–32.
- Edward J O’Brien, Anne E Cook, and Robert F Lorch. 2015. *Inferences during reading*. Cambridge University Press.
- Onkar Pandit and Yufang Hou. 2021. [Probing for bridging inference in transformer language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4153–4163, Online. Association for Computational Linguistics.
- 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.
- Yin Poon, John Sie Yuen Lee, Yu Yan Lam, Wing Lam Suen, Elsie Li Chen Ong, and Samuel Kai Wah Chu. 2024. [Few-shot question generation for reading comprehension](#). In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 21–27, Bangkok, Thailand. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Manav Rathod, Tony Tu, and Katherine Stasaski. 2022. [Educational multi-question generation for reading comprehension](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 216–223, Seattle, Washington. Association for Computational Linguistics.
- Ina Rösiger. 2018. [BASHI: A corpus of Wall Street Journal articles annotated with bridging links](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- John Sabatini, Jonathan Weeks, Tenaha O’Reilly, Kelly Bruce, Jonathan Steinberg, and Szu-Fu Chao. 2019. SARA Reading Components Tests, RISE forms: Technical Adequacy and Test Design. *ETS Research Report Series*, 2019(1):1–30.
- 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, Torino, Italia. ELRA and ICCL.
- Franz Schmalhofer, Mark A McDaniel, and Dennis Keefe. 2002. A unified model for predictive and bridging inferences. *Discourse Processes*, 33(2):105–132.
- Murray Singer, Peter Andruslak, Paul Reisdorf, and Nancy L Black. 1992. Individual differences in bridging inference processes. *Memory & cognition*, 20(5):539–548.
- Murray Singer and Gilbert Remillard. 2004. Retrieving text inferences: Controlled and automatic influences. *Memory & Cognition*, 32(8):1223–1237.
- Katherine Stasaski, Manav Rathod, Tony Tu, Yunfang Xiao, and Marti A Hearst. 2021. Automatically generating cause-and-effect questions from passages. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 158–170.
- Richard Thurlow and Paul van den Broek. 1997. Automaticity and inference generation during reading comprehension. *Reading & Writing Quarterly: Overcoming Learning Difficulties*, 13(2):165–181.
- 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.

- Paul van den Broek, Katinka Beker, and Marja Oudega. 2015. Inference generation in text comprehension: Automatic and strategic processes in the construction of a mental representation. In Edward J. O'Brien, Anne E. Cook, and Robert F. Lorch Jr., editors, *Inferences during reading*, pages 94–121. Cambridge University Press.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, et al. 2022. Fantastic questions and where to find them: Fairytaleqa—an authentic dataset for narrative comprehension. *arXiv preprint arXiv:2203.13947*.
- Jason D Yeatman, Jasmine E Tran, Amy K Burkhardt, Wanjing Anya Ma, Jamie L Mitchell, Maya Yablonski, Liesbeth Gijbels, Carrie Townley-Flores, and Adam Richie-Halford. 2024. Development and validation of a rapid and precise online sentence reading efficiency assessment. In *Frontiers in education*, volume 9, page 1494431. Frontiers Media SA.
- Eric Zelikman, Wanjing Ma, Jasmine Tran, Diyi Yang, Jason Yeatman, and Nick Haber. 2023. [Generating and evaluating tests for k-12 students with language model simulations: A case study on sentence reading efficiency](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2190–2205, Singapore. Association for Computational Linguistics.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

A Bridging Inference Examples

When we developed the taxonomy of bridging inference, we referred to a sample passage and a list of example questions provided from [Cain and Oakhill \(1999, p.495\)](#). Table 5 presents our analysis of the given questions based on the taxonomy.

B Annotation Guidelines

To validate the newly developed taxonomy of bridging inference questions, we annotated an in-house RC item bank. Annotation was done with regards to the text and the questions including stem, key and distractors (see details in Table 6).

C Prompts

We present examples of our few-shot prompting design for pronominal bridging (Figure 3) text-connecting (Figure 7) and gap-filling (Figure 8) respectively. The rules are identical for both the standard and CoT prompts; the only difference is that CoT includes a text hint and reasoning in the training examples (see blue highlight in the figure). Accordingly, in the CoT condition, we expect the output to include a text hint and reasoning along with the generated questions.

Reading Passage:

Debbie was going out for the afternoon with her friend Michael. By the time they got there they were very thirsty. Michael got some drink out of his duffel bag and they shared that. The orange juice was very refreshing. Debbie put on her swimming costume, but the water was too cold to paddle in, so they made sandcastles instead.


They played all afternoon and didn't notice how late it was. Then Debbie spotted the clock on the pier. If she was late for dinner, her parents would be angry. They quickly packed up their things. Debbie changed and wrapped her swimming costume in her towel. She put the bundle in her rucksack. Then they set off for home, pedalling as fast as they could. Debbie was very tired when she got home, but she was just in time for dinner.

Question	Annotation
Literal information	
Who did Debbie spend the afternoon with?	The answer is in the first sentence. There is a partial paraphrase: "going out for" vs. "spend".
Where was the clock?	The answer is in the second sentence of the second paragraph.
Text-connecting inference	
Where did Michael get the orange juice from?	This requires bridging inference: <i>drink = orange_juice</i> . This is both a referential and semantic link (hypernym: drink – hyponym: juice). Recognizing this link requires background knowledge and both components are near each other in the text.
Where did Debbie put her towel when she packed up her things?	The answer is in sentences 5–6 of the second paragraph. This involves recognizing a part-whole relationship (towel–bundle), which is an ad-hoc, situational reference.
Gap-filling inference	
Where did Debbie and Michael spend the afternoon?	One component (afternoon) is in the text, but the location (the beach) is not. It must be inferred as a plausible missing piece of the situation model.
How did Debbie and Michael travel home?	The text says "set off for home" (a paraphrase of "travel"). The mode of travel is inferred from "pedalled", enriching the situation model.

Table 5: Analysis of a reading passage and associated reading comprehension questions with inference annotations. The passage and questions are adapted from Cain and Oakhill (1999, p.495).

Dimension	Options	Note
Inference	Factual / Literal	The answer is explicitly stated in the text, exactly matching the question. No inference needed.
	Pronominal	Resolving pronouns (e.g., "Who does 'he' refer to?").
	Pronominal Bridging	Requires resolving a pronoun and using it as a cue to infer the correct answer.
	Text-Connecting	Requires connecting two explicitly stated components, typically using noun phrases.
	Gap-Filling	Involves filling in a missing but easily inferred piece of information not directly stated in the text.
	Vocabulary	Tests the reader's knowledge of word meanings.
	Other	Any other type, such as comparison or author intent.

Table 6: Annotation guidelines for the in-house item bank.



Task: Given a passage, you are going to generate text-connecting inference questions.

Follow these steps to answer the user queries.

Step 1 - find two concepts (primarily nouns or noun phrases) that are connecting AT LEAST 2 or 3 sentences, but their relationship is not explicitly stated.
Please follow the rules:

- The two concepts should not contain any same word. Incorrect example: "the Ocean" and "Pacific Ocean" share a word "Ocean". Correct example: "flowers" and "rose".
- The two concepts should only exist in two different sentences.
- The second concept should not be a pronoun that explicitly refers to the first concept.
- there are different possible subtypes of text-connecting you may find from the passage:

Subtype 1: Coreference without a pronoun nor repetition (share word): This refers to instances where two or three sentences are linked together by two noun phrases in the passage that refer to the same real-world entity. Correct examples: "boys and girls" referring to "students" from the previous sentence, "manager" referring to the "CEO" from the previous sentence. Incorrect examples: "he" referring to "John" (as "he" is a pronoun), "the show" referring to "TV show" (because this is a repetition and they share the word "show", unless there is more than one show described in the passage).

Subtype 2: Whole-to-part relation. For instance, "mom" refers to "parent", "bride" can refer to the "wedding" from the previous sentence, and "walls" can refer to the "construction project" mentioned earlier.

Subtype 3: implicit causal relation without a clue word

Subtype 4: events happen in the same time, etc.

Step 2 - based on two concepts you have identified, generate a multiple-choice question with three distractors. The question should use the relationship between the two concepts as a hint to connect information between sentences.

Step 3 - follow additional rules when writing the questions: 1) do not ask a question that requires extra background knowledge beyond this identified text-connecting relationship to answer. 2) do not ask a question that directly asking "what does XX refer to". 3) lightly paraphrase the question and option without introducing new inference. 4) do not write correct answer longer than the distractors.

Step 4 - iterate this process for 2 times to get 3 different questions. Do not force to generate more questions if you cannot find more places.

Step 5 - Output by following the exact format as examples so that it can be directly converted to csv format (do not have any title like (**questions**)).

Here are some example passages and example questions:

*****Given passage:*****
A greenhouse is a building where plants such as flowers and vegetables are grown. It usually has a glass ...




*****Examples:*****
 PassageName\Inference Type
 \Text Hint\Reasoning
 \Stem\Option 1\Option 2\Option 3\Option 4\Key

Greenhouse\text-connecting bridging

\Many vegetables and flowers are grown in greenhouses in late winter and early spring, when it is still too cold to grow plants outside. Then these plants move into the soil outside as the weather warms up.

\""these plants"" links to ""many vegetables and flowers"" as a part to whole relation in the previous sentence."


\When do greenhouse vegetables and flowers move into the soil outside?\when the weather warms up\when heating is not working\in early spring\when there is no rain\1..

*****New Passage:*****
Parallax is the perceived change in position of an object seen from two different places ...

Figure 7: Few-shot prompting using Chain-of-Thought for generating text-connecting inference.

Task: Given a passage, you are going to generate gap filling inference questions. This question asks for a piece of information outside of the text, i.e. general knowledge, with information in the text to fill in missing details in the passage.



Follow these steps to answer the user queries.

Step 1 - Find a concept in the passage that you think general background knowledge will be required to comprehend the text. There are three possible subtypes:
 Subtype 1: two or three sentences are connected without a pronoun but by a common sense that is not stated in the passage.
 Subtype 2: infer the result from a given situation based on a stated causal relationship. for example :The passage implies that if, then _____. The result should not appear in the passage.
 Subtype 3: to give an example based on the characteristics inferred from the text. for example: Which of the following could be an example of _____. Note that the example should not appear in the passage.

Step 2 - generate a multiple-choice question with three distractors.


Step 3 - follow additional rules when writing the questions: 1) do not ask a question that can be directly answered from the passage. 2) do not ask a question that directly asking "what does XX refer to". 3) do not write correct answer longer than the distractors. 4) the distractors should be incorrect and should not be confusing.

Step 4 - iterate this process for 2 times to get 3 different questions. Do not force to generate more questions if you cannot find more places. You don't need to generate each subtype.

Step 5 - Output by following the exact format as examples so that it can be directly converted to csv format (do not have any title like (**questions**)).


Here are some example passages and example questions:

*****Given passage:*****
A greenhouse is a building where plants such as flowers and vegetables are grown. It usually has a glass ...




*****Examples:*****
 PassageName\Inference Type
 \Text Hint\Reasoning
 \Stem\Option 1\Option 2\Option 3\Option 4\Key

Greenhouse\Gap-filling
 \Also, greenhouses can get very hot from the sun's heat, so gardeners have to make sure that it does not get too hot for the plants.
 Greenhouses usually have vents that can be opened to let excess heat out. Some greenhouses have electric exhaust fans that automatically turn on if it gets too hot in the greenhouse. A greenhouse is the place for tender plants such as tomatoes, cucumbers, and aubergines.
 \Infer the result from a given situation based on a stated causal relationship



\What is likely to happen if a greenhouse fails to control the heat in summer?\The greenhouse will grow more plants.\The greenhouse will become smaller.\Tender plants inside the greenhouse will not grow well.\Less gardeners will be needed to water the plants.\3



*****New Passage:*****
Parallax is the perceived change in position of an object seen from two different places ...

Figure 8: Few-shot prompting using Chain-of-Thought for generating gap-filling inference.

Investigating Methods for Mapping Learning Objectives to Bloom’s Revised Taxonomy in Course Descriptions for Higher Education

Zahra Kolagar

Technische Hochschule Augsburg, Germany
zahra.kolagar@tha.de

Frank Zalkow

Fraunhofer IIS, Erlangen, Germany
frank.zalkow@iis.fraunhofer.de

Alessandra Zarcone

Fraunhofer IIS, Erlangen, Germany
Technische Hochschule Augsburg, Germany
alessandra.zarcone@tha.de

Abstract

Aligning Learning Objectives (LOs) in course descriptions with educational frameworks such as Bloom’s revised taxonomy is an important step in maintaining educational quality, yet it remains a challenging and often manual task. With the growing availability of large language models (LLMs), a natural question arises: can these models meaningfully automate LO classification, or are non-LLM methods still sufficient? In this work, we systematically compare LLM- and non-LLM-based methods for mapping LOs to Bloom’s taxonomy levels, using expert annotations as the gold standard. LLM-based methods consistently outperform non-LLM methods and offer more balanced distributions across taxonomy levels. Moreover, contrary to common concerns, we do not observe significant biases (e.g. verbosity or positional) or notable sensitivity to prompt structure in LLM outputs. Our results suggest that a more consistent and precise formulation of LOs, along with improved methods, could support both automated and expert-driven efforts to better align LOs with taxonomy levels.

1 Introduction and Motivation

Learning Objectives (LOs) define the knowledge and competencies students are expected to acquire through educational activities, for example: “By the end of this course, students will be able to identify examples of symbolism in short stories and incorporate symbolism in their writing” (from the description of the course of literary studies). These objectives provide a clear and measurable framework for educators to evaluate student progress and align course instruction with desired learning outcomes (Mager and Peatt, 1962; Rodriguez and Albano, 2017; Fink, 2003).

LOs are articulated in course descriptions, which outline instructional activities, intended outcomes, and assessment methods for the course. The development of LOs follows the “Theory of Constructive

Alignment” (Biggs, 1996), ensuring that teaching and assessment are directly aligned with the LOs. This alignment allows educators to create a coherent structure where every aspect of the course is designed to support students in achieving the desired outcomes (Wang et al., 2013b; Jaiswal, 2019).

Among various educational frameworks used for constructive alignment, Benjamin Bloom’s taxonomy (Bloom et al., 1956), later revised by Anderson and Krathwohl (2001), is widely recognized in higher education to guide the development and assessment of LOs mentioned in the course description. The revised version defines six hierarchical cognitive levels—Remember, Understand, Apply, Analyze, Evaluate, and Create—which serve as a guide for developing and assessing LOs. Bloom’s taxonomy provides a structured approach to categorizing LOs and ensures that they are appropriately mapped to cognitive levels and aligned with the intended educational goals (Arafeh, 2016; Dubicki, 2019). Furthermore, it facilitates the alignment of classroom assignments and exams with the intended cognitive levels (Sterz et al., 2019; Biggs et al., 2022).

The mapping of LOs and Bloom’s taxonomy levels is performed by educators, curriculum designers, and assessment centers as part of quality assurance processes, such as course accreditation (Randhahn and Niedermeier, 2017; Kultusministerkonferenz, 2017). However, manual LO mapping can be time consuming, labor intensive, and error-prone (Biggs, 1996; Reeves and Hedberg, 2003; Hussey and Smith, 2008). Large language models (LLMs) have shown promising capabilities in similar tasks, such as data annotation and classification (see e.g., Tan et al., 2024b) that offer promising potential to automate this process (Wang et al., 2024; Xu et al., 2024). Yet, their reliability and robustness remain open questions. In particular, they can be sensitive to prompt formulation and other design choices, and exhibit bias such as position bias, where out-

puts are influenced by their placement in a list, and verbosity bias, where longer responses are favored, or a tendency to generate rationales that align with previously provided labels, which may affect the reliability of their outputs (Shen et al., 2023; Koo et al., 2023; Wu and Aji, 2023; Stureborg et al., 2024; Chen et al., 2024; Tan et al., 2024a; Choshen et al., 2024). Moreover, it remains unclear whether LLMs offer a substantial advantage over non-LLM methods in this setting, or whether simpler, more cost-effective methods may suffice.

This work investigates the effectiveness of both LLM- and non-LLM-based techniques for automating LO-to-taxonomy mapping, and examines how prompt and task design influence LLM behavior. The key research questions are:

- RQ 1: How do LLM-based and non-LLM methods compare in effectiveness when mapping LOs to Bloom’s taxonomy levels?
- RQ 2: To what extent do experiment design choices influence the performance of LLM in mapping LOs, and do these variations reflect model bias or sensitivity to task framing?

2 Background and Related Work

2.1 Bloom’s Revised Taxonomy

Bloom’s cognitive process dimension defines six ascending levels of complexity: (Anderson and Krathwohl, 2001). **Remembering** involves recalling or recognizing knowledge from memory, such as definitions or facts. **Understanding** entails constructing meaning by interpreting, summarizing, and explaining information. **Applying** involves using learned material in new situations, often through models or simulations. **Analyzing** requires breaking concepts down into parts to understand their relationships. **Evaluating** involves making judgments based on criteria, exemplified by critiques or recommendations. Lastly, **Creating** is about generating new ideas or products by reorganizing elements in innovative ways, making it the most complex cognitive process. Each taxonomy level comes with a selection of verbs that define the expected learning outcomes. Examples can be found in Table 3 in the Appendix.

2.2 Pre-LLM Approaches to LO Mapping

Before LLMs, researchers explored methods such as keyword dictionaries (Chang and Chung, 2009), TF-IDF-based classifiers (Echeverría et al., 2013),

and supervised machine learning models (Waheed et al., 2021; Mohammed and Omar, 2020). Most of these efforts focused on short texts such as exam or discussion questions, and while models showed promise at lower cognitive levels like “Remember,” performance dropped significantly for higher-order categories. A notable large-scale study by Li et al. (2022) introduced a dataset of over 21,000 manually labeled learning objectives and evaluated both traditional and BERT-based classifiers, reporting strong performance but relying on single-skill LOs.

2.3 LLMs for LO Mapping & Alignment in the Educational Domain

LLMs are increasingly being integrated into educational contexts. Research assessing GPT-4’s mastery according to Bloom’s taxonomy in answering psychosomatic medicine exam questions demonstrated that while the model yielded an average score of 92 % in high-order cognitive levels, it still encounters difficulties at low-order cognitive levels such as “Remember” and “Understand,” where it sometimes fails to recall specific details or correctly interpret conceptual relationships (Herrmann-Werner et al., 2024).

Al Ghazali et al. (2024) conducted a case study examining ChatGPT’s effectiveness in teaching chemistry to eleventh-grade students, employing Bloom’s taxonomy to categorize LOs and evaluate student performance in answering course-related questions. They found that, although the model performed well in knowledge recall and reasoning skills, it struggled with maintaining student engagement and achieving comparable outcomes to traditional teaching methods. Meanwhile, Maity et al. (2024) evaluated the efficacy of GPT-4 Turbo in generating educational questions aligned with Bloom’s taxonomy, revealing that while the model can generate questions for high-order thinking skills, its effectiveness varies between different cognitive levels, and the model demonstrates difficulties in crafting high-quality questions at more advanced taxonomy levels, such as “Create”.

Our task of mapping LOs to Bloom’s taxonomy is a multi-label classification problem. However, unlike standard classification tasks typically addressed with LLMs (see, e.g., Niraula et al., 2024; Reddy et al., 2024; Li et al., 2024), our problem poses unique challenges that go beyond standard tasks. While classification tasks typically rely on detecting surface-level features or patterns in the text, Bloom’s taxonomy requires an in-depth se-

semantic understanding of the cognitive processes implied by the LO. For example, distinguishing between “Understanding” and “Applying” involves subtle differences in the LO’s intents, such as whether the task involves interpreting information versus using it in a new context. Furthermore, the hierarchical nature of Bloom’s taxonomy adds an additional layer of complexity, as higher-order categories (e.g., “Evaluating” or “Creating”) often overlap with or build upon lower-order processes. This requires not only a fine-grained contextual analysis, but also a deep understanding of the underlying pedagogical framework.

3 Task and Evaluation

To create a gold standard dataset of LOs mapped to the corresponding levels of Bloom taxonomy, we collected LOs from university course descriptions. Given an LO like “Students should be able to recite the key principles of Newton’s laws of motion and analyze a given set of data to determine how well it demonstrates Newton’s laws in action” (from a Physics course), experts in pedagogy might map this LO to both “Remember” and “Analyze” levels (the data collection is described in Section 4). These expert mappings were used to create the gold standard dataset, and we report Krippendorff’s α (Krippendorff, 2004) to measure agreement.

We then evaluate the reliability of automatic methods in producing similar LO mappings as the experts, using both non-LLM (as baseline) and LLM-based methods. We compare the results from both LLM and non-LLM methods against the gold standard annotations and report the weighted F1 score as well as the different frequency distributions for each taxonomy level produced by the different methods.

The evaluation of LLM-based methods was additionally aimed at testing their robustness. We therefore present the LLMs with different formulations of the task to examine whether any biases manifest during the LO mapping process. With this goal, we compute agreement and correlations between the answers provided by LLM-based methods, model confidence (by analyzing the log probabilities retrieved from the model), and semantic similarity measures between the models’ generated rationales to assess their consistency.

Subject	No. Courses
Introduction to Psychology	4
Gerontology	4
Ancient Greek History & Literature	2
Literary Theory	6
Climate Change	4
Microeconomics	4
Introduction to Linguistics	4
Introduction to Anthropology	2
Animal Behaviorism	1
Blockchain	2
Political Philosophy	2

Table 1: Overview of collected course descriptions across various academic subjects. Each course description contains one learning objective section.

4 Data Acquisition

4.1 Data Collection and Preprocessing

We collected a total of 35 LOs from course descriptions¹ from the websites of German universities, comprising 25 bachelor-level and 10 master-level course descriptions. These descriptions present a diverse range of academic subjects and degree levels, as shown in Table 1. Even though the language of instruction was English, some of the course descriptions were only available in German.

We focus on the “learning objective” section of the course descriptions, which also exist internationally under different names such as “learning outcomes” or “course objectives”. We translated the course objectives from German into English using the DeepL API² and asked a bilingual person to revise them to ensure the correctness of the translations. The pre-processing of the collected data involved basic text-cleaning tasks to ensure consistent formatting.

4.2 Expert Annotation

We recruited five experts in higher education pedagogy to annotate the LOs. Each expert was provided with a combination of course titles and the corresponding LOs, along with the six levels of Bloom’s taxonomy. Their task was to identify and select all relevant taxonomy levels as shown in Figure 1 (full task instructions are reported in Figure 3 in the Appendix). We collected demographic information to evaluate the participants’ expertise and familiarity with Bloom’s taxonomy. All experts reported a high level of familiarity, with one with

¹The dataset including the 35 LOs and their annotations will be publicly released to support further research in this domain.

²<https://www.deepl.com/>

1–3 years of experience, two having 3–6 years of experience, and two having more than 6 years of experience.

Course Title: Greek History

Learning Objective:
Students will gain an overview of Greek history. After completing the module, they will be able to examine the sources and evidence from this period, place them in a wider historical context, and evaluate them.

Question: Which taxonomy levels are relevant to the given learning objective?

- Remember
- Understand
- Apply
- Analyze
- Evaluate
- Create

Figure 1: Sample task presented to expert annotators from the questionnaire.

We report Krippendorff’s α as a measure of inter-annotator agreement, calculated across all Bloom’s taxonomy levels for the entire set of LOs. Given one LO and a pair of annotators, we define as cases of agreement for each level all cases where both annotators either selected that level or did not select it, and as cases of disagreement all cases where one annotator did select that level but the other one did not. We obtained an α of 0.76. While this reflects a reasonably high level of agreement, one annotator noted a challenge with certain LOs:

“Many LOs focus more on the learning process itself (e.g., imparting foundational knowledge) rather than describing the competencies students should have achieved by the end of the learning unit. As a result, some of the commonly used verbs were not applied, making it somewhat difficult for me to classify them within the taxonomy levels. I was also uncertain about how to categorize the verb ‘reflect’.”

This issue is evident in the example, “Knowledge of basic literary categories and methods of interpretation along with a familiarity with fundamental questions of Greek literary history to deal critically with scientific questions and present their own

scientific results.” (Greek Literature I). While all annotators agreed on “Remember”, four selected “Analyze” and “Create”, three selected “Evaluate”, two selected “Understand” and “Apply”. For comparison, a literature-related LO from the dataset introduced by Li et al. (2022) reads: “A basic understanding of the main periods, styles, genres, intellectual preoccupations and socio-historical trends in German literature from the late eighteenth century to the early nineteenth century.” This was labeled as “Understand”, highlighting a key distinction that; although small in quantity, our dataset includes more abstract, multi-layered objectives that often span multiple levels of Bloom’s taxonomy, even challenging experts to reach consensus.

Finally, to create the gold standard annotations, we selected for each LO the taxonomy levels where at least three annotators agreed on a taxonomy level. The distribution of selected taxonomy levels in the gold standard labels is shown in Figure 2.

5 Automatic Methods for Mapping LOs to Bloom’s Taxonomy Levels

5.1 Non-LLM Mapping Methods

For non-LLM methods, we made use of regular expressions (regex), fuzzy matching³, the SpaCy library (Honnibal et al., 2020), and semantic similarity. We used Bloom’s identified set of measurable verbs that are linked to each taxonomy level to help the LO mapping process (Bloom et al., 1956; Anderson and Krathwohl, 2001), as shown in Table 3 in Section A of the appendix.

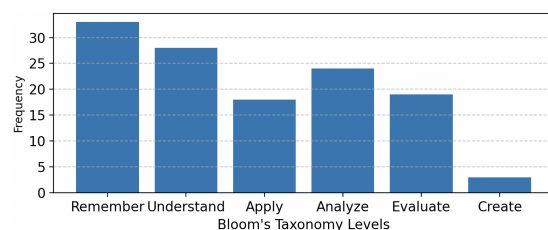


Figure 2: Distribution of gold standard annotations across taxonomy levels.

As an initial step, we applied **regex** and **fuzzy matching** to perform simple string matching of these verbs to their corresponding taxonomy levels. Every subsequent step aimed to address the limitations of the previous approach. Next, we used **spaCy**’s Part-of-Speech (POS)⁴ tagging and

³<https://pypi.org/project/fuzzywuzzy/>

⁴<https://spacy.io/usage/linguisticfeatures#pos-tagging>

dependency parsing⁵ capabilities. We began by segmenting the LOs into smaller sentence fragments using spaCy’s Sentencizer⁶, which produced **159 segments** from the 35 collected LOs. These segments were not only used for spaCy and semantic similarity methods but also all LLM-based approaches. POS tagging was applied to identify verbs in each segment, while dependency parsing provided additional grammatical context, improving the accuracy of verb identification by analyzing sentence structures. Following this, we applied regex to match the identified verbs against a pre-defined list of Bloom’s Taxonomy verbs for LO mapping.

Finally, we used **semantic similarity techniques**, comparing LOs directly with detailed descriptions of Bloom’s taxonomy levels, as outlined in Anderson and Krathwohl (2001), instead of relying solely on verb lists. The Sentence-BERT model (Reimers and Gurevych, 2019)⁷ was employed to measure semantic similarity between LO segments and the taxonomy-level descriptions. For each segment, the model calculated similarity scores to determine the best match.

5.2 LLM Mapping Methods

We utilized OpenAI’s GPT-4 model⁸ (OpenAI, 2023) and treated the model as an annotator. To prompt the model, we were inspired by tasks typically presented to human annotators, including multiple choice selection, pairwise comparison, best-worst scaling, binary annotation, ranking, and rating (Wang et al., 2013a; Bragg et al., 2018; Huynh et al., 2021). We presented a variety of tasks: multiple choice selection with paraphrase prompting and rationale generation (MCS), pair-wise comparison (PWC), best-worst scaling analysis (BWS) (Cohen, 2003; Louviere et al., 2015), binary annotation using a yes/no check with confidence analysis (BCA), and rating using point-wise relevance rating with confidence analysis (RCA), which is described in the subsequent paragraphs. Refer to Appendix Section D for the prompts used for each method.

To perform **MCS**, we collected paraphrases of Bloom’s taxonomy levels from educational resources (see Appendix Section C), resulting in four

⁵<https://spacy.io/usage/linguisticfeatures#dependency-parse>

⁶<https://spacy.io/api/sentencizer>

⁷https://www.sbert.net/docs/usage/semantic_textual_similarity.html

⁸The temperature was set to zero for all LLM-based methods.

paraphrased versions for each level in addition to the original descriptions from Anderson and Krathwohl (2001). Paraphrases aimed at ensuring that the model’s selection was guided by the conceptual meaning of each taxonomy level, rather than the specific phrasing of the taxonomy descriptions. We applied MCS to the segmented LOs described in Section 5.1, providing each segment along with the course title and one paraphrased version of Bloom’s taxonomy level descriptions for each level. The model was prompted to select the relevant category out of the six taxonomy levels, and their descriptions were provided as choices. We also asked the model to provide a rationale, with the specific prompt sequence varying according to the conditions outlined below:

- **Condition A:** isolates the task of rationale generation and the multiple-choice selection of the relevant taxonomy levels.
- **Condition B:** The model first generated a rationale and then selected the relevant taxonomy levels based on that rationale.
- **Condition C:** The model was prompted to choose relevant taxonomy levels first and then generate a rationale based on its choices.

We employed the same prompt for all conditions but altered the task sequence in Conditions B and C, and separated rationale generation from multiple-choice selection in Condition A. Then, we collected and normalized responses for each condition, removing any non-relevant values. Segments of each LO were aggregated back into the original LO, compiling selected taxonomy levels into a list with removed repetitions.

Moreover, we compared PWC and BWS results. **PWC** involved prompting the model to choose from two taxonomy levels—which could still be influenced by position bias, despite our efforts to mitigate it by varying the sequence. We generated unique pairs of taxonomy levels combined with segments from the LOs. With 6 taxonomy levels, we created 30 unique pairs (15 [A,B] and 15 [B,A] pairs) for all 159 segments, leading to 4770 pairs for evaluation. For **BWS**, we created 3-tuples from the 6 taxonomy levels, resulting in 20 unique 3-tuples per segment and 3180 distinct 3-tuples in total. We prompted the LLM to select the most and least relevant taxonomy level from the tuple. For both methods, scores were calculated based on

Category	Method	F1 score
non-LLM	regex	0.52
	fuzzy matching	0.54
	spaCy	0.45
	semantic similarity	0.50
LLM	MCS (condition A)	0.67
	MCS (condition B)	0.68
	MCS (condition C)	0.68
	PWC	0.60
	BWS	0.69
	BCA	0.66
	RCA (short)	0.66
RCA (long)	0.68	

Table 2: Weighted F1 scores for non-LLM and LLM methods.

the frequency of choices. We then identified the highest-scoring taxonomy level as the most relevant for each segment and aggregated across sentences to determine the most relevant levels for each LO.

Finally, the **BCA** method involved a binary relevance evaluation of each taxonomy level for the LO segments, whereas the **RCA** method required the model to rate the relevance of each taxonomy level on a scale from 1 (least relevant) to 5 (most relevant). Additionally for both methods, we estimated the model’s confidence in its decision by collecting log probabilities from the “logprobs” parameter of OpenAI’s Chat Completions API⁹. By calculating linear probabilities from these logprobs, we evaluated the model’s confidence levels, with higher scores indicating greater confidence. For BCA, we only collected the taxonomy levels where the linear probability was over 90 % for “Yes” answers.

To investigate verbosity bias in the RCA task, we calculated the number of tokens in the five paraphrases using spaCy’s tokenizer¹⁰. We identified the longest (1014 tokens) and shortest (775 tokens) paraphrases and prompted the model to rate the taxonomy levels. Logprobs were collected for both rating rounds to assess the impact of paraphrase length on ratings. For this analysis, we only considered taxonomy levels for which the model gave a rating of “5 (most related)”.

6 Results

6.1 Comparison with Expert Annotations

Weighted F1 scores for non-LLM and LLM methods are presented in Table 2. An example of the mapping result can be found in Figure 11 and Table

⁹https://cookbook.openai.com/examples/using_logprobs

¹⁰<https://spacy.io/api/tokenizer>

4 in the appendix.

Non-LLM Methods: As a first comparison with the gold standard, we compared the frequency of the selected taxonomy levels (Figure 12 in the Appendix). This frequency analysis shows a greater consensus between the human annotation and the other methods only for the “Evaluate” level, with high variability in other categories. This could be attributed to the fact that evaluation often involves more objective criteria and well-defined standards, such as assessing the validity of arguments or the accuracy of conclusions, which are less prone to interpretation compared to other taxonomy levels. Across all methods, “Apply” is the most frequently selected taxonomy level, while in general, the results show a large variance between the methods.

Regex and fuzzy matching achieved slightly higher F1 scores (0.52 and 0.54) than spaCy (0.45) due to their wider word capture, including nouns and adjectives, which inflates word frequency and taxonomy levels. spaCy, which focuses on verbs, is more selective and thus may miss some verbs, resulting in fewer mappings and lower F1 scores. The semantic similarity method (F1 = 0.50) offers flexible matching by emphasizing descriptions but can be less precise, leading to skewed results compared to human annotations.

LLM Methods: The F1 scores for LLM methods demonstrate better performance than non-LLM methods with BWS achieving the highest F1 score (0.69). The observed improvements highlight the potential of LLM-based approaches but also emphasize the need for deeper investigation into their consistency and reliability.

The frequency analysis (Figure 13 in the Appendix) reveals a more uniform distribution of taxonomy levels across the LLM methods when compared to non-LLM methods. However, when compared to the gold standard, LLM methods show a higher frequency of taxonomy levels across most categories. The exception is the “Remember” level, where the gold standard annotations have a higher value, though the difference is not substantial. Conversely, the “Create” level exhibits a significant variation: the gold standard has a markedly lower frequency (3) compared to LLM methods (Avg. 29). This indicates a notable discrepancy in how “Create” is represented in the gold standard versus the other methods.

The frequency distributions for the different methods are reported in Figures 14 (MCS), 15

(PWC), 16–17 (BWS) in the Appendix.

6.2 Consistency (LLM-Methods only)

MCS For the MCS method, we were interested in evaluating how consistent the rationales produced by the model were across different conditions and paraphrases of the taxonomy levels.

We used Sentence-BERT to calculate semantic similarity scores for the rationales provided for each learning objective, comparing across various paraphrases. The overall similarity score across all paraphrases was 0.92, with Condition A achieving a score of 0.94, while Conditions B and C each scored 0.92. These results indicate that the paraphrased wording has minimal influence on the outcomes, suggesting almost no bias for all conditions and paraphrases. For more details, refer to Table 5 in the Appendix.

PWC For the PWC method, we wanted to evaluate if the model showed a preference for levels in a specific position. We observed an intra-pair consistency of 86.79 % between the two different versions of the same item with a Cohen’s Kappa (Cohen, 1960) of 0.84. The model’s choices were categorized into three types:

- **“Left”**: The model selected the taxonomy level presented first in the pair.
- **“Right”**: The model selected the taxonomy level presented second in the pair.
- **“None”**: The model either did not provide a clear selection or returned a taxonomy level that did not match any of the expected options.

Among 159 segments, the distribution between “Left” (153 instances) and “Right” (154 instances) was nearly equal, excluding “None” responses and those not corresponding to the taxonomy levels, and a χ^2 -test (Pearson, 1900) yielded a p-value of 0.95, suggesting no statistically significant positional bias in the model’s choices, meaning that position does not significantly influence the model’s decision.

BWS For the BWS method, we were interested in evaluating how consistent the choices for the best and worst items in the 3-tuple were. After performing Cronbach’s α as a proxy for internal agreement within item triplets, which we used to estimate the internal consistency of the model’s preference orderings (Cronbach, 1951). Results

showed that the taxonomy levels like “Analyze” and “Understand” exhibit high consistency (0.84 and 0.69, respectively), indicating strong agreement in their classification. See Tables 6 and 7 for further qualitative in the Appendix.

BWS–PWC agreement The rank correlation between the BWS and PWC results reveals a lack of substantial agreement between the two approaches. The Spearman rank correlation coefficient (Spearman, 1904) is 0.169 with a p-value of 0.749, indicating a very weak and statistically insignificant positive correlation. Similarly, Kendall’s τ correlation coefficient (Kendall, 1938) is 0.086 with a p-value of 0.822, further suggesting minimal and non-significant agreement between the rankings produced by the two methods. Thus, these results suggest that the BWS and PWC methods do not produce comparable results, which is not entirely surprising, as the two methods differ in their approach to evaluating taxonomy levels.

BCA The BCA method yielded strong overall confidence in the model’s decisions, with an average linear probability of 0.956. In the subset of predictions where the model exhibited low confidence (with linear probabilities between 50 % and 60 %), the model produced “Yes” responses 56 times and “No” responses 44 times. Interestingly, most of these low-confidence predictions are associated with high-order taxonomy levels like “Create” and “Analyze,” suggesting that the model is less confident when handling more complex cognitive tasks. Moreover, the analysis of average confidence across taxonomy levels reveals that the model exhibits the highest confidence in its predictions for “Evaluate” (98.43 %) and “Create” (96.59 %). In contrast, while still high, the confidence for “Understand” (93.28 %) is slightly lower, reflecting the challenges in these areas. See Appendix Tables 8–10 in Section H.

RCA Finally, for the RCA method, we calculated the average linear probability for the short and long descriptions, which were 85.76 % and 83.03 % respectively, with minimum values of 37 % and 43 %. This indicates almost no difference in the model’s decisions between the short and long descriptions, with the average probability for the shorter description being slightly higher. Our results suggest no substantial verbosity bias, which may be attributed to the minimal difference in token length and the consistent use of associated verbs

with the taxonomies in both the shortest and longest paraphrases. See Figures 18–19 in the Appendix Section I.

7 Discussion

Non-LLM Methods: While prior research often assumes LLMs to outperform traditional NLP approaches, we include non-LLM baselines not merely for benchmarking but to reveal the types of errors these simpler systems make—especially regarding verb ambiguity and lack of contextual awareness. This diagnostic perspective is critical for understanding what specific challenges remain unsolved even by LLMs. Regex and fuzzy matching struggled with morphological and contextual variability (e.g., “design” fitting multiple taxonomy levels), while spaCy’s reliance on shallow parsing made it error-prone for compound objectives or those relying on deverbal nouns. Sentence-BERT, although semantically flexible, failed to resolve inferential tasks, such as distinguishing whether “critical thinking” corresponds to “Analyze” or “Evaluate.” These shortcomings underscore the limits of surface-level pattern recognition and basic lexical and semantic-level similarity in tasks requiring pedagogical reasoning.

LLM Methods: LLMs showed better ability to parse complex and implicit learning objectives, yet their strengths were uneven across taxonomy levels. Consistency analyses revealed high agreement for levels like “Understand” and “Analyze”—potentially due to clearer linguistic cues. However, lower agreement and confidence were found in “Remember” and “Create,” suggesting difficulty in anchoring either very low-level recall or high-level generative tasks. This could reflect both model limitations and ambiguities in how LOs are written by instructors.

The PWC and RCA methods further confirmed the model’s capacity to make consistent selections across different input formats, highlighting its reliability in both comparative and scalar evaluation tasks. In BCA, however, decreased confidence in “Create” and “Analyze” responses aligns with the annotation difficulties experts also expressed, pointing to shared challenges between human and machine reasoning in higher-order cognitive domains. For the BCA method, the model generally exhibited more difficulty with high-order taxonomy levels such as “Create” and “Analyze”. The MCS approach showed no significant differences in the

consistency of the rationale generated by the model were observed across different conditions. Compared to the gold standard, the observed discrepancies in the representation of “Create” highlight the need for more robust modeling and annotation practices for both ends of the taxonomy spectrum.

Bias and Robustness: Despite concerns raised in previous work regarding LLM susceptibility to framing-related biases, our results suggest that GPT-4 demonstrates a notable degree of robustness—though the presence of bias in other LLMs cannot be ruled out by this experiment alone. Specifically, we found no significant positional bias in pairwise comparisons (PWC). The nature of the taxonomy levels used in our study may have mitigated positional bias due to the clear distinctions between taxonomy descriptions. We also found no meaningful evidence of verbosity bias when comparing short and long taxonomy descriptions (RCA), which may be attributed to the minimal token length differences in the descriptions used. The model’s decisions remained stable across paraphrased prompts (MCS), further supporting its consistency. While not immune to uncertainty, particularly in assigning high-order categories like “Create”, the model’s behavior appears more influenced by the inherent complexity of certain taxonomy levels than by superficial prompt features. This contrasts with non-LLM methods, which exhibited more deterministic errors stemming from lexical surface features and lacked the inferential flexibility.

8 Conclusion

We analyzed various methods for mapping LOs to Bloom’s taxonomy levels, focusing on expert annotations compared to non-LLM and LLM techniques. non-LLM methods struggled with verb matchings and context-specific mappings. LLM methods generally demonstrated better performance and more uniform results. However, further improvement is necessary to address the challenges of LLM methods in automating the LO mapping process. Overall, we found that the LLM results from GPT-4 show minimal evidence of prompt-induced bias. These findings suggest that LLMs hold considerable promise in streamlining curriculum alignment tasks in educational settings, although careful design and validation remain essential to ensure pedagogical reliability.

Limitations

Firstly, utilizing LLMs, particularly closed-source models such as OpenAI's GPT-4, can be costly and lack transparency. Methods like BWS and PWC require multiple generations per item, which can become expensive at scale. Additionally, LLMs are susceptible to biases, including position bias, verbosity bias, and rationale-conditioning bias—where generating a rationale after a label may reinforce prior decisions. While our results did not show strong effects from these biases, we cannot entirely rule out their influence, especially since our study only examined GPT-4.

This reliance on a single LLM is a key limitation. GPT-4 was selected due to its strong performance and availability, but our findings may not generalize across other models. Future studies should replicate this analysis using different LLMs to assess robustness and uncover potential model-specific biases.

We also observed substantial but imperfect inter-annotator agreement among experts, reflecting the inherent ambiguity and interpretive nature of mapping LOs to Bloom's taxonomy. This suggests that ambiguities may originate from how LOs are written, and that more consistent instructional design practices could help. Mapping should ideally be integrated early in the curriculum development process, with educators selecting or revising LOs in alignment with desired cognitive levels.

Future work should also incorporate larger and more diverse datasets to enable broader generalization and better assessment of model behavior, and extend the study to additional languages such as German, whose linguistic structures may present unique challenges for LO classification.

Finally, while our evaluation focused on quantitative measures, integrating qualitative assessments—such as expert think-aloud protocols or post-task interviews (Creswell, 2009)—could offer deeper insights into both human and model reasoning. We encourage future research to explore hybrid workflows where LLMs and human experts collaborate to improve both mapping accuracy and pedagogical relevance.

Ethics Statement

To conduct human evaluations, we recruited five experts in higher education pedagogy, who were employed by one of our institutions and did not receive additional payment for the task. They took

part in the annotations voluntarily and could withdraw at any time. We did not collect personal or private information from the participants and ensured the confidentiality and anonymity of the annotators' responses.

References

- Shatha Al Ghazali, Nazar Zaki, Luqman Ali, and Saad Harous. 2024. Exploring the potential of ChatGPT as a substitute teacher: A case study. *International Journal of Information and Education Technology*, 14(2):271–278.
- 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, New York.
- Sousan Arafeh. 2016. Curriculum mapping in higher education: A case study and proposed content scope and sequence mapping tool. *Journal of Further and Higher Education*, 40(5):585–611.
- John Biggs. 1996. Enhancing teaching through constructive alignment. *Higher education*, 32(3):347–364.
- John Biggs, Catherine Tang, and Gregor Kennedy. 2022. *Teaching for Quality Learning at University 5e*. McGraw-hill education (UK).
- Benjamin S. Bloom, Max D. Engelhart, Edward J. Furst, Walker H. Hill, and David R. Krathwohl. 1956. *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain*. David McKay Company, New York.
- Jonathan Bragg, Mausam, and Daniel S Weld. 2018. Sprout: Crowd-powered task design for crowdsourcing. In *Proceedings of the 31st annual acm symposium on user interface software and technology*, pages 165–176.
- Wen-Chih Chang and Ming-Shun Chung. 2009. Automatic applying bloom's taxonomy to classify and analysis the cognition level of english question items. In *2009 Joint Conferences on Pervasive Computing (JCPC)*, pages 727–734. IEEE.
- Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? a study on judgement biases. *arXiv preprint arXiv:2402.10669*.
- Leshem Choshen, Ariel Gera, Yotam Perlitz, Michal Shmueli-Scheuer, and Gabriel Stanovsky. 2024. [Navigating the modern evaluation landscape: Considerations in benchmarks and frameworks for large language models \(LLMs\)](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024): Tutorial Summaries*, pages 19–25, Torino, Italia. ELRA and ICCL.

- Jacob Cohen. 1960. [A coefficient of agreement for nominal scales](#). *Educational and Psychological Measurement*, 20(1):37–46.
- Steven H. Cohen. 2003. Maximum difference scaling: Improved measures of importance and preference for segmentation.
- JW Creswell. 2009. Research design-qualitative, quantitative, and mixed methods approaches. *SAGE, California*.
- Lee J. Cronbach. 1951. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3):297–334.
- Eleonora Dubicki. 2019. Mapping curriculum learning outcomes to acrl’s framework threshold concepts: A syllabus study. *The Journal of Academic Librarianship*, 45(3):288–298.
- Vanessa Echeverría, Juan Carlos Gomez, and Marie-Francine Moens. 2013. Automatic labeling of forums using bloom’s taxonomy. In *Advanced Data Mining and Applications: 9th International Conference, ADMA 2013, Hangzhou, China, December 14-16, 2013, Proceedings, Part I 9*, pages 517–528. Springer.
- L Dee Fink. 2003. A self-directed guide to designing courses for significant learning. *University of Oklahoma*, 27(11):1–33.
- Anne Herrmann-Werner, Teresa Festl-Wietek, Friederike Holderried, Lea Herschbach, Jan Griewatz, Ken Masters, Stephan Zipfel, and Moritz Mahling. 2024. Assessing ChatGPT’s mastery of bloom’s taxonomy using psychosomatic medicine exam questions: Mixed-methods study. *Journal of Medical Internet Research*, 26:e52113.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength Natural Language Processing in Python](#).
- Trevor Hussey and Patrick Smith. 2008. Learning outcomes: A conceptual analysis. *Teaching in higher education*, 13(1):107–115.
- Jessica Huynh, Jeffrey P. Bigham, and Maxine Eskénazi. 2021. [A survey of NLP-related crowdsourcing HITs: What works and what does not](#). *ArXiv*, abs/2111.05241.
- Preeti Jaiswal. 2019. Using constructive alignment to foster teaching learning processes. *English Language Teaching*, 12(6):10–23.
- Maurice G. Kendall. 1938. [A new measure of rank correlation](#). *Biometrika*, 30(1-2):81–93.
- Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2023. Benchmarking cognitive biases in large language models as evaluators. *arXiv preprint arXiv:2309.17012*.
- Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. Sage Publications.
- Kultusministerkonferenz. 2017. [Qualifikationsrahmen für deutsche Hochschulabschlüsse](#). Im Zusammenwirken von Hochschulrektorenkonferenz und Kultusministerkonferenz und in Abstimmung mit Bundesministerium für Bildung und Forschung erarbeitet und von der Kultusministerkonferenz am 16.02.2017 beschlossen.
- Xintong Li, Jinya Jiang, Ria Dharmani, Jayanth Srinivasa, Gaowen Liu, and Jingbo Shang. 2024. [Open-world multi-label text classification with extremely weak supervision](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15084–15096, Miami, Florida, USA. Association for Computational Linguistics.
- Yuheng Li, Mladen Rakovic, Boon Xin Poh, Dragan Gašević, and Guanliang Chen. 2022. Automatic classification of learning objectives based on bloom’s taxonomy. In *Educational Data Mining 2022*, pages 530–537. International Educational Data Mining Society.
- Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.
- Robert Frank Mager and Nan Peatt. 1962. *Preparing Instructional Objectives*, volume 62. Fearon Publishers Palo Alto, California.
- 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?
- Manal Mohammed and Nazlia Omar. 2020. Question classification based on bloom’s taxonomy cognitive domain using modified tf-idf and word2vec. *PLoS one*, 15(3):e0230442.
- Nobal Niraula, Samet Ayhan, Balaguruna Chidambaram, and Daniel Whyatt. 2024. [Multi-label classification with generative large language models](#). In *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, pages 1–7.
- OpenAI. 2023. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Karl Pearson. 1900. [On the criterion that a given system of deviates from the probable is such that it can be supposed to have arisen from random sampling](#). *Philosophical Magazine*, 50(302):157–175.
- Solveig Randhahn and Frank Niedermeier. 2017. Quality assurance of teaching and learning in higher education institutions-training on internal quality assurance series module 3. *Training on Internal Quality Assurance Series (TrainQA)*, 3.
- Veerababu Reddy, Usha Rani Uppukonda, and N. Veer-anjaneyulu. 2024. [Enhancing multi-label text classification using adaptive promptify concepts](#). In *2024*

- 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–5.
- Thomas Charles Reeves and John G Hedberg. 2003. *Interactive Learning Systems Evaluation*. Educational Technology.
- 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.
- Michael Rodriguez and Anthony Albano. 2017. *The College Instructor’s Guide to Writing Test Items: Measuring Student Learning*. Routledge.
- Chenhui Shen, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4215–4233.
- Charles Spearman. 1904. [The proof and measurement of association between two things](#). *The American Journal of Psychology*, 15(1):72–101.
- Jasmina Sterz, Sebastian H Hofer, Maren Janko, Bernd Bender, Farzin Adili, Teresa Schreckenbach, Lukas Benedikt Seifert, and Miriam Ruesseler. 2019. Do they teach what they need to? an analysis of the impact of curriculum mapping on the learning objectives taught in a lecture series in surgery. *Medical teacher*, 41(4):417–421.
- Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *arXiv preprint arXiv:2405.01724*.
- Zhen Tan, Alimohammad Beigi, Song Wang, Ruocheng Guo, Amrita Bhattacharjee, Bohan Jiang, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024a. Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*.
- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024b. [Large language models for data annotation: A survey](#). *Preprint*, arXiv:2402.13446.
- Abdul Waheed, Muskan Goyal, Nimisha Mittal, Deepak Gupta, Ashish Khanna, and Moolchand Sharma. 2021. Bloomnet: A robust transformer based model for bloom’s learning outcome classification. *arXiv preprint arXiv:2108.07249*.
- Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. 2013a. Perspectives on crowdsourcing annotations for natural language processing. *Language resources and evaluation*, 47:9–31.
- 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](#). *Preprint*, arXiv:2403.18105.
- Xiaoyan Wang, Yelin Su, Stephen Cheung, Eva Wong, and Theresa Kwong. 2013b. An exploration of Biggs’ constructive alignment in course design and its impact on students’ learning approaches. *Assessment & Evaluation in Higher Education*, 38(4):477–491.
- Minghao Wu and Alham Fikri Aji. 2023. Style over substance: Evaluation biases for large language models. *arXiv preprint arXiv:2307.03025*.
- Hanyi Xu, Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Philip S. Yu. 2024. [Large language models for education: A survey](#). *Preprint*, arXiv:2405.13001.

A Verbs Associated with Bloom's Taxonomy Levels

Remember	Understand	Apply	Analyze	Evaluate	Create
				create	
		use		design	
	explain	compute	analyze	hypothesize	Judge
	summarize	solve	categorize	invent	Recommend
	paraphrase	demonstrate	compare	develop	Critique
	describe	apply	contrast	arrange	Justify
	illustrate	construct	separate	assemble	Appraise
arrange	classify	apply	apply	categorize	Argue
define	convert	change	change	collect	Assess
describe	defend	choose	discover	combine	Attach
duplicate	describe	compute	choose	comply	Choose
identify	discuss	demonstrate	compute	compose	Compare
label	distinguish	discover	demonstrate	construct	Conclude
list	estimate	dramatize	dramatize	create	Contrast
match	explain	employ	employ	design	Defend
memorize	express	illustrate	illustrate	develop	Describe
name	extend	interpret	interpret	devise	Discriminate
order	generalized	manipulate	manipulate	explain	Estimate
outline	give example(s)	modify	modify	formulate	Evaluate
recognize	identify	operate	operate	generate	Explain
relate	indicate	practice	practice	plan	Judge
recall	infer	predict	predict	prepare	Justify
repeat	locate	prepare	prepare	rearrange	Interpret
reproduce	paraphrase	produce	produce	reconstruct	Relate
select	predict	relate	relate	relate	Predict
state	Recognize	schedule	schedule	reorganize	Rate
	rewrite	show	show	revise	Select
	review	sketch	sketch	rewrite	Summarize
	select	solve	solve	set up	Support
	summarize	use	use	summarize	Value
	translate	write	write	synthesize	
				tell	
				write	

Table 3: Sample possible verbs associated with Bloom's taxonomy levels from [Anderson and Krathwohl \(2001\)](#). The six categories—**Remember**, **Understand**, **Apply**, **Analyze**, **Evaluate**, and **Create**—are ordered from lower- to higher-order cognitive processes, with the first three considered lower-order and the last three higher-order thinking skills.

B Expert Annotation Task and Results

Aligning Learning Objectives with Bloom's Taxonomy Levels

Thank you for taking the time to participate in this questionnaire. Your insights will help us for a study on LLM-based annotation of learning objectives.

Task Overview:

1. **Goal:** You will be provided with a course title and a learning objective from a module handbook, along with the six levels of Bloom's Taxonomy. There are 35 learning objectives that you are asked to align with the Bloom's taxonomy levels.
2. **Task:** Identify which taxonomy levels align with the given learning objectives. More than one level may be relevant. Choose all the relevant taxonomy levels that correspond to the presented learning objectives.

Below, you'll find descriptions of Bloom's taxonomy levels with some examples.

Bloom's Taxonomy Levels and Descriptions:

1. **"Remember":** "Remembering involves locating knowledge in long-term memory that is consistent with presented material and retrieving relevant knowledge from long-term memory."

Examples:

- "List the steps of the water cycle from memory."
- "Identify and define key terms related to cellular respiration."

2. **"Understand":** "Understanding involves constructing meaning from instructional messages, including oral, written, and graphic communication. This includes changing from one form of representation to another, finding a specific example or illustration of a concept or principle, determining that something belongs to a category, abstracting a general theme or major points, drawing a logical conclusion from presented information, detecting correspondence between two ideas, objects, and the like, and constructing a cause-and-effect model of a system."

Examples:

- "Summarize the main arguments of the Enlightenment philosophers in your own words."
- "Interpret the results of a scientific experiment and explain the significance of the findings."

3. **"Apply":** "Applying involves carrying out or using a procedure in a given situation. This includes applying a procedure to a familiar or an unfamiliar task."

Examples:

- "Apply the principles of supply and demand to analyze a real-world market scenario."
- "Use statistical methods to analyze a data set and interpret the results in a research report."

4. **"Analyze":** "Analyzing involves breaking material into its constituent parts and determining how the parts relate to one another and to an overall structure or purpose. This includes distinguishing relevant from irrelevant parts or important from unimportant parts of presented material, determining how elements fit or function within a structure, and determining a point of view, bias, values, or intent underlying presented material."

Examples:

- "Break down the components of a literary work to explore the relationship between its themes and character development."
- "Analyze the causes and effects of economic inflation in a specific historical period."

5. **"Evaluate":** "Evaluating involves making judgments based on criteria and standards. This involves detecting inconsistencies or fallacies within a process or product, determining whether a process or product has internal consistency, and detecting the appropriateness or the effectiveness of a procedure for a given problem."

Examples:

- "Evaluate the strengths and weaknesses of different approaches to climate change mitigation."
- "Judge the credibility of sources used in a research paper on public health policy."

6. **"Create":** "Creating involves putting elements together to form a coherent or functional whole, reorganizing elements into a new pattern or structure, coming up with alternative hypotheses based on criteria, devising a procedure for accomplishing some task, and inventing a product."

Examples:

- "Design an innovative solution to reduce carbon emissions in urban areas."
- "Construct a theoretical model to predict the impact of new technology on society."

Example Task:

Learning Objective: "Students should analyze the causes and effects of climate change and evaluate the effectiveness of current environmental policies in addressing these issues."

Question: Which taxonomy levels are relevant to the given learning objective?

Levels:

- Remember
- Understand
- Apply
- Analyze
- Evaluate
- Create

Possible Choices:

- Analyze
- Evaluate

Consent and Privacy:

- Please note that we do **not** collect any personal information during this questionnaire. **No** email addresses or identifying data will be saved.
- Your responses are **anonymous** and will be used solely for the purpose of this research.
- You can add any comments or suggestions in the **comment section** at the end of the questionnaire.

Thank you for your participation. Your contribution is greatly appreciated.

Figure 3: Annotation instruction presented to the experts. Example learning objectives are adapted from various instructional design resources and author-generated.

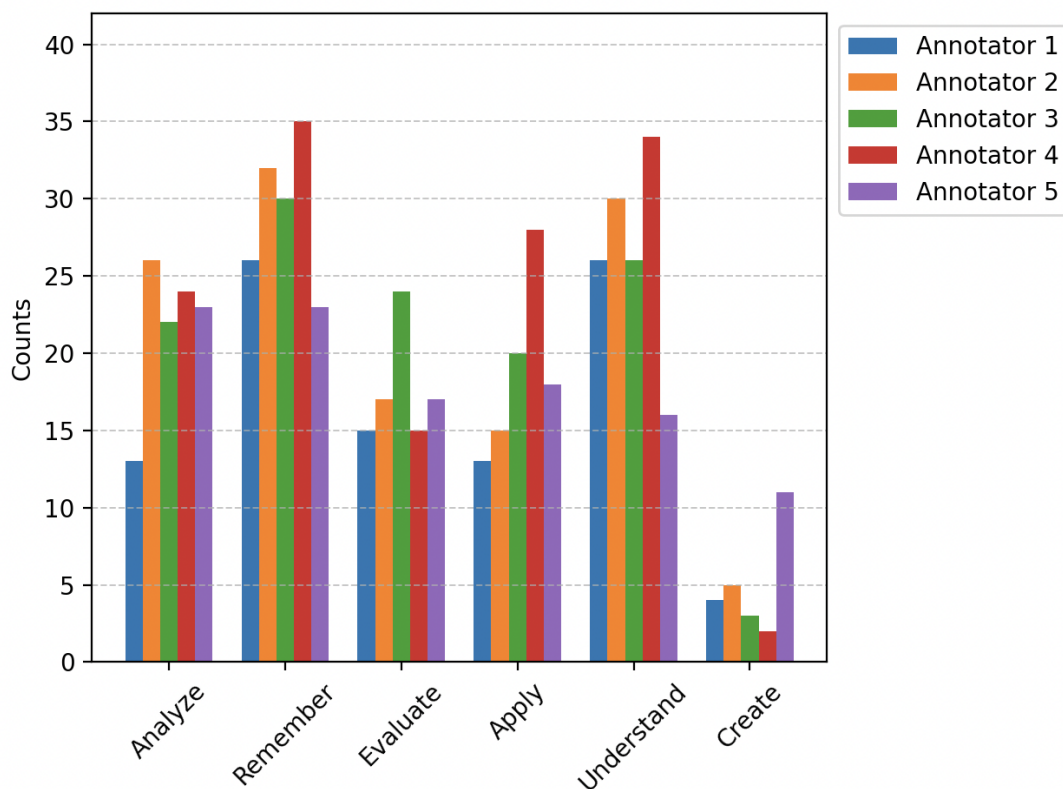


Figure 4: This figure shows the frequency of selection of the six categories in Bloom's taxonomy by the five annotators. Categories like "Remember" and "Understand" show more consistency across annotators, indicating higher consensus in assigning these levels. However, categories like "Create" and "Apply" show notable differences, suggesting interpretive variability in assigning LOs to these levels. The differences may reflect subjective biases or varying interpretations of the taxonomy levels, especially for categories that require high-order thinking skills (e.g., "Create"). This variability could indicate areas where further discussion is needed among annotators to reach a more uniform understanding.

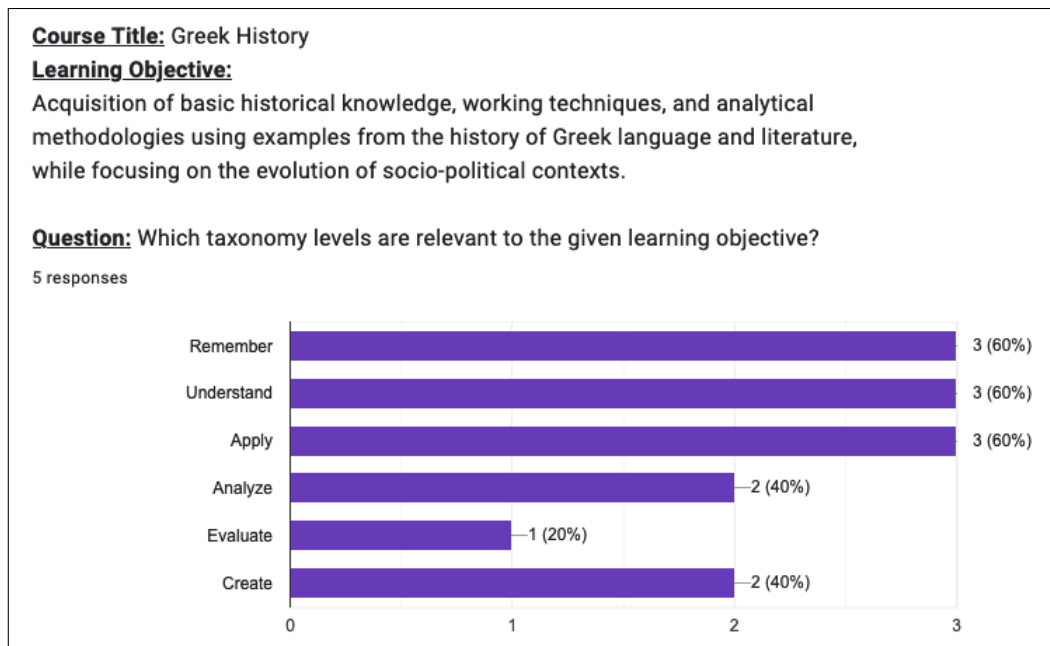


Figure 5: An example of high disagreement among expert annotators, driven by the complexity of the LO description provided by the educator for the course.

The learning objective blends several cognitive processes across Bloom’s taxonomy levels, making it challenging to determine the primary focus. “Acquisition of basic historical knowledge” aligns with **Remembering**, as it involves recalling historical facts and foundational knowledge. “Working technique” suggests **Applying**, since students are expected to practice and use specific methods in new contexts. “Analytical methodologies” leans toward **Analyzing**, as it requires breaking down examples of Greek language and literature into components, such as themes and structures, to better understand their function and meaning. Also, “focusing on the evolution of socio-political contexts” could be interpreted as **Understanding** (interpreting historical changes) or **Evaluating**, as it necessitates assessing the relationship between literature and its socio-political background.

Moreover, The connection between “historical knowledge” and “analytical methodologies” suggests a progression from lower-order skills (e.g., **Remembering** and **Understanding**) to higher-order skills (e.g., **Analyzing** and **Evaluating**). However, the LO does not specify which skill is prioritized, leading to annotators interpreting it differently based on their perspective. Finally, the inclusion of historical knowledge, literary analysis, and socio-political evolution adds a level of interdisciplinary complexity, as these dimensions often require varied cognitive processes to address.

C Paraphrases of Bloom’s Revised Taxonomy

- **Source:** Anderson and Krathwohl (2001)

- **Remember:** Remembering involves locating knowledge in long-term memory that is consistent with presented material and retrieving relevant knowledge from long-term memory.
- **Understand:** Understanding involves constructing meaning from instructional messages, including oral, written, and graphic communication. This includes changing from one form of representation to another, finding a specific example or illustration of a concept or principle, determining that something belongs to a category, abstracting a general theme or major points, drawing a logical conclusion from presented information, detecting correspondence between two ideas, objects, and the like, and constructing a cause-and-effect model of a system.
- **Apply:** Applying involves carrying out or using a procedure in a given situation. This includes applying a procedure to a familiar or unfamiliar task.
- **Analyze:** Analyzing involves breaking material into its constituent parts and determining how the parts relate to one another and to an overall structure or purpose. This includes distinguishing relevant from irrelevant parts or important from unimportant parts of presented

material, determining how elements fit or function within a structure, and determining a point of view, bias, values, or intent underlying presented material.

- **Evaluate:** Evaluating involves making judgments based on criteria and standards. This involves detecting inconsistencies or fallacies within a process or product, determining whether a process or product has internal consistency, and detecting the appropriateness or effectiveness of a procedure for a given problem.
 - **Create:** Creating involves putting elements together to form a coherent or functional whole, reorganizing elements into a new pattern or structure, coming up with alternative hypotheses based on criteria, devising a procedure for accomplishing some task, and inventing a product.
- **Source:** <http://www.nwlink.com/~donclark/hrd/bloom.html>
 - **Remember:** Remembering means recalling or retrieving previously learned information.
 - **Understand:** Understanding means comprehending the meaning, translation, interpolation, and interpretation of instructions and problems. State a problem in one's own words.
 - **Apply:** Applying means using a concept in a new situation or unprompted use of an abstraction. Applies what was learned in the classroom into novel situations in the workplace.
 - **Analyze:** Analyzing means separating material or concepts into component parts so that its organizational structure may be understood. Distinguishes between facts and inferences.
 - **Evaluate:** Evaluating means making judgments about the value of ideas or materials.
 - **Create:** Creating means building a structure or pattern from diverse elements. Put parts together to form a whole, with emphasis on creating a new meaning or structure.
- **Source:** <https://www.coloradocollege.edu/other/assessment/how-to-assess-learning/learning-outcomes/blooms-revised-taxonomy.html>
 - **Remember:** Remembering is retrieving, recalling, or recognizing relevant knowledge from long-term memory.
 - **Understand:** Understanding is demonstrating comprehension through one or more forms of explanation.
 - **Apply:** Applying is using information or skill in a new situation.
 - **Analyze:** Analyzing is breaking material into its constituent parts and determining how the parts relate to one another and/or to an overall structure or purpose.
 - **Evaluate:** Evaluating is making judgments based on criteria and standards.
 - **Create:** Creating is putting elements together to form a new coherent or functional whole; reorganizing elements into a new pattern or structure.
- **Source:** https://quincycollege.edu/wp-content/uploads/Anderson-and-Krathwohl_Revised-Blooms-Taxonomy.pdf
 - **Remember:** Remembering is recognizing or recalling knowledge from memory. Remembering is when memory is used to produce or retrieve definitions, facts, or lists, or to recite previously learned information.
 - **Understand:** Understanding is constructing meaning from different types of functions be they written or graphic messages or activities like interpreting, exemplifying, classifying, summarizing, inferring, comparing, or explaining.
 - **Apply:** Applying is carrying out or using a procedure through executing or implementing. Applying relates to or refers to situations where learned material is used through products like models, presentations, interviews, or simulations.
 - **Analyze:** Analyzing is breaking materials or concepts into parts, determining how the parts relate to one another or how they interrelate, or how the parts relate to an overall structure or purpose. Mental actions included in this function are differentiating, organizing, and attributing,

as well as being able to distinguish between the components or parts. When one is analyzing, he/she can illustrate this mental function by creating spreadsheets, surveys, charts, or diagrams, or graphic representations.

- **Evaluate:** Evaluating is making judgments based on criteria and standards through checking and critiquing. Critiques, recommendations, and reports are some of the products that can be created to demonstrate the processes of evaluation. In the newer taxonomy, evaluating comes before creating as it is often a necessary part of the precursory behavior before one creates something.

- **Create:** Creating is putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure through generating, planning, or producing. Creating requires users to put parts together in a new way, or synthesize parts into something new and different creating a new form or product. This process is the most difficult mental function in the new taxonomy.

• **Source:** <https://www.allencountyesc.org/Downloads/BloomsVerbsAlphabetized.pdf>

- **Remember:** Remember previously learned information.

- **Understand:** Demonstrate an understanding of the facts.

- **Apply:** Apply knowledge to actual situations.

- **Analyze:** Break down objects or ideas into simpler parts and find evidence to support generalizations.

- **Evaluate:** Make and defend judgments based on internal evidence or external criteria.

- **Create:** Compile component ideas into a new whole or propose alternative solutions.

D Custom Prompts for Different LLM Methods

```
1 prompt = f"""
2     Given the following learning objective: "{LO segments
3     appear here}",
4     compare it against the Bloom's Taxonomy level
5     descriptions provided below.
6
7     {Bloom's taxonomy descriptions and paraphrases appear
8     here}
9
10    **Instructions:**
11    1. First, provide a very brief reasoning for the
12    identified level.
13    The reasoning should not exceed three sentences and
14    should only
15    be based on the content of the learning objective
16    provided.
17    2. Then, return the identified taxonomy levels as a list
18    of strings.
19    """
```

Figure 6: Prompt used in the MCS method for condition B for identifying the appropriate Bloom's Taxonomy levels. We employed the same prompt for all MCS conditions but altered the task sequence in Conditions B and C, and separated rationale generation from multiple-choice selection in Condition A.

```

1 prompt = f"""
2     **Task:**
3     For each learning objective:
4     - Compare the sentence against the taxonomy options.
5     - Select the most relevant taxonomy level to the
6       sentence in each pair.
7     - Only choose one taxonomy level from the pair.
8     - If no taxonomy level matches the sentence given,
9       return 'None' but do not provide an explanation.
10
11     **Example:**
12     - Learning Objective: "List the steps of the
13       scientific method."
14     - pairs: {'Remember': 'Recall facts and basic
15       concepts', 'Understand': 'Explain ideas or
16       concepts', 'Evaluate': 'Justify a decision or
17       course of action'}}
18     - Output: 'Remember'
19
20     **Input:**
21     Learning Objective: "{LO segments appear here}"
22     Taxonomy Options: "{Taxonomy level pairs will appear
23       here}"
24
25     Which one is the most relevant taxonomy level to the
26     learning objective?
27     Answer:
28     """

```

Figure 7: Prompt used in the PWC method for selecting the most relevant Bloom's Taxonomy level.


```

1 prompt = f"""
2     **Task:**
3     For each learning objective:
4     - Compare the sentence against the taxonomy options.
5     - Select the taxonomy level that is the most related to
6       the sentence.
7     - Select the taxonomy level that is the least related to
8       the sentence.
9     - Do not provide an explanation.
10
11    **Example:**
12    - Learning Objective: "List the steps of the scientific
13      method."
14    - Taxonomy Options: {'remember': 'Recall facts and
15      basic concepts', 'understand': 'Explain ideas or
16      concepts', 'evaluate': 'Justify a decision or course
17      of action'}
18    - Output: {'most': 'remember', 'least': 'evaluate'}
19
20    **Input:**
21
22    Sentence: "{LO segments appear here}"
23    Taxonomy Options: "{Taxonomy level tuples appear here}"
24
25    What are the most and least related taxonomy levels to
26      the given sentence?
27
28    Answer:
29    """

```

Figure 8: Prompt used for selecting the most and least related Bloom's Taxonomy levels in BWS method.

```

1 prompt = f"""
2     **Task:**
3     For each learning objective:
4     - Compare the sentence against the taxonomy description
5       provided.
6     - Rate how relevant is the taxonomy description to the
7       learning objective on a scale of 1 to 5, where 1 is
8       the least relevant and 5 is the most relevant.
9     - Only use whole numbers from 1 to 5. Do not use
10      fractions or decimal values.
11     - Do not provide an explanation.
12
13     **Example:**
14     - Learning Objective: "List the steps of the scientific
15       method."
16     - Taxonomy level: {'remember': 'Recall facts and basic
17       concepts'}
18     - Answer: 5
19
20     **Input:**
21
22     Learning Objective: "{LO segments appear here}"
23     Taxonomy level: "{Taxonomy levels appear here}"
24
25     Rate the relevance of the taxonomy level to the given
26     learning objective (1 to 5):
27     Answer:
28     """

```

Figure 9: Prompt used in the RCA method for rating the relevance of taxonomy descriptions. The same prompt is used for short and long descriptions of the taxonomy levels.

```

1 prompt = f"""
2     **Task:**
3     Compare the sentence to the provided taxonomy
4         description. Determine if the taxonomy level and its
5         description accurately describe the sentence provided
6         .
7
8     Answer with "Yes" if the taxonomy level and description
9         accurately describe the sentence.
10    Answer with "No" if the taxonomy level and description
11    do not accurately describe the sentence.
12    Do not provide explanations, just the "Yes" or "No"
13    answer.
14
15    **Example:**
16
17    Sentence: "The student can recall key terms and concepts
18        from the lesson."
19    Taxonomy Level and description: "Remember: it refers to
20        recalling information."
21    Is the description accurate for the sentence?
22    Answer: Yes
23
24    **Input:**
25
26    Learning Objective: "{LO segments appear here}"
27    Taxonomy level: "{Taxonomy levels appear here}"
28
29    Is the description accurate for the sentence?
30    Answer:
31    """

```

Figure 10: Prompt used in the BCA method.

E Results from non-LLM and LLM Methods Compared to Gold Standard Annotation

Course Title: Facing Death: Basic Module

Learning Objective:

After successfully completing this course, students will be able to:

- name the objectives of the degree program and the professional fields for which the degree program qualifies;
- explain the various basic questions and contents of the degree program and how they relate to each other;
- clarify and identify their own focus and goals for the degree program;
- come to terms with their own death;
- describe the basic meanings of palliative care, spiritual care and self-care;
- reflect on basic questions of the individual and social relationship to death

Question: Which taxonomy levels are relevant to the given learning objective?

5 responses

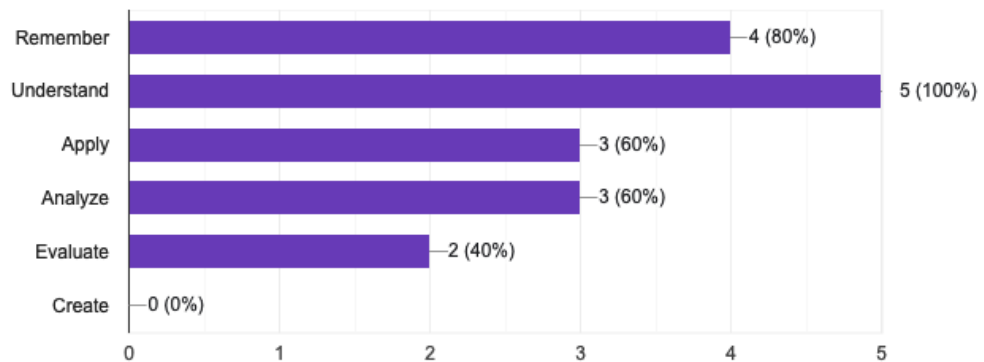


Figure 11: An example of expert annotations for a course LO description, mapped to Bloom's revised taxonomy levels by five expert annotators. The corresponding mappings by non-LLM and LLM-based methods are presented in Table 4.

Category	Method	Remember	Understand	Apply	Analyze	Evaluate	Create
Non-LLM	SpaCy		✓	✓	✓	✓	
	Regex		✓	✓	✓	✓	
	Fuzzy		✓	✓	✓	✓	✓
	Semantic Similarity		✓	✓	✓	✓	
LLM	BWS		✓	✓	✓	✓	
	PWC	✓	✓	✓	✓	✓	
	Short Rating	✓	✓	✓			
	Long Rating	✓	✓	✓			
	Binary Combinations	✓	✓	✓	✓	✓	✓
	MCS: Condition A	✓	✓	✓	✓	✓	
	MCS: Condition B	✓	✓		✓	✓	
	MCS: Condition C	✓	✓	✓		✓	

Table 4: Comparison of non-LLM and LLM-based methods in mapping the same course learning objective to Bloom’s revised taxonomy levels. Check marks indicate the taxonomy levels identified by each method.

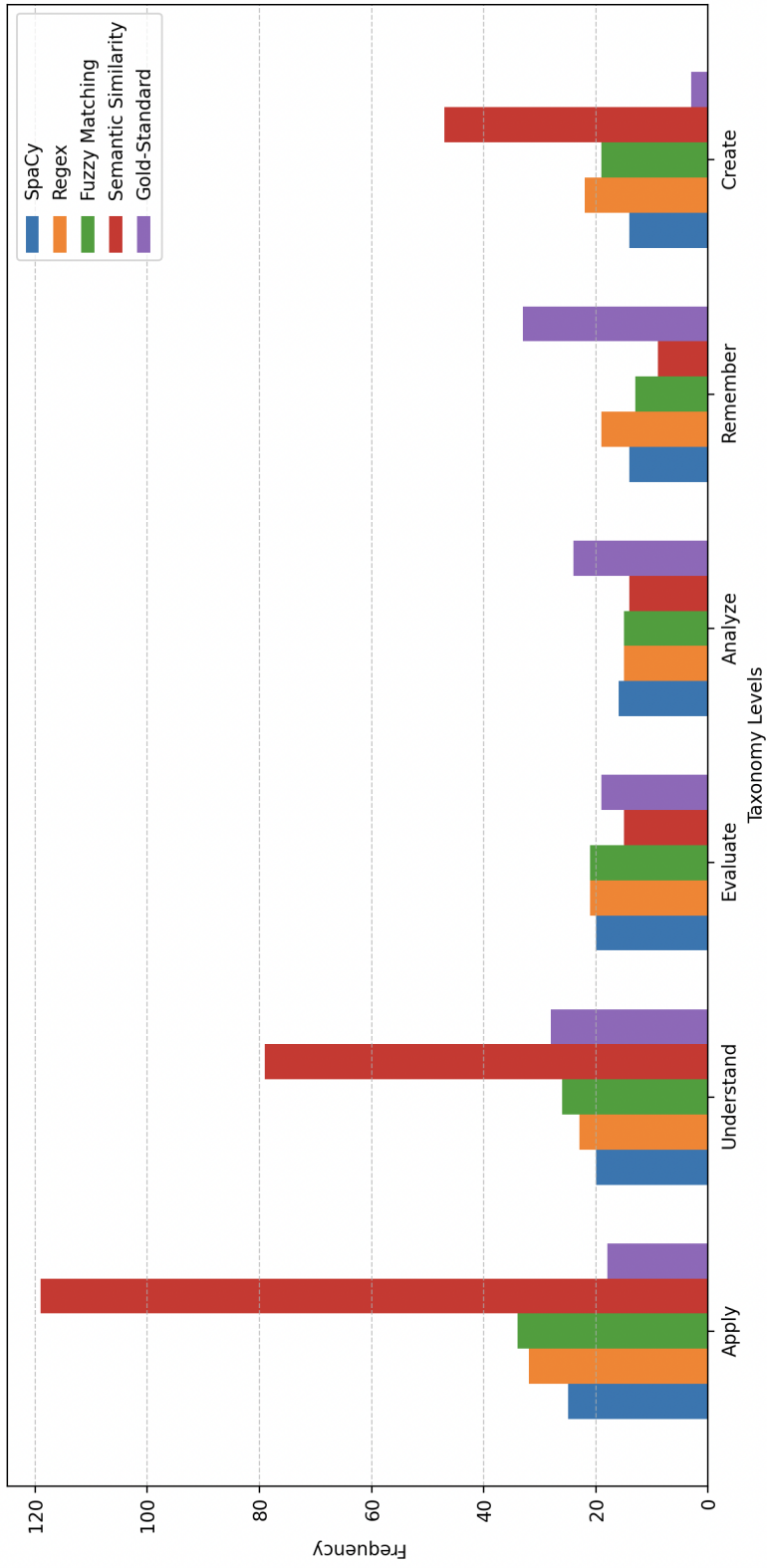


Figure 12: Frequency distribution of taxonomy levels for non-LLM methods and the gold standard annotations.

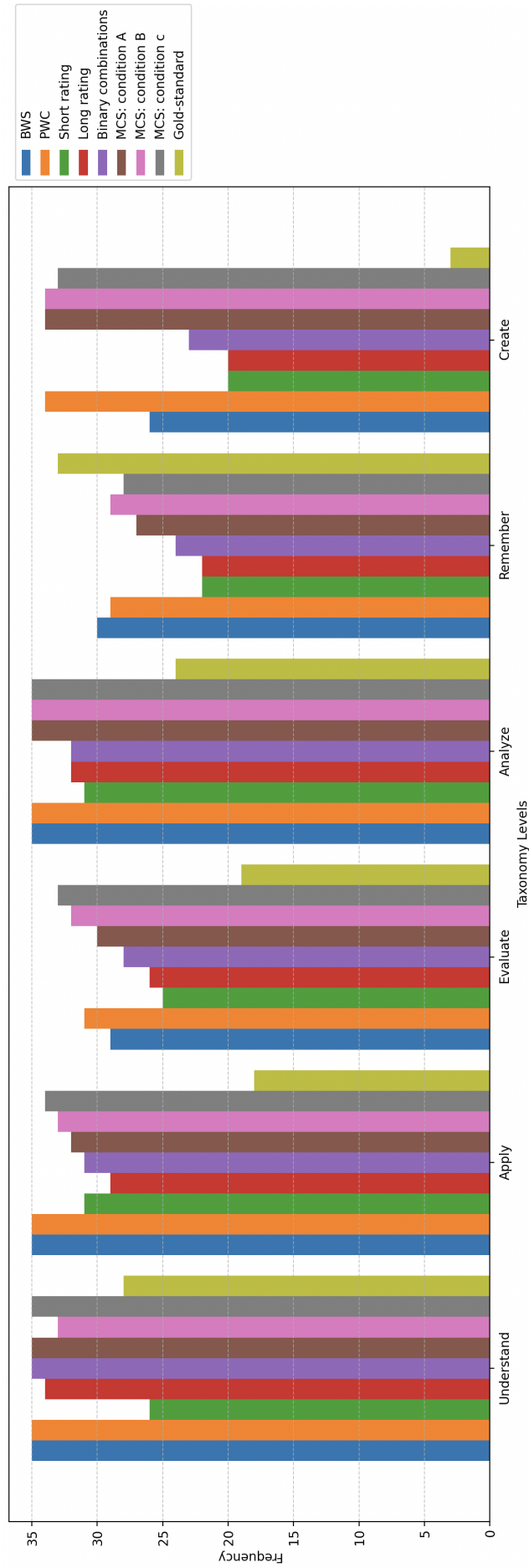


Figure 13: Frequency distribution of taxonomy levels for LLM methods and the gold standard annotations. The figure illustrates how different methods (e.g., BWS, PWC) align with the gold standard across various taxonomy levels.

F Results from Multiple Choice Selection with Paraphrase-Consistency Prompting and Rationale Generation

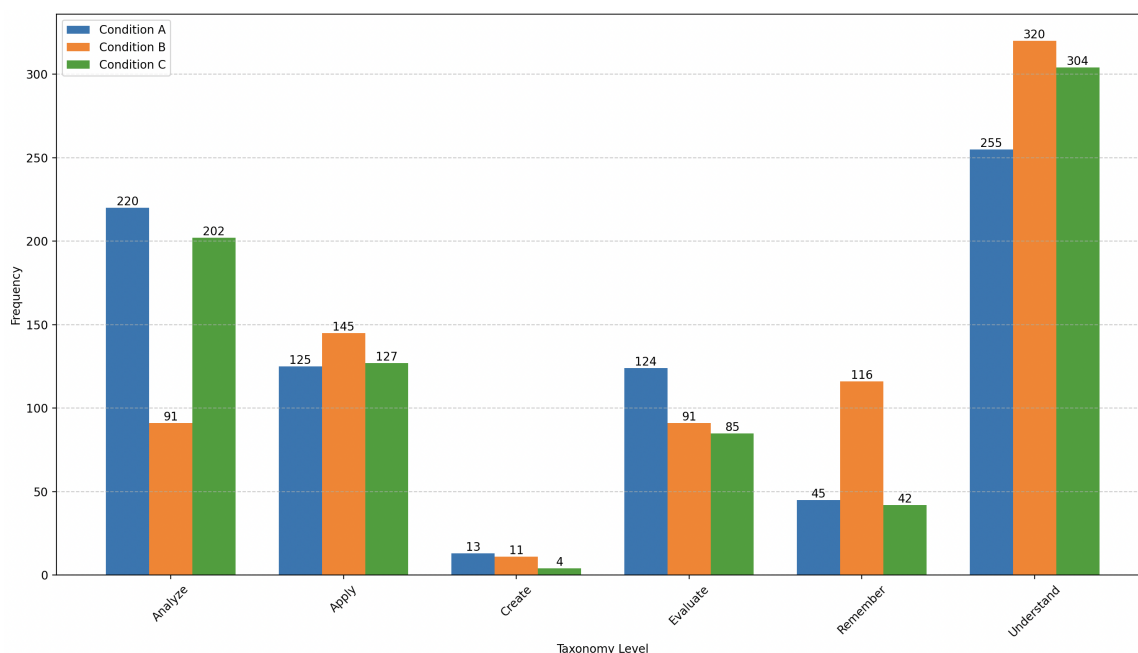


Figure 14: Frequency count of taxonomy levels per conditions. The frequency counts reveal that “Understand” is the most common taxonomy level across all conditions, while “Create” is the least frequent. There are notable variations in the frequencies of other taxonomy levels: for instance, “Apply” is more frequent in Condition B, and “Analyze” shows a higher frequency in Condition C.

Condition	Full Agreement Ratio	Partial Agreement Ratio
Condition A	0.23	0.92
Condition B	0.46	0.95
Condition C	0.76	0.98

Table 5: Agreement analysis for conditions A, B, and C. We present the model’s average alignment consistency score, highlighting cases of **full agreement** (where the model’s choice of taxonomy levels is identical across all paraphrases) and **partial agreement** (where the model’s choice is consistent in at least three of the five paraphrases) as detailed here. The results indicate that the selection-reasoning bias—where rationales tend to align with an initial label—is supported by the data. In Condition C, where the rationale is based on an initial selection, there is a higher alignment in taxonomy levels across paraphrases. Conversely, Conditions A and B show lower full agreement ratios, suggesting that without an initial selection to base the rationale on, the agreement among paraphrases is less consistent.

G Results from PWC vs. BWS Annotations

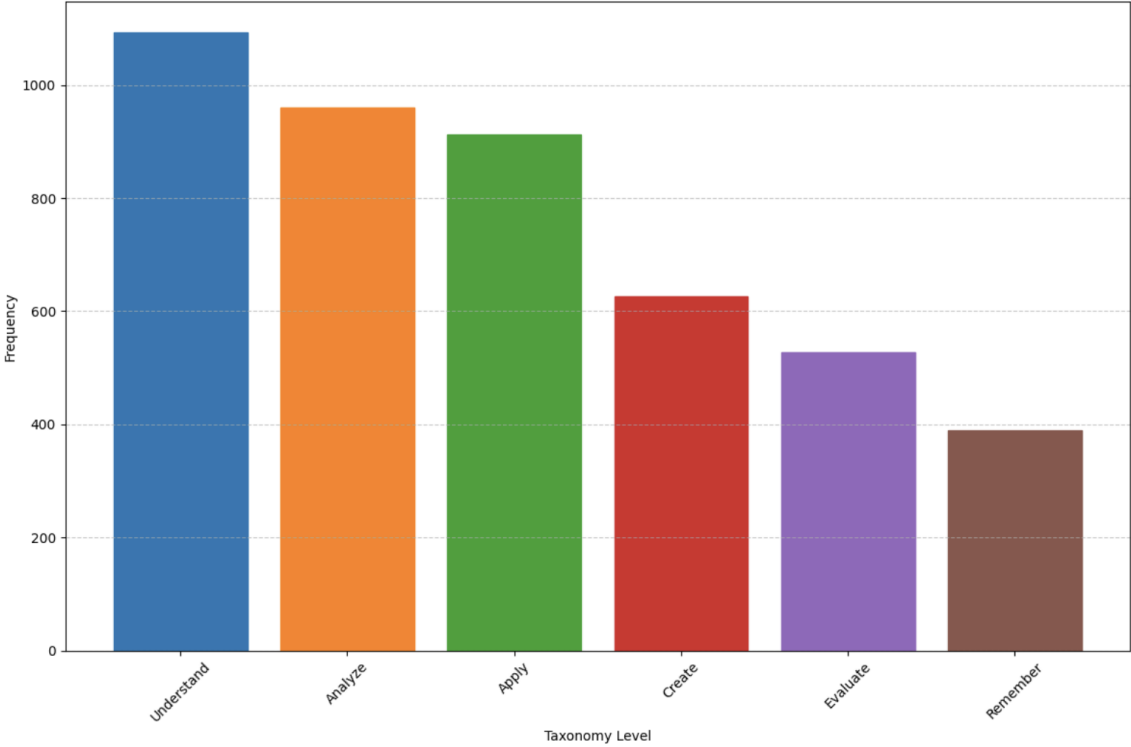


Figure 15: Frequency distribution of taxonomy levels in pair-wise analysis

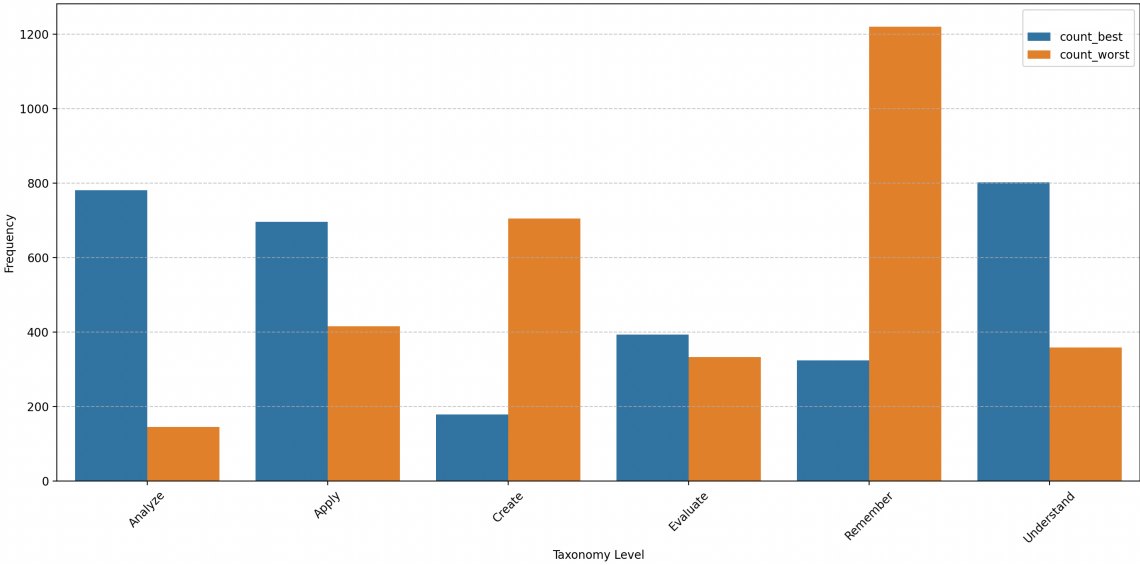


Figure 16: Best-worst frequency counts across taxonomy levels.

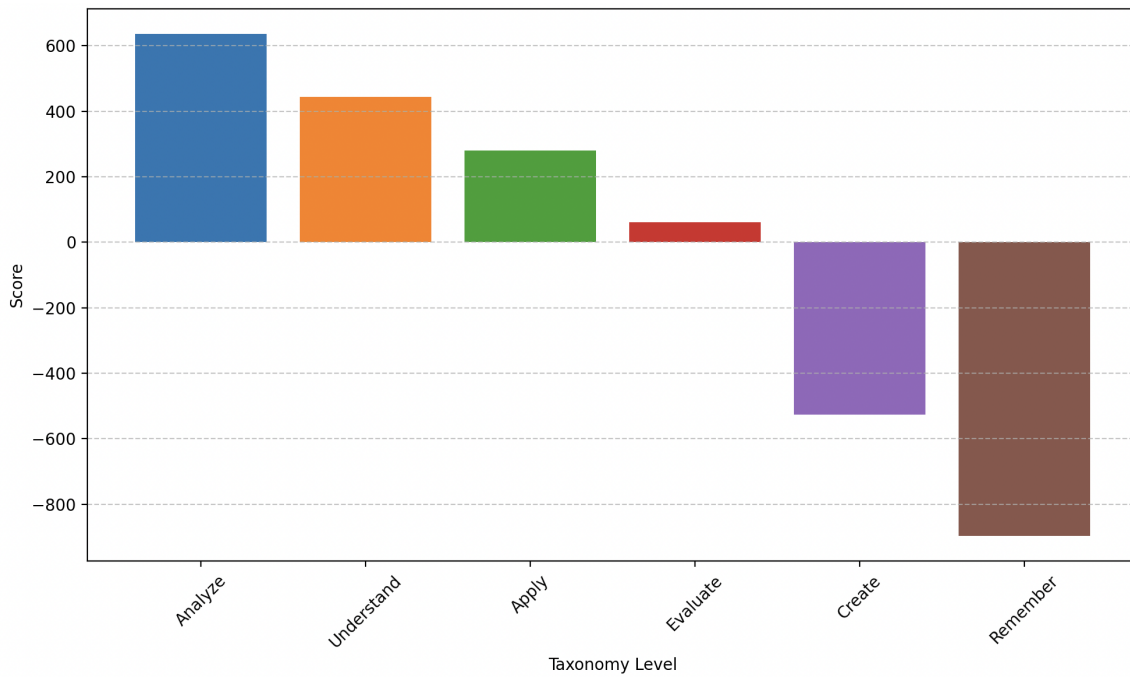


Figure 17: Best-Worst scaling scores: “Analyze” and “Understand” are the most preferred or relevant taxonomy levels, as reflected by their high positive scores.

Taxonomy Level	Consistency Score
Remember	0.21
Evaluate	0.54
Understand	0.69
Apply	0.62
Create	0.20
Analyze	0.84

Table 6: Cronbach α 's measure of internal consistency scores for taxonomy levels

Taxonomy Level	Mean Rank
Remember	6.0000
Understand	2.0025
Apply	2.9975
Analyze	1.0000
Evaluate	4.0000
Create	5.0000

Table 7: **Sensitivity Analysis (Mean Ranks)**: This analysis assesses the stability of rankings across various samples or iterations. The mean ranks reflect the relative significance assigned to each taxonomy level by the model, where a higher score indicates lower significance. “Analyze” and “Understand” are ranked as the most important, while other levels show varying degrees of relevance.

H Results from Binary Annotations

Threshold	Yes_Count	Total_Count	Yes_Percentage
80	274	872	31.42 %
85	267	853	31.30 %
90	248	821	30.21 %
95	227	775	29.29 %

Table 8: **Threshold Variation Analysis:** The analysis demonstrates how varying confidence thresholds impact the proportion of “Yes” responses. As the threshold increases from 80 to 95, the percentage of “Yes” responses slightly declines from 31.42 % to 29.29 %. This suggests that higher thresholds may reduce the model’s overall affirmative responses, potentially filtering out less confident predictions.

Taxonomy	Average Correct Binary Rate
Analyze	0.459119
Apply	0.371069
Create	0.176101
Evaluate	0.232704
Remember	0.157233
Understand	0.572327

Table 9: **Comparison with Multi-Class Classification for Bloom Taxonomy:** The model’s performance varies significantly across different Bloom’s taxonomy levels. For example, “Understand” has the highest average correct binary rate at 57.23 %, while “Remember” and “Create” are much lower, at 15.72 % and 17.61 %, respectively. This indicates that the model is better at aligning with high-order thinking skills such as “Understand” and “Analyze” but struggles more with “Create” and “Remember.”

Taxonomy Level	Average Confidence
Remember	96.07
Understand	93.28
Apply	94.20
Analyze	95.47
Evaluate	98.43
Create	96.59

Table 10: Comparison across D different taxonomy levels: The analysis of average confidence across taxonomy levels reveals that the model exhibits the highest confidence in its predictions for “Evaluate” (98.43 %) and “Create” (96.59 %). In contrast, while still high, the confidence for “Understand” (93.28 %) is slightly lower.

I Results from Rating Annotations

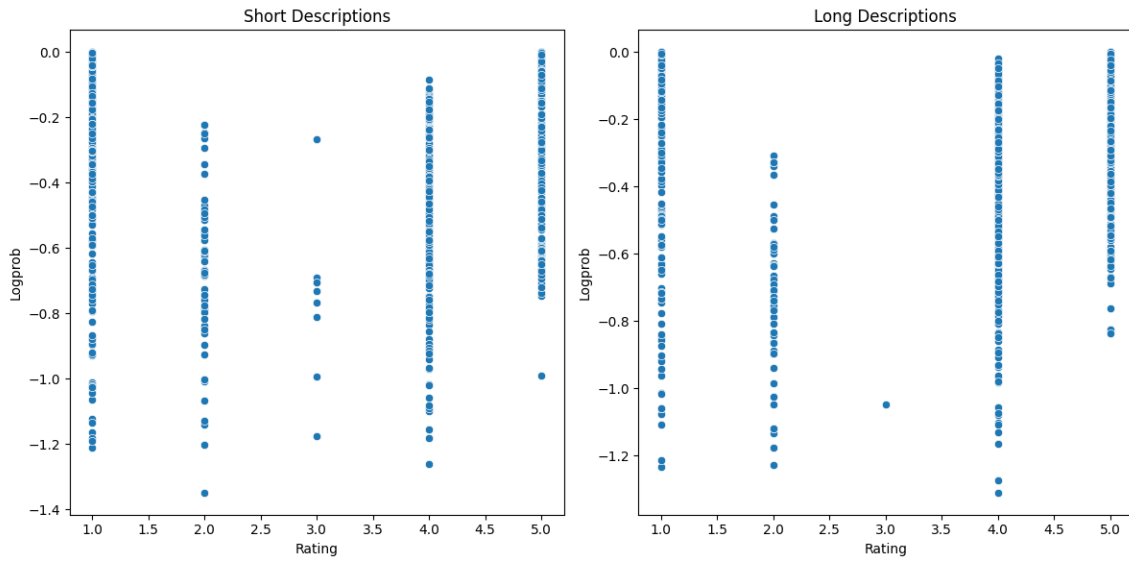


Figure 18: The logprob values for the **short descriptions** are relatively consistent across ratings, without any clear trend, suggesting that the model’s confidence in its rating wasn’t strongly influenced by its rating when using short descriptions.

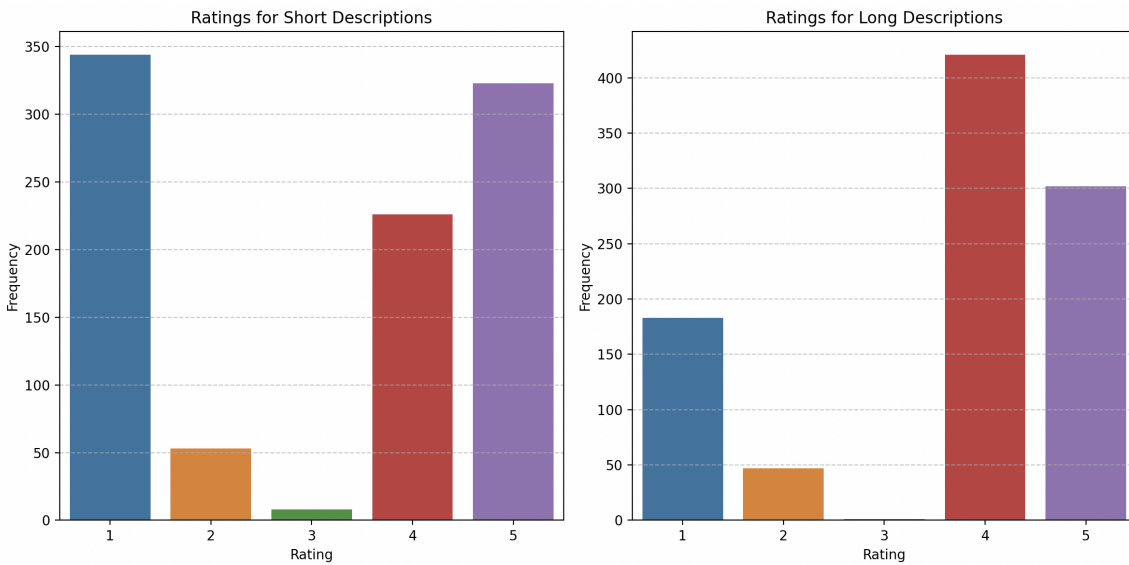


Figure 19: For the **short descriptions**, the most frequent ratings are at the extremes: 1 and 5. The high frequency of 1s indicates that many learning objectives were poorly aligned with the taxonomy level when only a short description was provided. On the other hand, there is also a significant cluster at 5, suggesting that some objectives were still rated highly despite the brevity of the descriptions. While for **long descriptions** there is still a notable peak at 1, indicating poor alignment for some objectives, the second peak is at 4, and there is a considerable amount of ratings at 5. The peak at 4, with a significant tail towards 5, indicates that the detailed descriptions helped many objectives align better with the taxonomy level.

LangEye: Toward ‘Anytime’ Learner-Driven Vocabulary Learning From Real-World Objects

Mariana Shimabukuro, Deval Panchal, and Christopher Collins

Ontario Tech University, Oshawa, ON, Canada

{mariana.shimabukuro, christopher.collins}@ontariotechu.ca

Abstract

We present LangEye, a mobile application for contextual vocabulary learning that combines learner-curated content with generative NLP. Learners use their smartphone camera to capture real-world objects and create personalized “memories” enriched with definitions, example sentences, and pronunciations generated via object recognition, large language models, and machine translation. LangEye features a three-phase review system — progressing from picture recognition to sentence completion and free recall. In a one-week exploratory study with 20 French (L2) learners, the learner-curated group reported higher engagement and motivation than those using pre-curated materials. Participants valued the app’s personalization and contextual relevance. This study highlights the potential of integrating generative NLP with situated, learner-driven interaction. We identify design opportunities for adaptive review difficulty, improved content generation, and better support for language-specific features. LangEye points toward scalable, personalized vocabulary learning grounded in real-world contexts.

1 Introduction

Creating contextual learning opportunities remains a major challenge in second language (L2) acquisition, particularly for learners situated in non-native environments. Immersive experiences, such as studying abroad or participating in language-rich communities, are often inaccessible due to financial, geographic, or logistical barriers (Galloway and Ruegg, 2020). Mobile-Assisted Language Learning (MALL) addresses this by leveraging the ubiquity and portability of smartphones to support “anytime” micro-learning and situated learning approaches (Arakawa et al., 2022; Byrne, 2019; Tran et al., 2023). Yet, many current MALL systems provide limited flexibility in adapting dynamically to learners’ immediate context, personal

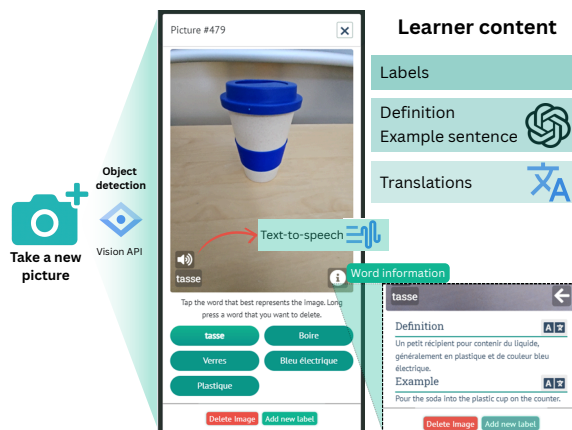


Figure 1: System diagram of LangEye, illustrating the flow from learner image capture through API-based vocabulary enrichment. The application integrates Google Cloud Vision, Cloud Translation, and Text-to-Speech APIs, along with OpenAI’s GPT model, to generate personalized vocabulary “memories” enriched with labels, definitions, translations, and pronunciation.

interests, and cognitive availability, and often rely exclusively on pre-curated, static content.

We introduce LangEye¹, a mobile application for vocabulary learning that turns real-world objects into interactive “memories” through a learner-curated workflow. Using smartphone cameras and NLP services — including computer vision, large language models, and machine translation — LangEye generates personalized lexical entries with definitions, example sentences, and pronunciation. Learners engage with this content through a structured review system that supports progressive recall and production in the target language. An overview of the system architecture and API integration is shown in Figure 1.

Designed specifically to empower self-directed learners, LangEye supports short, personalized learning interactions directly tied to learners’ physi-

¹Public demo and repository to be made available at <https://vialab.ca/langeye>.

cal environments and motivations. Crucially, all review sessions are initiated by learners and grounded in their uniquely captured contexts, promoting deeper personalization, engagement, and learner autonomy. However, due to this highly personalized and learner-curated design, traditional standardized assessments of vocabulary learning outcomes — such as standardized pre- and post-tests — are challenging, as vocabulary items vary greatly across individuals.

To explore the feasibility, learner acceptance, and design implications of LangEye, we conducted an initial one-week exploratory study with 20 French L2 learners, comparing a *camera group* (using LangEye to generate personalized vocabulary entries) and a *control group* (using pre-curated vocabulary). Preliminary findings highlight the motivational and engagement benefits of integrating learner-curated AI-generated content, while also revealing limitations associated with computer vision accuracy and AI-generated contextual sentences. These insights lay the groundwork for our planned longitudinal evaluation, which will rigorously measure personalized vocabulary acquisition and retention over extended use periods. Additionally, future iterations of LangEye will incorporate advanced object detection methods (e.g., YOLO-E) and more dynamic, interactive scenarios such as gamified object treasure hunts, further enhancing contextual vocabulary learning through data-driven methods.

2 Background and Related Work

2.1 Learner-curated Vocabulary with MALL Applications

Compared to Computer-Assisted Language Learning (CALL), MALL excels in accessibility and context-driven learning, making it effective for vocabulary acquisition (Alhuwaydi, 2022; Klimova, 2021). Micro-learning involves short, targeted activities (e.g., 5–15 minutes) (Leong et al., 2020). For instance, MiniHongo (Tran et al., 2023) integrates location and activity data to deliver contextual vocabulary lessons, demonstrating the efficacy of location-relevant micro-learning. Similarly, VocaBura (Hautasaari et al., 2019) utilizes audio and location-based prompts to teach vocabulary during real-world interactions.

VocabEncounter (Arakawa et al., 2022), a CALL application, applies contextual and micro-learning by integrating target vocabulary into web content

via natural language processing (NLP) and machine translation (MT) techniques. Comparable techniques embed vocabulary into audiovisual content through automatic glossing and lexical simplification (Alm, 2021; Fievez et al., 2023). VocabNomad (Tsourounis and Demmans Epp, 2016) provides a highly personalized MALL experience with progress tracking, contextual recommendations, and learner-curated entries. By allowing users to add vocabulary, record pronunciations, and browse visual collections, it fosters situated and personalized learning.

These studies highlight the importance of integrating relevant and contextual vocabulary learning into daily life, leveraging micro-learning and personalized approaches. However, most rely on static content or fixed corpora, with limited opportunities for learners to drive content creation based on their immediate environment.

2.2 AI-Enabled Context Personalization for Vocabulary Learning

The advent of large language models (LLMs), beginning with ChatGPT², has enabled more dynamic natural language generation, allowing for real-time synthesis of definitions, example sentences, and explanations. While generative AI presents challenges such as ethical concerns and content accuracy (Campolo and Crawford, 2020), it has opened new possibilities in personalized educational applications, particularly in language learning.

Applications like Storyfier (Peng et al., 2023) leverage generative AI to create vocabulary-rich narratives based on learner input. Although they showed limited learning gains, users appreciated the contextualization and narrative integration. Similarly, Leong et al. (2024) found that AI-generated personalized prompts enhanced learner motivation, despite modest measurable gains in vocabulary retention.

Recent systems also incorporate generative AI into mixed-reality environments. WordSense (Vazquez et al., 2017) pioneered contextual vocabulary learning through object recognition linked to dynamically generated content. More recently, FluencyAR (Hollingworth and Willett, 2023) integrated augmented reality (AR) with generative feedback for self-talk, and CuriosityXR (Vaze et al., 2024) allowed educators to create multi-modal, contextual mini-lessons. These

²<https://openai.com/research/overview>. Accessed April 2025

works emphasize engagement and curiosity, often powered by NLP-driven interfaces.

LangEye extends MALL, integrating micro-, situated-, and contextual-learning with modern NLP technologies, including object recognition, large language models for content generation, machine translation, and text-to-speech. Unlike prior systems that personalize content using static corpora or predefined curricula, LangEye allows learners to initiate the content pipeline through real-world object interactions, enabling highly contextualized and self-directed vocabulary acquisition. This learner-driven approach aims to promote both personalization and autonomy, but it also challenges traditional evaluation methods, as vocabulary exposure varies widely across individuals. As such, LangEye raises important questions around how to evaluate open-ended, NLP-enhanced learning systems, where learner agency and environmental context shape the learning trajectory.

3 LangEye Design: Create and Review Memories

LangEye’s core interaction is structured around learner-generated *memories* — vocabulary entries tied to real-world images captured by the learner. These memories are enriched using NLP services to provide multilingual definitions, contextual sentences, and audio pronunciation, supporting both vocabulary learning and retention. In this case, the vocabulary items are associated with the pictures taken by the learners. Figure 2 illustrates the learner taking a picture of a cup (*tasse* in French) and interacting with the generated memory’s word definition and example sentence. Therefore, the learned words are tied to a familiar object, which is more effective when compared to unfamiliar or no pictures for vocabulary learning (Hwang et al., 2014; Kang, 1995; Saidbakhramovna et al., 2021).

The app creates situational learning opportunities by allowing the learner to interact with objects around them in three ways: (1) take pictures of objects which they can interact with in-situ via editing or exploring the picture (*memory*); (2) take pictures of objects now, but choose to edit or explore the picture (*memory*) later; and (3) start a *Review Memories* session with a desired length — 3 to N words for up to 3 phases per session. Figure 2 (c, d, and e) shows examples of review activities for each phase.

3.1 Creating Memories

In *picture mode*, the learner can aim their device’s camera at an object they wish to interact with in the target language (TL) and take a picture of it. As shown in Figure 2, in response, the app returns a list of five likely labels (names) for the object, sorted by probability (most likely object name as the top result). The learner can explore the definition and a sample sentence for each label, and optionally select a different top label based on that information. Alternatively, they can confirm the default suggestion. On the same screen, learners also have the option to listen to the pronunciation of the object name. At this point, they may choose to take another picture to explore more objects or end their study session. The app saves all objects, pictures, and names to the learner’s *memories* for further review.

These definitions and sample sentences are dynamically generated using OpenAI’s GPT-3.5 API and then translated into the TL via Google Translate.

Prompt template. LangEye uses a standardized prompt designed for beginner learners to generate consistent and level-appropriate definitions and examples:

You are a language tutor for beginner French learners. Given the following word: <object-label>, provide: (1) a clear, beginner-friendly English definition of the object, and (2) a short example sentence using the object in everyday context. Make both responses simple and age-appropriate for learners at A1–A2 level.

See Table 1 for an output example: **cup** → *tasse*.

Table 1: Example of a vocabulary memory generated for cup.

Label	cup
Definition	A cup is a small container used for drinking.
Sentence	She drank tea from her favourite cup.
French	<i>Une tasse est un petit récipient utilisé pour boire. Elle a bu du thé dans sa tasse préférée.</i>

This structure supports flexible and learner-driven study sessions, allowing learners to create and engage with memories at their own pace, based on what they encounter in their daily environments. By combining visual, textual, and auditory modalities, LangEye supports deeper encoding of vocabulary through multiple channels of reinforcement. This two-stage generation pipeline balances personalization and control: GPT-3.5 generates beginner-

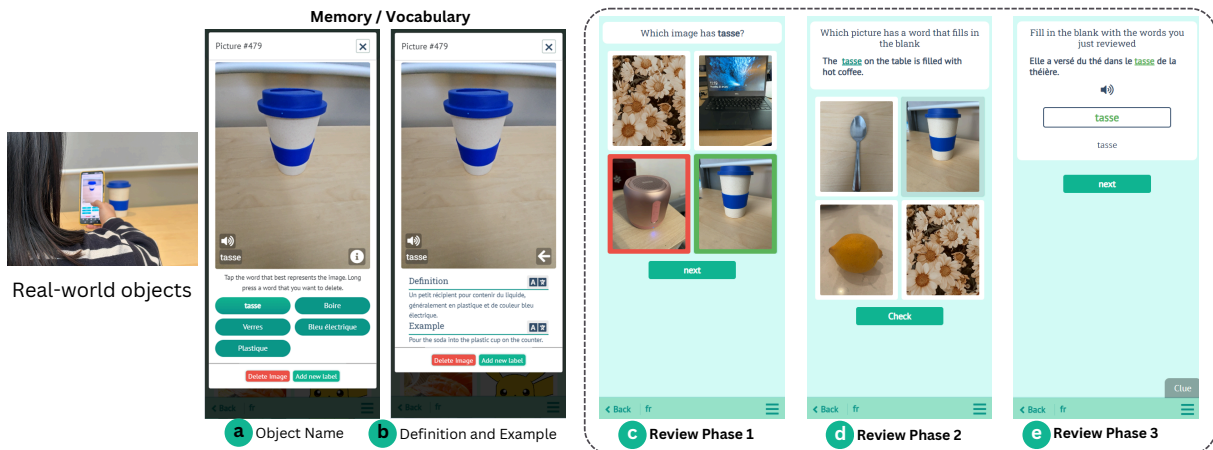


Figure 2: Memories are created from real-world objects. (a) A memory contains the name of the detected object (cup: *tasse* in French) and its pronunciation. (b) Additional information, including a definition and a sample sentence, can be displayed. This content is dynamically generated using OpenAI’s GPT-3.5 API and translated into the learner’s target language (here, French) via machine translation. Learners can practice their vocabulary through the *Memory Review* structure, which includes three progressively challenging phases: (c) **Phase 1: Picture Recognition** prompts learners to identify the correct image; feedback is immediate, highlighting the chosen image in green if correct or red if incorrect. (d) **Phase 2: Sentence Completion** requires selecting an image to fill in a blank within a sentence presented in French. (e) **Phase 3: Free Recall** displays a sentence in French and prompts learners to type the object’s name; small typos are accepted, and hints are available.

friendly English definitions and sentences, while Google Translate handles multilingual output. This modular design supports better quality control, simplifies debugging, and ensures broader language support, particularly for low-resource languages where LLMs may struggle with robust translation performance. At the time of system development, GPT-3.5 was the most stable and accessible option for generating consistent content.

Editing labels. To mitigate possible computer vision and translation errors, learners can add their own label to the picture using the *add label* button — see Figure 2 (a, b). This allows learners to input a text label they find more appropriate if the app’s suggestions are insufficient. Learners type the label in their source language, and the app provides the TL translation — eliminating the need to know the TL term. These custom labels appear alongside their generated definition and sentence — see Figure 2 (b). If a label is deemed irrelevant, learners may long-press it to delete it. In the example shown in Figure 2 (a), the label “*Bleu Électrique*” (Electric Blue) refers to the colour of the cup; the learner might find it unrelated and choose to remove it.

3.2 Languages Supported

LangEye currently supports the following source or target languages: English, French, Spanish, and Portuguese. The *source language* (SL) is the lan-

guage the learner is familiar with or learning from, and the *target language* (TL) is the language to be learned. The interface elements, such as the menu and activity instructions, can be set to any of the listed languages as the SLs. Likewise, the names of objects and learning content (i.e., definitions and example sentences) can be displayed in any listed language as the TL. Learners can define their SL and TL in the *Settings* menu option. This is enabled using machine translation.

3.3 Reviewing Memories

This feature is a classic quiz-style review of the collected memories (vocabulary words). The *Memory Review* has three phases or types of quiz questions, each increasing in difficulty and reducing support. These phases are (1) Picture Recognition, (2) Sentence Completion, and (3) Free Recall. The system chooses N memories (words/objects) to review during a Memory Review session. For each phase, learners are tested on those same N words. Learners must complete earlier phases to unlock later ones, but they may choose to end the session between phases. This progressive design aims to gradually increase cognitive load and promote long-term retention by reinforcing vocabulary through multiple retrieval formats.

Phase 1: Picture Recognition prompts learners to identify an object by selecting the correct picture

from four gallery options (Figure 2 (c)). Feedback is immediate, with correct choices highlighted in green and incorrect in red. Learners have up to four attempts per word, ensuring they review all three target words before moving to the next phase.

In **Phase 2: Sentence Completion**, learners answer “Fill in the blank” questions by selecting a picture that completes an example sentence (Figure 2 (d)). The chosen picture’s word fills the blank, allowing reflection before submission. Incorrect answers reveal the correct choice, and sentences are shown in the SL to support beginners. However, this design may limit advanced learners who prefer tasks entirely in the TL.

Phase 3: Free Recall introduces open-ended vocabulary production. Learners type the names of the three target words without visual cues (Figure 2 (e)). A *Clue* button provides definitions if needed, and the system tolerates minor typos, while still highlighting the correct spelling for feedback. Sentences are now in the TL, catering to advanced learners and promoting grammar understanding.

This phased approach bridges the gap between beginner and advanced learners, enabling gradual mastery of vocabulary and TL proficiency. See Table 2 for a feature and language comparison of all three phases.

3.4 Tracking Vocabulary Learning Progress

The *Achievements* feature in LangEye tracks learners’ Memory Review history and accuracy. For Phase 1, it records the average number of guesses, while for Phases 2 and 3, it calculates accuracy. Learners can sort words by accuracy, with TL initials (e.g., “fr” for French) displayed for context. While these metrics provide insight into learner behaviour and memory usage, they do not directly measure vocabulary acquisition or retention — a challenge we revisit in our discussion of evaluation.

4 User Study

To explore LangEye’s potential as a personalized vocabulary learning tool, we conducted a one-week exploratory study with 20 French (L2) learners. This formative evaluation investigated how learner-curated content and NLP-driven interactions support engagement and vocabulary study in real-world contexts. The study was reviewed and approved by our institution’s Research Ethics Board.

Participants were randomly assigned to two groups:



Figure 3: Sample images taken by participants and their AI-generated counterparts used in the study for the control group.

Control group ($N = 10$): used a version of the app with pre-defined vocabulary and AI-generated content based on pre-curated images; and **Camera group** ($N = 10$): used the full app, including features for taking and uploading images and dynamically generating content through integrated NLP services.

Study Design. The study comprised two sessions: **Session 1:** in-lab training, background survey, and post-session usability feedback. **Session 2:** online exit interview after using the app for at least five days. **Between Sessions:** participants were instructed to use the app daily for five days, completing short usability surveys after each use. Reminder emails were sent daily. Each 50-minute session included surveys, session recordings, and app usage data. Photos taken between study sessions were also collected for the camera group. See Appendix A for detailed information on the study sessions and materials; the semi-structured interview questions are included in the Supp. Material. Figure 3 shows a sample of the generated images used by the control group participants.

Recruitment. Participants were 20 French learners, evenly split into control and camera groups. Participants received \$10 CAD for Session 1 and \$20 CAD for Session 2, recruited through posters in high-traffic campus areas.

Room Setup. The room held eight household objects (an apple, cup, fork, paper, scissors, spoon, sunglasses, and watch) for the camera group to explore and photograph. The control group experienced the same room with objects, but they interacted exclusively with pre-curated memories.

Control Group Memory Curation. Control group memories were curated using data from the cam-

Table 2: Feature and language support comparison across the three Memory Review phases: **Phase 1: Picture Recognition**, **Phase 2: Sentence Completion**, and **Phase 3: Free Recall**. As learners progress through the phases, visual support (e.g., images and multiple-choice options) is gradually reduced, while the use of the target language (TL) increases. This design makes later phases more cognitively demanding. Learners can choose to save and exit the review session at any point between phases.

		Phase 1	Phase 2	Phase 3
Task type	Identify picture		—	—
	Fill in the blank	—		
Answer support	Picture			—
	Multiple choice			—
	Type/Spell	—	—	
	Clue	—	—	Word definition
Answer and corrective feedback	Instantaneous check		—	—
	Submit and check	—		
	Corrective feedback			
Cognitive task	Word recall	Picture	Picture	Spell
	Word collocation	—		
Language <i>Source: Source Language</i> <i>TL: Target Language</i>	UI elements	Source	Source	Source
	Vocabulary word	TL	TL	TL
	Sample sentences	—	Source	TL
	Word definition	—	—	Source

era group to ensure comparability between groups. Camera group participants collected an average of 11 pictures (*Median* = 9; *min* = 5; *max* = 19), resulting in 24 unique objects.

To ensure uniformity and preserve privacy, we generated realistic images of the objects using OpenAI’s Dall-E 3 (full list in Supp. Material). The curated vocabulary ensured consistency without introducing additional biases or privacy concerns. This initial study focused on usability, motivation, and learner perceptions, rather than directly measuring vocabulary acquisition, which is addressed in planned longitudinal follow-ups.

4.1 Study Results: Engagement and Usability

This section presents the user study results, including the pre-session, post-session, exit interview, and software usage data. *Qualitative data analysis*. The questionnaire and the semi-structured interview open-ended questions were coded into categories following commonly mentioned themes in the participants’ answers.

4.1.1 Pre-session background questionnaire

Participants Background. Participants (N=20) were aged 18–24 (N=15) and 25–30 (N=5). All were fluent in English, though only 6 identified it as their first language. Most participants (N=15) self-reported as beginners (A1/A2), with 9 in the camera group. The control group included most ad-

vanced learners (B1/B2; N=4), who reported studying French for over a year. See Appendix B for details. Participants spoke diverse languages, including Tamil (N=5), Hindi (N=4), and Urdu (N=3). Additional languages learned alongside French included Arabic (N=2), Spanish, and Italian, while 11 participants were not learning another language.

Technology for Language Learning. Duolingo was the most commonly used app (N=12), followed by platforms like Udemy and Memrise. Eight participants, mostly in the control group (N=5), reported not using any apps. Only one camera group participant used apps daily, with most others engaging less frequently (once or 3–6 times a week). Mobile devices were the preferred learning platform (N=14), followed by desktop (N=5) and a single participant choosing “either.” Preferences were evenly distributed across groups.

Language Learning Goals. The primary motivation for learning French was career-related (N=11; 7 camera, 4 control), followed by leisure (N=4) and travel (N=3). Other reasons included academic and family goals (each N=1).

4.1.2 Post-Session 1 Feedback and Exit Interview Results

Usability. Ratings for Memories, Review Memories, Achievements, and Picture Mode were measured on a 10-point scale. Participants rated their experience with LangEye after Session 1 (first im-

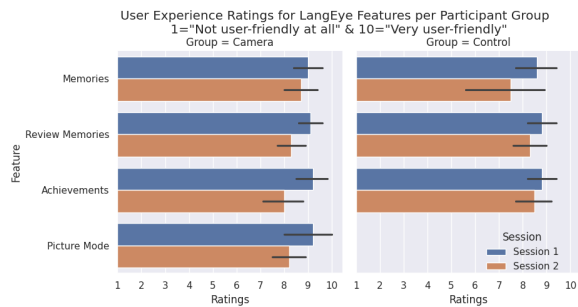


Figure 4: Using a 10-point user experience scale where (1) “Not user-friendly at all” and (10) “Very user-friendly” participants provided ratings on each of four LangEye features: *Memories*, *Review Memories*, *Achievements*, and *Picture Mode* — for camera group only. The chart on the left represents the camera groups and the control group is shown on the right. Overall, Session 1 had higher ratings than Session 2, and the camera group had higher ratings than the control group.

pressions) and Session 2 (after five days of use). *Review Memories* yielded an average of 8.7 for the camera group and 8.6 for the control group. Session 1 scores were higher for both groups compared to Session 2. Overall, ratings dropped from Session 1 to 2 for all features/groups (Figure 4), with the *Memories* (9.0 to 8.7 and 8.6 to 7.5) and *Achievements* (9.2 to 8.0 and 8.8 to 8.5) features showing the most decline for the camera and control groups, respectively. *Picture Mode* ratings, available only for the camera group, averaged 8.7, with Session 1 scoring 9.2 and Session 2 scoring 8.2.

Comfort and Control. Participants expressed comfort using LangEye for both vocabulary review and learning, based on 5-point Likert scale ratings. The camera group (4.3 and 4.2) reported similar comfort levels for both activities, while the control group showed lower comfort (4.0 and 4.2) when learning new vocabulary, likely due to their pre-curated and limited vocabulary set. Four control group participants requested options to expand pre-curated content. Self-efficacy ratings over learning were consistent across groups, with both reporting a mean score of 4.3 and a range of 3—5.

Most and Least Favourite Features. The camera group’s most liked features were *Picture Mode* (4), *Review Memories* (3), and *Memories* (3). Participants appreciated the personalization offered by *Picture Mode*: “[It] allows real-time learning with objects around me” (P8, A2). The control group preferred *Review Memories* (8), with Phase 1: *Picture Recognition* praised for its simplicity. The least liked features for the camera group in-

cluded Phase 3: *Free Recall* (3), Phase 2: *Sentence Completion* (2), and *Achievements* (2), with some noting confusing sentences in Phases 2 and 3. The control group disliked *Achievements* (4), citing low interactivity.

Motivation. Self-reported motivation levels, however, showed divergence. The camera group maintained a steady mean of 4.3 across sessions, while the control group’s mean dropped from 4.1 in Session 1 to 3.8 in Session 2. This decline may reflect lower engagement with pre-curated content. These findings underscore the benefits of learner-curated content in enhancing comfort, control, and sustained motivation. They also suggest that giving learners agency to drive the content creation process—supported by generative NLP—can foster deeper engagement compared to static, pre-defined content.

Learner Perceptions Compared to Other Tools.

Learning with Pictures. Participants valued LangEye’s use of pictures for vocabulary learning, citing improved memorization and contextual association compared to dictionaries: “*Images make it easier to memorize and associate vocab with objects*” (P15, B1). **Self-curated Memories.** Camera group participants praised the personalization and relevance of self-curated content: “[LangEye] uses my own pictures, making vocabulary more memorable” (P3, A1). They suggested combining pre-populated and user-generated content for flexibility. **Multiple Labels per Image.** LangEye’s ability to associate multiple concepts with a single image was seen as helpful for intermediate learners but confusing for beginners: “*Pictures have more context and words, good for intermediate learners*” (P14, A1).

4.2 Thematic Learner Feedback

App reminders and gamification. While some participants appreciated the absence of in-app reminders (P12), others requested daily notifications to encourage engagement (P1, P2). **Aesthetics improvements.** Participants (9/20) recommended a more colourful interface, sound effects for feedback, and larger buttons for easier interaction.

Content customization. Participants valued the use of personalized images, citing improved memory and relevance. A camera group participant noted, “*This app adds personal attachment to the picture, making it easier to remember*” (P8).

AI-generated content and robustness. Participants reported object detection errors and overly technical definitions. Cluttered backgrounds and

multiple objects caused incorrect labels, while some sentences had mismatched vocabulary contexts. For instance, “wood” (noun) was replaced with “wooden” (adjective). Participants suggested cropping tools and improved AI prompts to reduce errors.

Review memories design. Beginners (A1) preferred Phase 1: Picture Recognition and Phase 2: Sentence Completion but found Phase 3: Free Recall overwhelming, while advanced learners preferred Phase 3 in the target language (TL). Participants generally praised the multi-phase system for its progressive difficulty. P9 said, “*I like the three phases, but the second phase in English was not helpful due to gender issues.*”

Language-specific considerations. Participants highlighted issues with gendered nouns in French during Phase 2: Sentence Completion. Gender information was lost in English translations, causing confusion, especially for A2–B2 learners. Suggestions included displaying gender indicators (e.g., P1, P8, P9).

4.3 Daily Usage and Technical Issues

Figure 5 visualizes the decline in daily feedback form submissions across the five study days. The control group maintained more stable participation, while the camera group showed a sharper drop-off, despite initially similar engagement levels. This suggests that while learner-curated content may drive early motivation, maintaining sustained engagement over time remains a challenge. Overall ratings for ease of use, engagement, and vocabulary learning were mostly positive, leaning toward “Strongly agree” or “Neutral.” However, control group ratings for learning new words were lower, likely due to limited pre-curated vocabulary. *Technical Issues* 26% of submissions reported technical difficulties, including delays in label loading over mobile networks and incorrect object labels. Sentence quality was another concern, as participants noted that some sentences were contextually incorrect or mismatched vocabulary. The detailed data is available in B.1.

5 Discussion

This exploratory study demonstrates the promise and challenges of combining learner-curated content with generative AI to support vocabulary learning in mobile contexts. While our goal was not to directly measure vocabulary gains, the findings

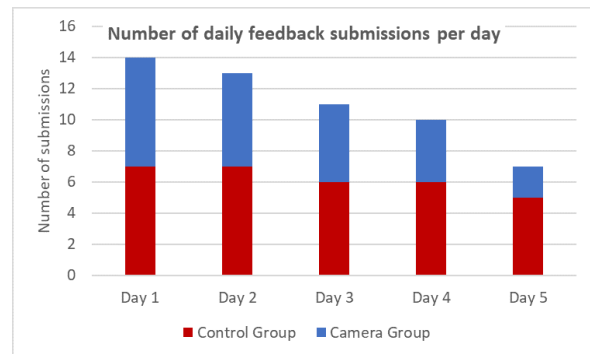


Figure 5: Number of daily feedback form submissions across the five-day study. While total submissions began at 14 on Day 1, they declined to 7 by Day 5. The control group’s submissions remained relatively stable (7 to 5), whereas the camera group showed a sharper decline (7 to 2). Participant ratings per day are available in B.1.

offer formative insights into learner experience, system usability, and the design trade-offs inherent to NLP-enhanced educational tools. Below, we reflect on key lessons learned and identify opportunities for future improvement.

Evaluation and Research Implications. Future work will incorporate longitudinal vocabulary tracking and explore adaptive evaluation strategies aligned with learner-curated content. Because LangEye supports open-ended, learner-defined content creation, traditional pre- and post-testing are difficult to apply consistently. Even usage-based metrics, such as phase completions or accuracy scores, are complicated by the variability in content difficulty and prior learner knowledge. These challenges reflect broader tensions in evaluating personalized, generative learning systems and call for alternative strategies such as learner modelling or adaptive diagnostics.

Balancing Pre- and Self-Curated Content. Learner-curated vocabulary fosters autonomy and engagement but also introduces variability in vocabulary scope and difficulty. A hybrid approach—integrating structured, pre-curated content alongside learner-generated memories—may better support novice learners while preserving personalization for advanced users. This balance also enhances scalability across languages without requiring expert-authored corpora.

Tensions in AI-Generated Content. Although generative AI enabled dynamic and personalized vocabulary entries, participants frequently encountered issues such as overly technical definitions, inappropriate word senses, and context mismatches.

For example, “wood” was rendered as “wooden” (Table 3), and cluttered images led to irrelevant labels. These issues reflect broader limitations of prompt-based generation in educational contexts. Future iterations will incorporate prompt tuning, simpler output targets, and human-in-the-loop validation to improve robustness and learner alignment.

Table 3: Example of a vocabulary memory with a word sense mismatch due to ambiguous object labelling.

Label	wood
Definition	Wooden means made of wood.
Sentence	I sat on the wooden chair.
Issue	Learner expected a noun definition for wood, but GPT returned the adjective form wooden.

Language-Specific Considerations. LangEye supports multiple target languages via machine translation; however, users of gendered languages (e.g., French) have noted grammatical issues, particularly in Phase 2, where translations often lack gender agreement. This suggests the need for grammar-aware translation strategies and visual indicators for noun gender, especially in beginner-focused review phases.

Our use of a hybrid generation pipeline, employing GPT-3.5 for English definitions and Google Translate for multilingual output, was driven by a need for modularity, consistency, and broad language coverage. This approach provided control over linguistic complexity in the initial prompt while leveraging production-grade translation tools for low-resource languages, where LLM performance remains less benchmarked. This modular architecture proved essential for supporting LangEye’s multilingual scope but also contributed to mismatches and errors in translated content, underscoring the importance of future refinement in prompt tuning and translation alignment.

Learner Engagement and Personalization. Learners consistently emphasized the motivational value of interacting with vocabulary grounded in their own environment. This supports situated learning theory and highlights how self-curated images can improve recall by reinforcing personal relevance. However, engagement declined over time—particularly in the camera group—suggesting a need for better pacing, reminders, or gamified retention mechanisms to sustain interest.

Review System Calibration. Participants appre-

ciated the phased review design, but feedback suggests the need for difficulty calibration. Phase 3: Free Recall was overwhelming for beginners, while some advanced learners desired more TL immersion earlier. Dynamically adapting review complexity based on learner level and behaviour (e.g., accuracy, completion history) may improve retention and reduce frustration.

Toward Context-Aware Learning Scenarios. LangEye’s current design centers on object-driven vocabulary. Future iterations could support more dynamic interactions, such as context-aware prompts, adaptive content sequencing, and gamified activities (e.g., real-world “treasure hunts”). These enhancements—combined with more accurate object detection (e.g., YOLO-E)—could transform LangEye into a broader platform for situated, task-based language learning.

6 Conclusion

This paper presented LangEye, a mobile language learning application that leverages generative NLP and learner-curated content to support contextual vocabulary acquisition. By combining object recognition, machine translation, and dynamic content generation, LangEye enables self-directed learners to engage in personalized, real-world language practice. Findings from our exploratory study highlight the system’s usability, motivational benefits, and learner preference for personalized visual content.

While this formative evaluation did not assess vocabulary acquisition directly, the results inform design implications for learner-driven, AI-enhanced educational tools. Future work will include longitudinal studies to track learning outcomes, and expand LangEye’s capabilities through adaptive review difficulty and improved language-specific support. Additionally, we envision incorporating more accurate computer vision models (e.g., YOLO-E) to enable dynamic, context-aware interactions such as real-world object “treasure hunts” or live situational vocabulary tasks, further bridging the gap between everyday experiences and language learning.

Limitations

This work has several limitations that inform the scope of its findings and highlight directions for future research.

First, this was an exploratory and short-term

study focused on learner engagement and usability. While participants interacted with generative NLP features and learner-curated content, we did not directly assess vocabulary acquisition or retention through pre- and post-testing. Future studies with longer durations and individualized baseline assessments are necessary to evaluate learning outcomes rigorously.

Second, the evaluation was constrained by the personalized nature of the learner-curated content. Since learners selected their own vocabulary items, it was not feasible to apply a standardized test or compare vocabulary gains across participants. While this personalization is central to LangEye's design, it introduces challenges for controlled, quantitative evaluation.

Third, the generative NLP components (e.g., definitions, sample sentences) sometimes produced inconsistent or overly complex outputs. This was especially problematic for beginner learners, who occasionally found definitions too advanced or mismatched in word sense. Our system relies on prompt-based content generation, which can be brittle without careful tuning and contextual awareness. While we did not run expert benchmarking of the AI-generated content in this pilot, this remains an important step for future work, especially for language education applications."

Finally, although LangEye supports multiple languages, our study only examined English–French learners. Language-specific features—such as grammatical gender—presented challenges in the translation pipeline and feedback design, limiting generalizability across linguistic contexts. Further studies should explore broader language pairs and adapt the system to handle grammar-sensitive features more effectively.

Ethical Considerations

This study was reviewed and approved by our institution's Research Ethics Board (REB). All participants provided informed consent prior to participation and were compensated for their time. Data collected during the study, including app usage logs and participant feedback, was anonymized prior to analysis.

To protect participant privacy, especially in the camera group, no personally identifying photos were stored or analyzed. For the control group, object images were generated using OpenAI's DALL·E 3 to avoid the use of participant-provided

media.

LangEye integrates generative AI tools (e.g., GPT-3.5, Google Translate) to produce multilingual learning content. While this automation enables scalability, care was taken to limit content generation to isolated vocabulary contexts, and the system does not store user data beyond local app sessions. Limitations of AI output—such as occasional mismatches in word sense—were disclosed to participants, and learners had full control over which content to save and review.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

- Abdulhameed A Alhuwaydi. 2022. A review on vocabulary learning-designed mall applications in the efl context. *Theory and Practice in Language Studies*, 12(10):2191–2200.
- Antonie Alm. 2021. Language learning with netflix: extending out-of-class l2 viewing. In *2021 International Conference on Advanced Learning Technologies (ICALT)*, pages 260–262. IEEE.
- Riku Arakawa, Hiromu Yakura, and Sosuke Kobayashi. 2022. Vocabencounter: Nmt-powered vocabulary learning by presenting computer-generated usages of foreign words into users' daily lives. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–21.
- Jason Byrne. 2019. Anytime autonomous english mall app engagement. *International Journal of Emerging Technologies in Learning (IJET)*, 14(18):145–163.
- Alexander Campolo and Kate Crawford. 2020. Enchanted determinism: Power without responsibility in artificial intelligence. *Engaging Science, Technology, and Society*.
- Isabeau Fievez, Maribel Montero Perez, Frederik Cornil-lie, and Piet Desmet. 2023. Promoting incidental vocabulary learning through watching a french netflix series with glossed captions. *Computer Assisted Language Learning*, 36(1-2):26–51.
- Nicola Galloway and Rachael Rugg. 2020. The provision of student support on english medium instruction programmes in japan and china. *Journal of English for Academic Purposes*, 45:100846.
- Ari Hautasaari, Takeo Hamada, Kuntaro Ishiyama, and Shogo Fukushima. 2019. Vocabura: A method for supporting second language vocabulary learning

- while walking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–23.
- Shanna Li Ching Hollingworth and Wesley Willett. 2023. Fluencyar: Augmented reality language immersion. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–3.
- Wu-Yuin Hwang, Holly SL Chen, Rustam Shadiev, Ray Yueh-Min Huang, and Chia-Yu Chen. 2014. Improving english as a foreign language writing in elementary schools using mobile devices in familiar situational contexts. *Computer assisted language learning*, 27(5):359–378.
- Sook-Hi Kang. 1995. The effects of a context-embedded approach to second-language vocabulary learning. *System*, 23(1):43–55.
- Blanka Klimova. 2021. Evaluating impact of mobile applications on efl university learners’ vocabulary learning—a review study. *Procedia Computer Science*, 184:859–864.
- Joanne Leong, Pat Pataranutaporn, Valdemar Danry, Florian Perteneder, Yaoli Mao, and Pattie Maes. 2024. [Putting things into context: Generative ai-enabled context personalization for vocabulary learning improves learning motivation](#). In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI ’24, New York, NY, USA. Association for Computing Machinery.
- Kelvin Leong, Anna Sung, David Au, and Claire Blanchard. 2020. A review of the trend of microlearning. *Journal of Work-Applied Management*, 13(1):88–102.
- Zhenhui Peng, Xingbo Wang, Qiushi Han, Junkai Zhu, Xiaojuan Ma, and Huamin Qu. 2023. Storyfier: Exploring vocabulary learning support with text generation models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–16.
- Azizova Fotimakhon Saidbakhramovna, Rahmatova Nargiza Valijonvna, and Kurbanbayeva Dilnoza Sharofidinovna. 2021. The method of educating vocabulary in a foreign language or target language. *Linguistics and Culture Review*, 5(S1):1649–1658.
- Nguyen Tran, Shogo Kajimura, and Yu Shibuya. 2023. Location-and physical-activity-based application for japanese vocabulary acquisition for non-japanese speakers. *Multimodal Technologies and Interaction*, 7(3):29.
- Stephen Tsourounis and C Demmans Epp. 2016. Learning dashboards and gamification in mall: Design guidelines in practice. *The international handbook of mobile-assisted language learning*, pages 370–398.
- Aaditya Vaze, Alexis Morris, and Ian Clarke. 2024. Curiosityxr: Context-aware education experiences with mixed reality and conversation ai. In *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, pages 41–49. IEEE.
- Christian David Vazquez, Afika Ayanda Nyati, Alexander Luh, Megan Fu, Takako Aikawa, and Pattie Maes. 2017. Serendipitous language learning in mixed reality. In *Proceedings of the 2017 CHI conference extended abstracts on human factors in computing systems*, pages 2172–2179.

A Research Methods

A.1 Control Group Memory Curation

One of the main challenges in comparing the two groups of participants is to curate the control group’s memories. As discussed in section 4, our approach to this problem was to run both sessions of the study with the camera group first. This allowed us to use the collection of memories created by that camera group as the control group’s memories. On average, the camera group collected 11 pictures ($Median = 9; min = 5; max = 19$). The aggregation of duplicates of the camera group memories resulted in a total of 24 objects (memories) listed below:

apple	floor	lemon	speaker
bag	flooring	paper	spoon
cup	food	peripheral	stapler
dishware	fork	scissors	tableware
drinkware	glasses	serveware	watch
eyewear	hand	slipper	wood

Recalling that, to create uniformity in the images, avoid bias toward the quality of images taken by the camera group, and preserve the participants’ privacy, OpenAI’s Dall-E 3 was used to create the images for each of these memories. The prompt included the object name and instructions for the illustration to be “realistic” to mimic a photo taken of the object — sample shown in Figure 3. The pairs of all AI-generated images and labels can be found in Supplemental Materials. This approach was used to present a similar curation of vocabulary while not adding words that might not have been added using a smartphone camera.

A.2 Detailed Study Sessions

A.2.1 Study Session 1: Introducing LangEye

Session 1 was conducted in a controlled lab environment with separate setups for the camera and control groups.

Room Setup. The room featured a collection of eight household objects (see Figure 6) for the camera group to explore and photograph. The control group experienced the same room setup but interacted exclusively with pre-curated memories.

Pre-Session Questionnaire. Participants completed a brief background survey about their French language learning experience and use of language learning apps.



Figure 6: Top: Room setup with video recording. Bottom: Objects available for the camera group to explore and create memories.

Training Tasks. Participants were introduced to the app’s features through a demonstration and a printed tutorial. Both groups explored the app’s main features, with the control group focusing on editing pre-curated memories and the camera group using the camera mode to create their own. Participants could ask questions during the session and were required to interact with each feature before proceeding to the post-session survey.

Post-Session Questionnaire. Participants evaluated LangEye’s usability and practicality, providing feedback on the app’s usefulness for language learning.

A.2.2 Between Sessions

Participants were instructed to use LangEye daily for five days between Sessions 1 and 2. Daily reminder emails prompted them to complete a short feedback form covering usability, error reporting, and general app impressions. The second session was scheduled 5–10 days after the first.

A.2.3 Study Session 2: Exit Semi-Structured Interview

In Session 2, participants reflected on their experiences with LangEye, discussing usability, vocabulary acquisition, and the accuracy of object recognition and labeling. The camera group shared insights on creating memories, while the control group focused on pre-curated content. Interviews were recorded using Google Meet, capturing video, audio, and transcripts. Transcripts were reviewed

Table 4: Summary of participants' background information per study group: camera and control.

Attribute	Camera	Control	Total
Age			
18–24 years old	6	9	15
25–30 years old	4	1	5
L1			
Other	8	6	14
English	2	4	6
French level			
A1	5	5	10
A2	4	1	5
B1	–	4	4
B2	1	–	1

for accuracy and used alongside structured session notes for qualitative analysis of participant responses.

B Results Data Visualizations

Visual representations of some of the results are available in this section. Table 4 shows the tabulated participants demographics information. Table 5 shows the tabulated data on participants' French learning background.

B.1 Daily Feedback Submissions

Participants were asked to submit a daily feedback form after using the app in between sessions. While the number of daily submissions (Figure 5) remained somewhat stable for the control group (from 7 to 5), the camera group had the most decline (from 7 to 2). When aggregating both groups, at Day 1 there were 14 submissions, which was reduced to 7 at Day 5. The charts in Figures 7 and 8 show the participants' ratings (5-Point Likert scale for agreement) per day. The difference in the volume of submissions makes it difficult to compare across groups, but overall, the ratings lean toward "Strongly agree" to "Neutral" throughout the study days. Here are the statement items participants were asked to rate:

- "Overall, this app is easy to use."
- "I'm having fun using this app."
- "I have learned new French words using this app."

Table 5: Table shows participants' main method for learning French and the duration of their studies. The study was run in Canada, a bilingual country (English and French are official languages). Thus, French immersion schools are commonly available in Canadian education. Courses for French ("Course at school") as a foreign language are also common in Anglophone schools. And other "French course" or classes are easily accessible in language institutes. Participants who indicated "None" were never enrolled in a course or followed a specific method.

French Study		Camera	Control	Total
Duration	Method			
1 week or less	None	3	1	4
less than 6 months	French immersion	–	1	1
	Online course or resource	1	–	1
	None	1	–	1
1 year+	Course at school	–	1	1
	Online course or resource	1	1	2
5 years+	Course at school	1	5	6
	French course	1	–	1
10 years+	French course	1	–	1
	French immersion	1	1	2

- "I feel more in control of my French vocabulary learning progress and content since using this app."

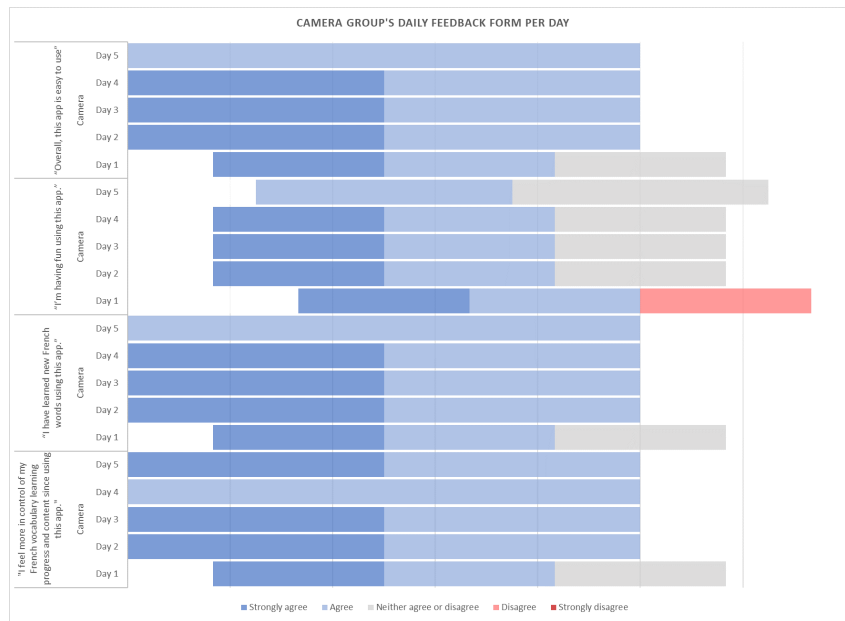


Figure 7: Camera group participants' ratings for the daily feedback form per day. The Figure 5 shows the number of responses per day. While at Day 1 there were 7 submissions, that number declines along the days. This chart shows the distribution of the respondents' answers to the 5-Point Likert scale agreement statement items at each day, from bottom (Day 1) to top (Day 5) at each item. Although there is a shift to "Strongly agree"/"Neutral" as days pass the number of responses are reduced.

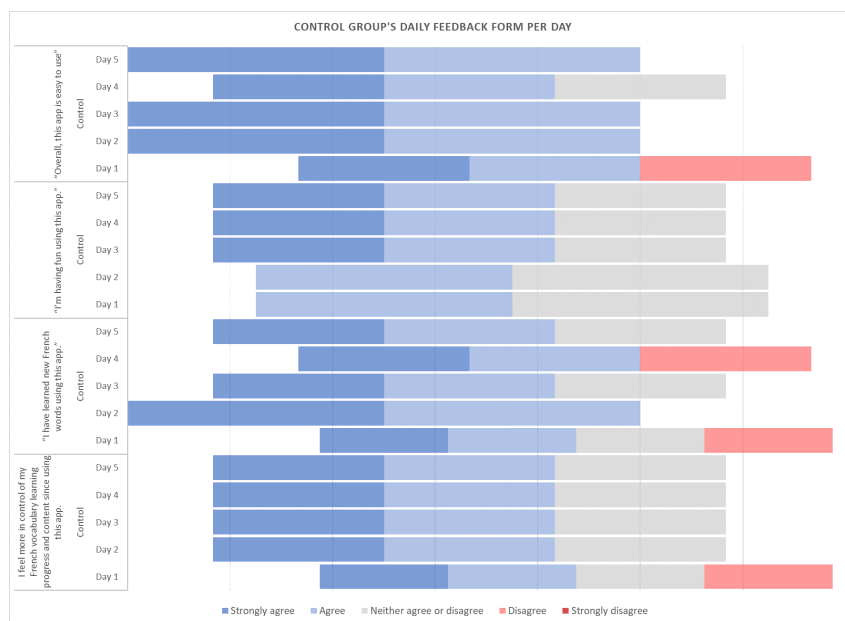


Figure 8: Control group participants' ratings for the daily feedback form per day. The Figure 5 shows the number of responses per day. While at days 1 and 2 there were 7 submissions, that number declines to 5 at Day 5; which is higher than the camera group's Day ($N = 2$). This chart shows the distribution of the respondents' answers to the 5-Point Likert scale agreement statement items at each day, from bottom (Day 1) to top (Day 5) at each item. Although there is a shift to "Strongly agree"/"Neutral" as days pass the number of responses are reduced.

Costs and Benefits of AI-Enabled Topic Modeling in P-20 Research: The Case of School Improvement Plans

Syeda Sabrina Akter¹, Seth B. Hunter², David S. Woo³, Antonios Anastasopoulos^{1,4}

¹Department of Computer Science, George Mason University

²Department of Educational Leadership and Policy, George Mason University

³Department of Educational Leadership and Policy, The University of Utah

⁴Archimedes, Athena Research Center, Greece

sakter6@gmu.edu

Abstract

As generative AI tools become increasingly integrated into educational research workflows, large language models (LLMs) have shown substantial promise in automating complex tasks such as topic modeling. This paper presents a user study that evaluates AI-enabled topic modeling (AITM) within the domain of P-20 education research. We investigate the benefits and trade-offs of integrating LLMs into expert document analysis through a case study of school improvement plans, comparing four analytical conditions. Our analysis focuses on three dimensions: (1) the marginal financial and environmental costs of AITM, (2) the impact of LLM assistance on annotation time, and (3) the influence of AI suggestions on topic identification. The results show that LLM increases efficiency and decreases financial cost, but potentially introduce anchoring bias that awareness prompts alone fail to mitigate.¹

1 Introduction

Educators are increasingly turning to artificial intelligence to streamline research and administrative workflows, particularly within P-20 contexts, which cover education from Pre-K through graduate levels and workforce training. It has sparked considerable interest in the potential of generative AI tools to tackle complex analytical tasks (Kasneci et al., 2023; Xu et al., 2024). Among these applications, topic modeling (TM)—a method for uncovering hidden themes in unstructured text—has become a prominent technique in P-20 research over the past decade (Brookes and McEnery, 2019; Daenekindt and Huisman, 2020; Sun et al., 2019; Wang et al., 2017). In contrast to conventional text

data analysis (CTDA), which often requires substantial human input and can be constrained by its labor-intensive nature, subjectivity, and potential for inconsistency, Artificial Intelligence-enabled Topic Modeling (AITM), driven by sophisticated LLMs like GPT-4 (OpenAI et al., 2024), holds the potential for significant improvements in efficiency and scalability by automating or assisting with these demanding procedures (Dell'Acqua et al., 2023; Grossmann et al., 2023).

Implementing AITM offers several key benefits, notably a reduction in the time required for analysis and the potential for more consistent and thorough topic identification. These efficiencies can significantly influence research productivity and, importantly, may lead to qualitatively different research findings compared to CTDA due to variations in identified themes. Nonetheless, the rapid adoption of AITM raises important concerns about potential drawbacks, such as financial costs and environmental impacts associated with substantial computational resource utilization. At present, there is a lack of empirical research that compares these costs to those of traditional methods, especially within the field of K12 educational research.

Another critical but underexplored concern with AITM is the psychological phenomenon known as anchoring bias—the tendency for humans to rely excessively on initially presented information when making subsequent judgments or decisions (Nagtegaal et al., 2020). In contexts where humans interact with AI-generated insights, anchoring bias may skew human analysts' judgments, thus, affecting the final research outcomes (Zhao et al., 2024; Choi et al., 2024).

Given these critical gaps, we investigate the financial, environmental, cognitive, and analytical trade-offs of integrating AITM into P-20 research. Our case study focuses on principal-written school improvement plans (henceforth "Plans") from a formal field-based principal evaluation sys-

¹Code available [here](#). The data are not publicly available due to privacy restrictions but can be requested through the Network for Educator Effectiveness (NEE) at the University of Missouri and the Missouri Department of Elementary and Secondary Education (DESE).

tem in hundreds of K12 districts in the Midwest USA. We systematically evaluate four analytic conditions: AI-Only, Human-Only, AI-Human, and AI-Human-Deanchoring. Through this comparative analysis, we address three research questions:

- **RQ1:** What are the marginal financial and environmental costs of implementing AITM in P-20 research?
- **RQ2:** What are the causal effects of different analytic approaches on analysis time?
- **RQ3:** What are the causal effects of these analytic approaches on the topics identified?

Preliminary findings suggest that AI analysis significantly reduces costs and analysis time per document compared to human analysis, although AI-assisted methods vary slightly in terms of speed. Additionally, when humans and AI were provided with pre-specified topic lists, only minor differences emerged in the topics identified. Through a thorough evaluation of these aspects, we aim to offer an empirical understanding of AITM’s value proposition for P-20 educational research.

2 Related Work

The field of topic modeling has seen significant advancements, moving from traditional probabilistic methods to more contemporary AI-driven techniques. Early models, such as Latent Dirichlet Allocation (LDA; Blei et al., 2003), conceptualized documents as combinations of topics, with each topic characterized by a distribution of words. While widely adopted, LDA and similar approaches often required substantial manual interpretation, as they yielded clusters of words without clear semantic labels (Gao et al., 2024b). Subsequent neural network-based models, like BERTopic (Grootendorst, 2022), improved the coherence of topics by leveraging transformer embeddings that capture richer contextual meaning. More recently, frameworks leveraging large language models (LLMs), such as TopicGPT (Pham et al., 2024), have further enhanced the accessibility and interpretability of topic modeling by generating human-readable topic labels and summaries (Overney et al., 2024; Gao et al., 2024a).

Within educational research, topic modeling has proven to be a powerful tool for analyzing large-scale textual data, such as curricula, school improvement plans, and scholarly literature. Studies have applied topic modeling to uncover latent

themes in educational leadership, policy discourse, and reform strategies (Wang et al., 2017; Sun et al., 2019; Daenekindt and Huisman, 2020). These methods claim to significantly reduce the labor associated with traditional qualitative coding, making large-scale analysis more scalable and helping to address a fundamental impediment to research use by educators: the amount of time it takes to conduct research (Drahota et al., 2016; Asmussen and Møller, 2019).

As AI tools, particularly LLMs, become more prominent in education research and practice, they are being increasingly adopted for tasks such as writing content, analyzing student responses, or synthesizing research findings (Liu and Wang, 2024; Cambon et al., 2023; Jaffe et al., 2024). However, effective adoption in educational contexts requires addressing the environmental and financial costs of model training and inference (Strubell et al., 2019; Hershcovich et al., 2022), challenges around the reliability and interpretability of model outputs (Mittelstadt et al., 2016; Sahoo et al., 2024), and cognitive pitfalls such as automation and anchoring bias that may skew human judgment during analysis (Goddard et al., 2012; Koo et al., 2024; Echterhoff et al., 2024). This is particularly concerning in high-stakes domains like education, where premature reliance on AI-suggested outputs can limit critical thinking, reduce analytical diversity, and ultimately affect the integrity of findings (Al-Zahrani, 2024; Sallam, 2023).

Furthermore, bias mitigation remains a pressing challenge. LLMs have been shown to inherit and sometimes amplify social and cultural biases (Resnik, 2024). Interestingly, emerging research suggests that strategies such as structured group discussions and collaborative review can counteract some of these effects, promoting more balanced and reflective decision making in AI-assisted workflows (Horst et al., 2019; Rachael A. Hernandez and Teal, 2013; Michaelsen et al., 2002).

3 Data

We use a proprietary dataset from the Network for Educator Effectiveness (NEE), an educator evaluation system widely implemented across K–12 school districts in Missouri. This dataset spans the academic years 2005–2006 through 2022–2023 and comprises de-identified, text-based portfolios authored by school principals. These documents, formally known as *Building Improvement Plans*

B. Major Objectives and Strategies of the Plan	
Goal Alignment (Element 3)	
Objective - May use a SMART approach. Use a reasonable number of objectives that can be adequately monitored and completed.	
1. Student engagement will increase by 15% as measured by the NEE School Peer Observation form from the beginning of the year data collection to the end of the year form. 2. Student proficiency in the area of ELA standard RI 9 will increase by 15% as measured by the beginning of the year to end of the year STAR report. 3. Math MAP scores in the area of problem solving and communicating reasoning will increase by 10% as measured by the 2019 MAP test to the 2020 MAP test. 4. Student proficiency in the area of science will increase from 47.7% to 57.7% as measured by 2020 MAP data. 5. Student proficiency in the area of reading fluency will increase from 65% to 75% by April 2020 as measured by DIBELS.	CSP Goals Goal 1 - Develop and enhance quality educational/instructional programs to improve performance and enable students to meet their personal, academic, and career goals.
Baseline Data (Element 4)	
What baseline data was used to measure progress toward this objective?	
We identified data gaps in MAP. Math specifics related problem solving, modeling, and data analysis and communicating reasoning (second lowest area). In ELA, third grade listening was the lowest. In fourth and fifth grades, it was reading, but in fifth grade in particular research was the lowest with an average 35%.	
DIBELS data was utilized to determine the need for improved fluency for our students to be successful readers.	
In the area of Science, we used MAP results to determine that we need to address the scientific inquiry process. This is the lowest performance area for our students and it impacts most of the science content. The next lowest performance areas by standards were properties and principles of matter and energy and properties and principles of force and motion.	

Figure 1: The two elements extracted from the Building Improvement Plans (BIPs) used in our goal-based study.

(BIPs) or *School Improvement Plans*, are submitted annually as part of a standardized evaluation process and are structured around seven performance criteria (referred to as *elements*) evaluated by principal supervisors using a consistent rubric.

For our study, we randomly selected 23 BIPs and focused on two specific elements from each plan: (1) the major objectives stated for school improvement, and (2) the data principals planned to use to measure progress toward those objectives (Figure 1). These elements are highly relevant to evaluating strategic goal-setting and progress tracking in educational leadership and K12 school improvement. The documents are entirely text-based and machine-readable, making them ideal for qualitative analysis via topic modeling.

4 Experimental Design

To investigate the integration of LLMs into educational research, we have adapted our methodology from the user study conducted by Choi et al. (2024), which examined the efficiency and precision of LLMs in specialized tasks through a structured user study focused on human-LLM interactions. Their findings showed that while LLMs significantly increased task speed, they also led users to anchor on AI-provided suggestions. Informed by their findings on anchoring bias, we expand on their experimental framework by adding a novel treatment condition: **AI-Human-Deanchoring**. This condition is designed to reduce the over-reliance on LLM by making participants explicitly aware of potential anchoring effects in LLM-generated suggestions (see Figure 2).

Our study is structured in two stages:

- **Stage 1: Topic Discovery**, in which participants identify and curate a list of topics from a shared set of BIPs.
- **Stage 2: Topic Assignment**, in which participants apply those topics to a new set of documents under controlled conditions.

Document	A1	A2	A3	A4	A5	A6
<i>Stage 1</i>						
D1–D11	T2	T3	T4	T2	T3	T4
<i>Stage 2</i>						
D12–D15	T2	T3	T4	T2	T3	T4
D16–D19	T3	T4	T2	T3	T4	T2
D20–D23	T4	T2	T3	T4	T2	T3

Table 1: Document assignments for Stages 1 and 2. In Stage 1, each analyst analyzed the full set of 11 documents (D1–D11) under a single assigned condition; experimental conditions are defined as T2: Human-Only, T3: AI-Human, and T4: AI-Human-Deanchoring. In Stage 2, analysts analyzed documents D12–D23, assigned in a balanced design across all experimental conditions to ensure multiple annotations per document.

We designed the following four treatment conditions:

1. **AI-Only:** Tasks were performed solely by the LLM without human intervention, providing a benchmark for AI performance.
2. **Human-Only:** Participants performed tasks without any AI assistance, serving as the baseline for human performance.
3. **AI-Human:** Participants received suggestions from an LLM before performing tasks, allowing us to assess the influence of AI assistance.
4. **AI-Human-Deanchoring:** Participants were presented with LLM-generated suggestions with explicit instructions to be skeptical of them due to potential anchoring bias. By encouraging participants to thoughtfully evaluate and adjust AI-generated recommendations, we aim to improve the trustworthiness and credibility of AI-generated results.

To assign treatment conditions in the 12 school improvement plans (BIPs) in stage 2, we used a Latin square design (Montgomery, 2017). Each of the six human participants was assigned a specific sequence of treatment conditions across different plans, ensuring a balanced and systematic distribution of the Human-Only, AI-Human, and AI-Human-Deanchoring settings (see Table 1). Analysts proceeded in the order of conditions T2 → T3 → T4 in stage 2. Participants in the AI-assisted settings (T3, T4) were provided with LLM-generated topic annotations, while those in the Human-Only setting (T2) worked independently without any AI input. Analysts were unaware of the condition until they accessed the designated docu-

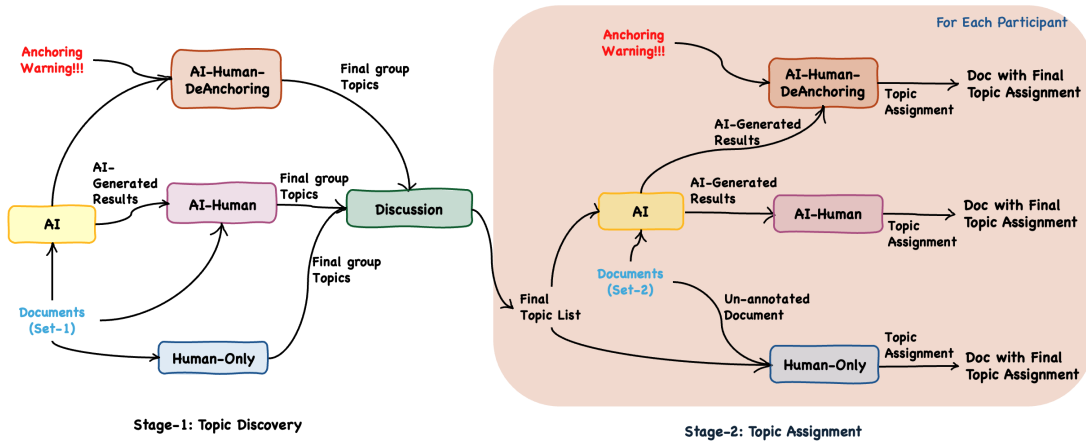


Figure 2: Overview of our Topic Modeling workflow and experimental settings. Stage-1: Topic Discovery involves discovering latent topics within documents. Team discussion occurred at the end of Stage-1 in order to develop the Final Topic List. Stage-2 involves assigning topics to a different set of documents in all treatment conditions.

ment in Label Studio (Tkachenko et al., 2020) that we used to conduct our user study. Each analyst was asked to indicate whether each topic t from the Final Topic List appeared in each paragraph/field f of the assigned BIPs.

4.1 Stage 1: Topic Discovery

We used the data from Stage 1 to examine how topic lists were generated across different analytic conditions. First, all six participants analyzed the same set of 11 BIPs, working individually under one of three assigned conditions: **Human-Only**, **AI-Human**, or **AI-Human-Deanchoring**, with two participants per condition.

Participants in the AI-Human and AI-Human-Deanchoring conditions were provided with an LLM-generated topic list before beginning their analysis, while only the latter were explicitly warned about potential anchoring effects (see Appendix A for the full instruction). Analysts in the Human-Only condition received no AI input.

Each analyst independently reviewed all 11 BIPs and recorded a preliminary list of topics. After this individual phase, participants met in their respective condition groups for a 30-minute discussion to consolidate their findings into a group-specific topic list. Finally, all six analysts engaged in a 60-minute cross-condition discussion to synthesize the **Final Topic List**, which was later used as the reference framework in Stage 2. All individual and group topic lists are included in Appendix B.

Results In Stage 1, we collected three topic lists: from the Human-Only group, AI-Human group, and AI-Human-Deanchoring group. The analysts unanimously curated a final list of 13 topics after

reviewing all three.

Despite differences in conditions, we observed moderate overlap: six of the 13 final topics (46%) appeared in all three lists, though not always as exact matches. For instance, some themes were phrased differently across settings, such as *Educational Technology* from the AI-Human-Deanchoring list and *Technology Integration* from the Human-Only list which were conceptually merged into a single topic, *Technology Use/Integration*, in the final list. This highlights how interpretive nuance plays a role in topic curation.

Comparing each user-generated list with the AI-Only list revealed systematic differences. The AI-Only list included 8 topics. The Human-Only list had 15, with 4 overlapping (26.67%), while AI-Human and AI-Human-Deanchoring lists identified 8 and 11 topics with 3 (37.5%) and 7 (63.64%) overlapping, respectively. The Human-Only list had a more granular set of topics tailored to the dataset, whereas the AI-Only, AI-Human, and AI-Human-Deanchoring lists tended to include broader, more generic themes that echoed the LLM’s original suggestions. This suggests that the presence of LLM suggestions may have influenced annotators to propose fewer, more AI-aligned topics. In contrast, the final topic list—compiled after collaborative review—shared only 4 of 13 topics (30.77%) with the AI-Only list. This divergence suggests that discussion among annotators helped complement AI outputs by adding nuanced topics that the model did not generate.

Comparing Final Topic List and AI-Only List	# of Topics
Exact topic matches between Final Topic List & AI-Only List	4
Topics Present (or Discovered) in the Final Topic List, but not in the AI-Only List	5
Two or more topics from Final Topic List subsumed under one broader AI-Only topics ²	4
Topics completely discarded by the annotators from AI-Only List	2
Total topics in Final Topic List	13

Table 2: The comparison of the AI-Only List with respect to the Final Topic List shows that there are few topics that the model has failed to cover in its overall generation task.

However, 5 of the 13 topics in the final topic list were not present in the AI-Only list at all (see Table 2). These “missing” topics—such as *Classroom Environment* and *Attendance*—often represented context-specific or nuanced areas that the LLM failed to surface.

Additionally, annotators explicitly discarded two LLM topics, *Education* and *School Improvement Planning*, as overly broad. This further illustrates a recurring pattern: while LLMs are helpful in identifying broad thematic content, they may struggle with generating the fine-grained, action-relevant topics that human experts prioritize in education policy contexts. These findings are consistent with prior work by Choi et al. (2024), which similarly highlighted LLMs’ limitations in capturing nuanced, context-specific insights.

4.2 Stage 2: Topic Assignment

Participants used the Final Topic List to annotate a new set of 12 BIPs, each segmented into three paragraph-level fields. Participants were randomly reassigned to one of the three human-in-the-loop conditions. Those in AI-Human and AI-Human-Deanchoring received LLM-generated topic suggestions; those in Human-Only did not. We recorded the time spent on each document to facilitate an efficiency analysis.

Results Following Stage 2, we analyzed expert annotations across three conditions: Human-Only, AI-Human, and AI-Human-Deanchoring. Each paragraph in the dataset was represented as a 14-element vector—13 corresponding to topics from the Final Topic List established in Stage 1, and

²Multiple Final Topic List entries (e.g., “Academic Assessments” and “Academic Goals”) were grouped under a single LLM topic (e.g., “Student Assessment and Achievement”)

Metric	Human-Only	AI-Human	AI-Human-Deanchoring
Avg Precision	0.68	0.84	0.83
Avg Recall	0.55	0.69	0.67
Avg Annotation Speed (words/min)	73.75	71.15	89.91
Avg Annotator Agreement with AI-Only (%)	54.64	73.44	71.41
Avg Inter-Annotator Agreement (κ)	0.57	0.71	0.69

Table 3: Summary of Stage 2 results across the three settings. Metrics include annotation speed (words per minute), agreement with LLM outputs (%), and inter-annotator agreement (Cohen’s κ). See Appendix C for detailed results and metrics definitions.

one for “None”—indicating whether annotators assigned relevant topics. This structure allowed us to assess the impact of LLM suggestions on annotation behavior.

Participants used the **Final Topic List** to annotate a new set of 12 school improvement plans (BIPs), each segmented into three paragraph-level fields. Each field was annotated independently by five human analysts, resulting in 12 plans \times 3 fields \times 5 analysts = 180 annotations. Additionally, each field was annotated once under the AI-Only condition, yielding 36 more entries, for a total of 216 topic-field-annotator combinations.

We evaluated the LLM’s ability to replicate expert topic assignments using precision and recall, with the Human-Only condition treated as ground truth³. The AI-Only treatment achieved an average precision of 0.68 and recall of 0.55 when compared to Human-Only annotations, suggesting that while AI outputs are often accurate, they miss nearly half of expert-identified topics.

Annotators were significantly faster in the AI-Human-Deanchoring condition (89.91 words / min) than in the Human-Only (73.75 words/min) or AI-Human (71.15 words/min) conditions. This may reflect a tendency to anchor on LLM-generated suggestions, even when warned, leading to faster—but potentially *less critical*—annotation behavior.

Annotator agreement with AI-Only treatment was highest in the AI-Human condition (73.44%), followed by AI-Human-Deanchoring (71.41%), and lowest in Human-Only (54.64%). These findings suggest that LLM suggestions strongly influ-

³We consider the Human-Only annotations as the ground truth because, typically, experts work independently without AI-assistance. This makes the annotations the closest representation of real-life expert results in our study.

Source	Reported Cost in the paper	Standardized Cost (per 100 tokens)
Walther (2024)	\$0.001 per 100 input, \$0.003 per 100 output	\$0.004 roundtrip
DeepLearning.AI (2024)	\$4 per million tokens (GPT-4o); \$2 per million tokens (Batch API)	\$0.0002–\$0.0004
Chen et al. (2023)	\$0.20–\$300 per 10M tokens (GPT-J to GPT-4 Turbo)	\$0.000002–\$0.003
Irugalbandara et al. (2024)	5×–29× cost reduction over GPT-4	\$0.00014–\$0.0008
Samsi et al. (2023)	3–4 Joules per token (LLaMA-65B)	0.000083–0.000111 kWh
Husom et al. (2024)	0.000083–0.0023 kWh per query (2B–70B)	0.000083–0.0023 kWh
Calma (2023)	>10× increase in energy per query	Relative 10× increase (qualitative only)

Table 4: Reported and standardized LLM inference costs from recent sources. All values in the third column are standardized to cost per 100 tokens—monetary in USD and environmental in kilowatt-hours (kWh).

ence annotator decisions, and simple warnings are not sufficient to mitigate anchoring effects.

Pairwise agreement (Cohen’s κ ; Cohen, 1960) between annotators was highest when both had access to LLM suggestions (AI–Human: 0.71, AI–Human–Deanchoring: 0.69), and lowest in the Human–Only condition (0.57), reflecting a possible anchoring effect in which annotators align more closely—not with each other independently—but around the AI–provided suggestions.

5 RQ1: Estimating AI Inference Costs

Methodology To evaluate the marginal cost of using LLMs in our topic modeling workflow, we synthesized pricing and energy consumption data from peer-reviewed literature, arXiv preprints, and blog sources. For environmental costs, we reviewed the literature that estimates kilowatt-hour (kWh) usage and dollar-converted emissions per LLM inference. To enable comparison across studies with differing units and assumptions, we standardized all monetary costs to U.S. dollars per 100 tokens and converted energy-related figures to kilowatt-hours (kWh) per 100 tokens using a conversion factor of $1 \text{ kWh} = 3.6 \times 10^6 \text{ joules}$. While we do not report pretraining costs—since our study involves only inference—we present a plausible range of energy costs based on similar LLM use cases.

Results We synthesized recent estimates of both the monetary and environmental costs of LLM inference by reviewing peer-reviewed publications, technical reports, and industry analyses. Table 4 summarizes the most relevant findings.

Our analysis shows that LLM inference costs range from \$0.0002 to \$0.004 per 100-token roundtrip, depending on the model, pricing tier, and batching strategy (Walther, 2024; DeepLearning.AI, 2024; Chen et al., 2023). Models like GPT-4 Turbo average around \$0.004 per inference,

while batching can further reduce costs to as low as \$0.0002. Open-source alternatives offer additional savings, with some deployments reporting cost reductions of up to $29\times$ (Irugalbandara et al., 2024). Although not directly reporting numeric costs, theoretical analyses from Aryan et al. (2023) further support these findings by emphasizing significant potential for cost optimization through efficient deployment strategies.

Environmental costs also scale significantly with model size and usage. For example, generating 100 tokens with LLaMA-65B consumes approximately $8.3 \times 10^{-5} - 1.1 \times 10^{-4} \text{ kWh}$ (Samsi et al., 2023), while inference across commercial models ranging from 2B to 70B parameters consumes between 8.3×10^{-5} and $2.3 \times 10^{-3} \text{ kWh}$ per 100 tokens (Husom et al., 2024). Although these values may appear small in isolation, they accumulate rapidly at scale. As Calma (2023) note, the widespread integration of LLMs, such as their integration into search platforms, could increase the energy footprint per query by more than tenfold, underscoring the need for energy-efficient deployment strategies.

To contextualize these findings, we also consider the cost of human-led topic modeling, which is approximately \$48 per document per analyst (Carrell et al., 2016; Dernoncourt et al., 2017). Compared to this baseline, LLMs offer dramatic reductions in marginal financial cost per query. However, these monetary savings come with trade-offs: unlike human labor, LLM usage incurs measurable environmental impact that scales rapidly with deployment.

Moreover, since **our analysis draws from a diverse and evolving set of sources, both cost and energy estimates should be viewed as approximate benchmarks rather than fixed values**. These results underscore the importance of balancing cost-efficiency with sustainability when adopting AITM in educational research.

Setting	Coef. (s)	Std Err	z	p-value
Intercept (Human-Only)	383.7	94.8	4.1	<0.001
AI-Human	-1.6	132.8	-0.01	0.99
AI-Human- Deanchoring	-126.4	132.8	-0.95	0.34
AI-Only	-382.7	156.7	-2.4	0.015

Random Effects (Annotator): Variance = 849.67

Table 5: Linear mixed-effects model predicting annotation time (in seconds) across LLM support conditions with Human-Only as the reference category. The AI-Only condition significantly reduced annotation time, while partial AI support (AI-Human, AI-Human-Deanchoring) showed no statistically significant speed gains.

6 RQ2: Measuring Impact on Annotation Time

Methodology We used the data from stage 1 of the study to analyze annotation time.

For each human analyst, the total annotation time is calculated as:

$$\text{time}_a = \sum_{p=1}^{11} \text{time}_{ap} + 90 \text{ minutes}$$

Here, time_{ap} denotes the time spent by analyst a on Plan p , and the additional 90 minutes accounts for two structured group discussions—one 30-minute within-treatment session and one 60-minute cross-treatment session.

In total, we collected 77 person-by-document entries: 6 human analysts \times 11 Plans = 66 human entries, plus 11 entries from the AI-Only condition (1 AI \times 11 Plans). To estimate the impact of treatment on time-on-task, we fit a linear mixed-effects model:

$$\text{time}_{ap} = \text{Treatment}_{ap} + \phi_a + \varepsilon_{ap}$$

where, time_{ap} is the annotation time recorded by analyst a for Plan p , Treatment_{ap} is a fixed effect with four levels: Human-Only, AI-Only, AI-Human, and AI-Human-Deanchoring, with Human-Only as the reference category. ϕ_a is a random intercept for each analyst (6 humans + 1 AI), which accounts for analyst-specific baseline differences and increases the precision of estimates, helping us isolate the impact of the treatment more reliably. ε_{ap} is the residual error term.

Results Table 5 presents the results of this analysis. The baseline annotation time in the

Human-Only condition was approximately 384 seconds. The AI-Human condition showed virtually no difference in speed (Coef = -1.6 s, $p = 0.99$) relative to the Human-Only condition. The AI-Human-Deanchoring condition was faster by about 126 seconds relative to the Human-Only condition, but this difference was not statistically significant ($p = 0.341$). Notably, the AI-Only condition led to a statistically significant reduction of approximately 383 seconds ($p = 0.015$), representing a 6.4-minute decrease relative to the Human-Only condition. The random effect variance for annotators was estimated at 849.67, suggesting meaningful variability in baseline annotation speed between individuals. Some annotators were consistently faster or slower than others, regardless of treatment condition.

The **AI-Only** condition significantly reduces annotation time compared to **Human-Only**, suggesting that full AI support accelerates expert decision-making. However, **partial AI support** (i.e., AI-Human or AI-Human-Deanchoring) does not lead to statistically significant time savings. This indicates that the participants may have spent additional time reviewing and deliberating on the suggestions generated by the LLM. Rather than simply accepting AI outputs, Annotators have reportedly felt compelled to cross-check or validate these suggestions against their own judgment, leading to more careful and possibly slower decision-making. **This extra layer of comparison may have introduced hesitation or cognitive load, offsetting any potential efficiency gains from having AI support.** In contrast, participants in the Human-Only condition could rely solely on their intuition and expertise, resulting in a more streamlined workflow. This indicates that annotators may not gain measurable speed advantages unless they fully offload the task to the AI.

7 RQ3: Measuring Impact on Topic Identification

Methodology To evaluate how treatment condition influenced topic identification, we analyzed the Stage 2 annotation dataset described in Section 4. Each observation is a binary outcome indicating whether topic t was assigned to field f of plan p by annotator a . We fit the following multilevel linear probability model:

$$\Pr(\text{topic}_{fpat} = 1) = \text{treatment} + \eta_p + \varepsilon_{fpa}$$

Outcome	Human-Only Coef (SE) (reference)	AI-Only Coef (SE)	AI-Human Coef (SE)	AI-Human-Deanchoring Coef (SE)	Joint Test of Treat- ments (p-value)	Plan RE Variance (SE)
Academic Assessments	0.1979 (0.0715)	-0.0313 (0.0770)	0.0114 (0.0673)	0.0615 (0.0673)	0.6496	0.0344 (0.0171)
Academic Goals	0.3541 (0.0776)	-0.0763 (0.0906)	-0.0519 (0.0793)	-0.0105 (0.0793)	0.8054	0.0349 (0.0185)
Attendance	0.3098 (0.0877)	-0.1153 (0.0769)	-0.0360 (0.0674)	-0.0266 (0.0674)	0.5063	0.0654 (0.0297)
Behavioral Goals	0.1824 (0.0668)	-0.0435 (0.0717)	-0.0148 (0.0628)	0.0176 (0.0628)	0.8538	0.0301 (0.0150)
Classroom Management	0.0326 (0.0159)	-0.0326 (0.0242)	-0.0337 (0.0210)	-0.0140 (0.0210)	0.3577	0.0004 (0.0005)
College and Career Readiness	0.0505 (0.0394)	0.0051 (0.0408)	0.0078 (0.0357)	-0.0093 (0.0357)	0.9682	0.0110 (0.0054)
Curriculum	0.1067 (0.0556)	-0.0233 (0.0588)	-0.0090 (0.0515)	0.0390 (0.0515)	0.7016	0.0213 (0.0105)
Graduation	0.0167 (0.0159)	-0.0167 (0.0243)	-0.0009 (0.0212)	0.0009 (0.0212)	0.8886	0.0004 (0.0005)
Instruction	0.0674 (0.0335)	-0.0674 (0.0439)	-0.0246 (0.0383)	0.0056 (0.0383)	0.3472	0.0047 (0.0029)
Parent/Community Engagement	0.1982 (0.0699)	-0.0871 (0.0669)	-0.0418 (0.0585)	-0.0193 (0.0585)	0.6048	0.0383 (0.0179)
Professional Development	0.2266 (0.0786)	-0.0877 (0.0732)	-0.0348 (0.0641)	0.0550 (0.0641)	0.2369	0.0497 (0.0231)
Technology Use Integration	0.0756 (0.0636)	-0.0478 (0.0257)	-0.0223 (0.0226)	0.0122 (0.0226)	0.0935	0.0455 (0.0189)
Classroom Environment or Culture	0.1059 (0.0463)	-0.0503 (0.0510)	-0.0185 (0.0446)	-0.0491 (0.0446)	0.6521	0.0139 (0.0070)

Table 6: Coefficients (with SEs) from multilevel linear probability models estimating the impact of treatment on topic identification, relative to the Human-Only baseline. Joint tests assess whether all AI-based treatments collectively differ from the baseline. No statistically significant differences were observed across any treatment, indicating that topic identification remained stable despite varying levels of AI assistance.

Here, topic_{fpat} is 1 if topic t was identified by analyst a in field f of plan p , and 0 otherwise. The model includes treatment as a fixed effect (with Human-Only as the reference condition) and η_p as a random intercept for each plan. This structure captures the hierarchical nature of the data while accounting for differences in topic prevalence across plans. ε_{fpa} accounts for the residual error.

We tested several alternative model specifications, including crossed and nested analyst effects, but these did not improve model fit or alter the results meaningfully. Thus, we retained the simpler formulation, which allows us to isolate the effect of treatment condition on topic identification behavior across annotators.

Results The results of the regression is given in Table 6. We used Human-Only as the reference condition and computed coefficients for each AI-based treatment: AI-Only, AI-Human, and AI-Human-Deanchoring. Each row in Table 6 presents the estimated probability of a topic being identified under each treatment, along with standard errors and joint significance test results.

For the topic *Academic Assessments*, the baseline Human-Only coefficient is 0.1979. Compared to this, the AI-Only coefficient is about 3 percentage points lower, the AI-Human coefficient is 1.1 percentage points higher, and the AI-Human-Deanchoring coefficient is 6.2 percentage points higher, respectively.

When comparing the Human-Only and AI-Human conditions reveals minimal differences across topics, with coefficients typically within ± 5 percentage points and no statistically significant deviations. This suggests that introducing AI support does not substantially shift topic identification patterns, and expert judgments remain largely consistent with the Human-Only baseline.

Next, examining the AI-Only and AI-Human conditions relative to the Human-Only baseline, we find that human analysts working with AI suggestions tend not to diverge far from the original AI-Only outputs. Instead, the AI-Human estimates tend to fall between the AI-Only and Human-Only values, implying that humans may be partially influenced— or anchored— by AI suggestions in their decision-making.

A similar pattern holds when comparing AI-Human and AI-Human-Deanchoring, each relative to the Human-Only baseline. Despite the presence of explicit deanchoring warnings, the estimates in these two conditions show minimal deviation from each other when considered through their differences from the baseline. In some cases, the deanchoring estimates are numerically closer but not statistically different to the AI-Human ones than to the Human-Only baseline. This indicates that, in this context, explicit instructions to critically evaluate AI suggestions had limited observable effect.

However, the joint significance test ($p = 0.6496$) does not indicate statistically significant differences between the treatment groups. This pattern holds across most topics. Joint significance tests across all 13 outcomes yielded p -values greater than 0.05, suggesting that the combination of effects from the three AI-based treatments does not reflect a systematic deviation from the Human-Only condition. In other words, there was no consistent pattern across the three AI conditions that significantly distinguished them from the Human-Only baseline.

The findings suggest that **while human annotators may incorporate AI input into their judgments, they are not significantly over-relying on it compared to the Human-Only condition.** Deanchoring prompts offered limited additional benefit in mitigating potential anchoring effects. **Topic identification remained stable across all treatment conditions, indicating that different approaches to incorporating AI did not produce meaningful divergence in these results.**

8 Conclusion

This study examined how AI-enabled topic modeling (AITM) can be integrated into educational research workflows, focusing on its financial, environmental, cognitive, and analytical trade-offs. Our findings show that while LLMs provide clear efficiency benefits, especially by speeding up annotation and lowering costs, these gains come with important risks. In both stages of human-in-the-loop annotation, we found evidence of anchoring bias: human analysts who saw LLM suggestions were more likely to stick with them, even when explicitly cautioned. However, when we looked at topic-level outcomes, we did not find statistically significant differences in which topics were identified across the treatment conditions. This suggests that while anchoring may shape how an-

notators approach the task, for example, in how quickly they work or how much they agree with AI, it doesn't necessarily change the final set of topics they choose.

As institutions consider scaling up AI-based analysis, the trade-off between speed and depth becomes harder to ignore. AI can definitely help efficiency and cost reduction, but human judgment is still crucial, especially for subtle, context-specific details that models tend to miss. Relying only on AI might make things more efficient, but it also risks losing the kinds of insights that matter most for real-world decisions. A balanced approach, where AI helps with the heavy lifting, but humans stay in the loop, seems like the best way to get both speed and substance.

Limitations

While this study offers important insights into the use of LLMs for topic modeling in educational research, it is essential to acknowledge its limitations. First, our analysis is based on a relatively small sample of 23 school improvement plans from a single state, which may limit the generalizability of our findings to other contexts. Second, our study focused on a specific type of text document. While these documents are relevant to educational leadership and policy, the findings may not be directly transferable to other forms of educational text, such as student essays, teacher evaluations, or policy documents. Third, our investigation of anchoring bias relied on a single de-anchoring intervention. While this allowed us to isolate the effect of such prompts, future research could explore the efficacy of other de-biasing techniques, such as structured protocols or collaborative decision-making strategies. Finally, the rapidly evolving nature of LLM pricing and energy consumption means that these figures of our cost analysis should be interpreted as indicative rather than definitive.

Acknowledgments

We are grateful to our study participants and anonymous reviewers for their valuable feedback, and also thank NEE, DESE for sharing their data and Dr. Thomas Hairston for his helpful input. This work was supported in part by a grant from the Minerva Research Initiative of the Department of Defense (Award No: FA9550-22-1-0171) and by the National Science Foundation under award IIS-2327143.

References

- Abdulrahman M Al-Zahrani. 2024. Unveiling the shadows: Beyond the hype of ai in education. *Heliyon*, 10(9).
- Abi Aryan, Aakash Kumar Nain, Andrew McMahon, Lucas Augusto Meyer, and Harpreet Singh Sahota. 2023. The costly dilemma: generalization, evaluation and cost-optimal deployment of large language models. *arXiv preprint arXiv:2308.08061*.
- Claus Boye Asmussen and Charles Møller. 2019. Smart literature review: a practical topic modelling approach to exploratory literature review. *Journal of Big Data*, 6(1):1–18.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. [Latent dirichlet allocation](#). *J. Mach. Learn. Res.*, 3:993–1022.
- Gavin Brookes and Tony McEnery. 2019. The utility of topic modelling for discourse studies: A critical evaluation. *Discourse Studies*, 21(1):3–21.
- Justine Calma. 2023. [Ai is an energy hog, but deepseek could change that](#). Accessed: 2025-04-11.
- Alexia Cambon, Brent Hecht, Benjamin Edelman, Donald Ngwe, Sonia Jaffe, Amy Heger, Mihaela Vorvoreanu, Sida Peng, Jake Hofman, Alex Farach, Margarita Bermejo-Cano, Eric Knudsen, James Bono, Hardik Sanghavi, Sofia Spatharioti, David Rothschild, Daniel G. Goldstein, Eirini Kalliamvakou, Peter Cihon, and 3 others. 2023. [Early llm-based tools for enterprise information workers likely provide meaningful boosts to productivity](#). Technical Report MSR-TR-2023-43, Microsoft.
- David S Carrell, David J Cronkite, Bradley A Malin, John S Aberdeen, and Lynette Hirschman. 2016. Is the juice worth the squeeze? costs and benefits of multiple human annotators for clinical text de-identification. *Methods of information in medicine*, 55(04):356–364.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2023. [Fragalgpt: How to use large language models while reducing cost and improving performance](#). *CoRR*, abs/2305.05176.
- Alexander Choi, Syeda Sabrina Akter, J.P. Singh, and Antonios Anastasopoulos. 2024. [The LLM effect: Are humans truly using LLMs, or are they being influenced by them instead?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22032–22054, Miami, Florida, USA. Association for Computational Linguistics.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- Stijn Daenekindt and Jeroen Huisman. 2020. Mapping the scattered field of research on higher education. a correlated topic model of 17,000 articles, 1991–2018. *Higher Education*, 80(3):571–587.
- DeepLearning.AI. 2024. [Falling llm token prices and what they mean for ai companies](#). Accessed: 2025-04-11.
- Fabrizio Dell’Acqua, Edward McFowland III, Ethan R Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Kraye, François Candelon, and Karim R Lakhani. 2023. Navigating the jagged technological frontier: Field experimental evidence of the effects of ai on knowledge worker productivity and quality. *Harvard Business School Technology & Operations Mgt. Unit Working Paper*, (24-013).
- Franck Dernoncourt, Ji Young Lee, Özlem Uzuner, and Peter Szolovits. 2017. [De-identification of patient notes with recurrent neural networks](#). *J. Am. Medical Informatics Assoc.*, 24(3):596–606.
- AMY Drahota, Rosemary D Meza, Brigitte Brikho, Meghan Naaf, Jasper A Estabillo, Emily D Gomez, Sarah F Vejnaska, Sarah Dufek, Aubyn C Stahmer, and Gregory A Aarons. 2016. Community-academic partnerships: A systematic review of the state of the literature and recommendations for future research. *The Milbank Quarterly*, 94(1):163–214.
- Jessica Maria Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. 2024. [Cognitive bias in decision-making with LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12640–12653, Miami, Florida, USA. Association for Computational Linguistics.
- Jie Gao, Yuchen Guo, Gionnieve Lim, Tianqin Zhang, Zheng Zhang, Toby Jia-Jun Li, and Simon Tangi Perreault. 2024a. [Collabcoder: A lower-barrier, rigorous workflow for inductive collaborative qualitative analysis with large language models](#). In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI ’24, New York, NY, USA. Association for Computing Machinery.
- Sally Gao, Milda Norkute, and Abhinav Agrawal. 2024b. [Evaluating interactive topic models in applied settings](#). In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI EA ’24, New York, NY, USA. Association for Computing Machinery.
- Kate Goddard, Abdul Roudsari, and Jeremy C Wyatt. 2012. Automation bias: a systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association*, 19(1):121–127.
- Maarten Grootendorst. 2022. [Bertopic: Neural topic modeling with a class-based TF-IDF procedure](#). *CoRR*, abs/2203.05794.
- Igor Grossmann, Matthew Feinberg, Dawn C Parker, Nicholas A Christakis, Philip E Tetlock, and William A Cunningham. 2023. Ai and the transformation of social science research. *Science*, 380(6650):1108–1109.

- Daniel Hershcovich, Nicolas Webersinke, Mathias Kraus, Julia Bingler, and Markus Leippold. 2022. [Towards climate awareness in NLP research](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2480–2494, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Alexis Horst, Brian D. Schwartz, Jenifer A. Fisher, Nicole Michels, and Lon J. Van Winkle. 2019. [Selecting and performing service-learning in a team-based learning format fosters dissonance, reflective capacity, self-examination, bias mitigation, and compassionate behavior in prospective medical students](#). *International Journal of Environmental Research and Public Health*, 16(20).
- Erik Johannes Husom, Arda Goknil, Lwin Khin Shar, and Sagar Sen. 2024. The price of prompting: Profiling energy use in large language models inference. *arXiv preprint arXiv:2407.16893*.
- 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 llms in production](#). In *2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 280–291.
- Sonia Jaffe, Neha Parikh Shah, Jenna Butler, Alex Farach, Alexia Cambon, Brent Hecht, Michael Schwarz, and Jaime Teevan. 2024. [Generative ai in real-world workplaces](#). Technical Report MSR-TR-2024-29, Microsoft.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, 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.
- Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2024. [Benchmarking cognitive biases in large language models as evaluators](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 517–545, Bangkok, Thailand. Association for Computational Linguistics.
- Yan Liu and He Wang. 2024. [Who on earth is using generative ai?](#) Policy Research Working Paper 10870, World Bank. License: CC BY 3.0 IGO.
- Larry Michaelsen, Arletta Knight, and L. Fink. 2002. *Team-based learning: a transformative use of small groups*.
- Brent Daniel Mittelstadt, Patrick Allo, Mariarosaria Taddeo, Sandra Wachter, and Luciano Floridi. 2016. The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2):2053951716679679.
- Douglas C Montgomery. 2017. *Design and analysis of experiments*. John Wiley & sons.
- Rosanna Nagtegaal, Lars Tummers, Mirko Noordegraaf, and Victor Bekkers. 2020. Designing to debias: Measuring and reducing public managers’ anchoring bias. *Public Administration Review*, 80(4):565–576.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Cassandra Overney, Belén Saldías, Dimitra Dimitrakopoulou, and Deb Roy. 2024. [Sensemate: An accessible and beginner-friendly human-ai platform for qualitative data analysis](#). In *Proceedings of the 29th International Conference on Intelligent User Interfaces, IUI ’24*, page 922–939, New York, NY, USA. Association for Computing Machinery.
- Chau Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. 2024. [TopicGPT: A prompt-based topic modeling framework](#). 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 2956–2984, Mexico City, Mexico. Association for Computational Linguistics.
- Anne C. Gill Rachael A. Hernandez, Paul Haidet and Cayla R. Teal. 2013. [Fostering students’ reflection about bias in healthcare: Cognitive dissonance and the role of personal and normative standards](#). *Medical Teacher*, 35(4):e1082–e1089. PMID: 23102159.
- Philip Resnik. 2024. [Large language models are biased because they are large language models](#). *Preprint*, arXiv:2406.13138.
- Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. [A comprehensive survey of hallucination in large language, image, video and audio foundation models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11709–11724, Miami, Florida, USA. Association for Computational Linguistics.
- Malik Sallam. 2023. Chatgpt utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, volume 11, page 887. MDPI.
- Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadeppally. 2023. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9. IEEE.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Min Sun, Jing Liu, Junmeng Zhu, and Zachary LeClair. 2019. Using a text-as-data approach to understand reform processes: A deep exploration of school improvement strategies. *Educational evaluation and policy analysis*, 41(4):510–536.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020. [Label Studio: Data labeling software](#). Open source software available from <https://github.com/heartexlabs/label-studio>.

Stephen Walther. 2024. [How much does it cost to call openai apis?](#) Accessed: 2025-04-11.

Yinying Wang, Alex J Bowers, and David J Fikis. 2017. Automated text data mining analysis of five decades of educational leadership research literature: Probabilistic topic modeling of eaq articles from 1965 to 2014. *Educational administration quarterly*, 53(2):289–323.

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, Mexico City, Mexico. Association for Computational Linguistics.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. 2024. [Fact-and-reflection \(FaR\) improves confidence calibration of large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8702–8718, Bangkok, Thailand. Association for Computational Linguistics.

A AI-Human-Deanchoring Warning

To address the anchoring bias minimally, we introduced a new treatment AI-Human-Deanchoring. Similar to the AI-Human setting, the AI-Human-Deanchoring group also received the results generated by AI (i.e., AI-Only list) paired with the following prominently displayed instructions:

This document has annotations suggested by the LLMs. It will be your task to decide whether these annotations are correct or not. Delete or modify annotations as you see fit. However, we have found evidence of anchoring bias when annotators receive LLM suggestions. Anchoring bias is a cognitive bias where an individual relies too heavily on an initial piece of information (the “anchor”) when making decisions. This means that the initial suggestions provided by the LLM might disproportionately influence the final labels you create, potentially reducing the diversity and originality of the Final Topic List. It is important for you to be aware of this bias and make conscious efforts to critically evaluate and adjust your topics and suggestions to ensure the annotations are accurate and unbiased. We ask you to be extra critical while annotating these documents.

Our intention was to observe how experts react to the awareness of anchoring bias from LLM suggestions and whether they adjust their behavior accordingly. We also aimed to evaluate if merely knowing about the bias was effective enough to help annotators de-anchor.

B Stage 1: All topic lists

B.1 AI-Only topic list

Topic Name	Topic Definition
Education	This topic encompasses various aspects of the educational process, including instructional strategies, curriculum development, assessment methods, professional development for educators, student performance tracking, and educational objectives and goals alignment with standards.
Student Assessment and Achievement	This topic covers the processes and methodologies involved in evaluating student performance, including standardized testing, reading assessments, and other forms of academic evaluation. It also includes strategies for improving student achievement levels in core subjects like math, ELA, and science.
Professional Development	This topic involves the continuous education and skill development of teachers and educational staff, including the implementation of best teaching practices, collaboration among educators, and the use of technology and data to enhance teaching effectiveness.
Curriculum and Instruction	This topic focuses on the design, implementation, and evaluation of educational curricula and instructional materials. It includes the alignment of curriculum with educational standards, the development of instructional strategies to meet diverse learning needs, and the integration of technology into the learning environment.
School Improvement Planning	This topic covers the strategic planning processes schools undertake to improve academic performance and operational efficiency. It includes setting and aligning goals with educational standards, data-driven decision-making, and the implementation of interventions and supports to meet educational objectives.
Behavioral Interventions and Supports	This topic addresses strategies and programs designed to improve student behavior and create positive school environments. It includes the implementation of Positive Behavior Interventions and Supports (PBIS), discipline management strategies, and efforts to increase student engagement and accountability.
Parent and Community Engagement	This topic involves strategies and practices for involving parents and the community in the educational process. It includes parent-teacher communication, community partnerships to support student achievement, and stakeholder involvement in school decision-making processes.
Educational Technology	This topic covers the use of technology in educational settings, including the implementation of digital tools and resources to support teaching and learning, the use of assessment technologies, and the training of educators in effective technology integration.

Table 7: AI-Only topic list for stage 1. We generated the list using GPT-4o-mini model using chatGPT API.

B.2 Final topic list after stage 1

Topic Name	Topic Definition
Academic Assessments	This topic includes mandated annual state assessments like MAP and other district and school level assessments to evaluate academic progress.
Academic Goals	This topic covers the strategic planning processes schools undertake in aligning goals with educational standards and the implementation of interventions and supports to meet educational objectives in core subjects like math, ELA, and science.
Behavioral Goals	This topic addresses strategies and programs designed to improve student behavior and create positive school environments. It includes the implementation of Positive Behavior Interventions and Supports (PBIS), discipline management strategies, and efforts to increase student engagement and accountability.
Classroom Management	This topic covers how teachers develop and implement procedures to maximize instructional time/space/transitions/activities for efficiency in the classroom.
Classroom Environment/Culture	This topic covers how all members of the school community (administrators, teachers, and students) develop and implement pro-social behaviors inside and outside of academic instruction. This can include social-emotional learning (SEL) and fostering of pro-social attitudes and behaviors.
Curriculum	This topic covers what teachers do to plan, design, and develop materials to promote learning. This can include collaboration through professional learning communities (PLCs) as long as it is specifically around curriculum design.
Instruction	This topic covers what teachers do to deliver instruction during active academic time with students in the classroom. This includes instructional strategies and also collaboration in professional learning communities (PLCs) as long as it is specifically about how teachers engage with students in academics, instructional strategies, academic press, critical thinking, or formative assessment.
Professional Development	This topic involves the continuous education and skill development of teachers and educational staff, including evaluation of teachers, classroom observation, and collaboration around improving what teachers do to work with students.
Parent/Community Engagement	This topic involves strategies and practices for involving parents and the community (including school boards) in the educational process. It includes parent-teacher communication, community partnerships to support student achievement, and stakeholder involvement in school decision-making processes.
Technology Use/Integration	This topic covers the use and integration of technological tools, resources, and materials.
College and Career Readiness (CCR)	This topic covers college and career readiness (CCR) of students including Career & Technical Education credit hours and employment, military, and college placement.
Graduation	This topic involves the matriculation between grades and completed secondary state requirements. This is often expressed in the graduation rates of students.
Attendance	This topic involves the attendance rates and percents of students.

Table 8: Stage 1 Final Topic List curated by the participants.

B.3 All Group-Specific Topic List

Human-Only List	AI-Human List	AI-Human-Deanchoring List	Final Topic List	AI-Only List
State Assessment	School Assessment and Achievement	Student Assessment and Achievement	Academic Assessments	Student Assessment and Achievement
Localized Assessment			Academic Goals	
	Data-Driven Decisionmaking			
Behavioral Goals/ Classroom Management	Behavioral Interventions and Support	Behavioral Interventions and Supports	Behavioral Goals	Behavioral Interventions and Supports
Student Support	Data-Driven Decisionmaking			
		Classroom Management	Classroom Management	
Student/ Teacher Relationships		Classroom Culture/ Environment	Classroom Environment/ Culture	
Localized Curriculum	Curriculum and Instruction	Curriculum	Curriculum	Curriculum and Instruction
	Collaboration			
Teaching Strategies	Curriculum and Instruction	Instruction	Instruction	
Teacher Evaluation Components	Collaboration			
Professional Development	Professional Development	Professional Development	Professional Development	Professional Development
Instructional Coach				
Stakeholder Engagement	Parent and Community Engagement	Parent and Community Engagement	Parent/Community Engagement	Parent and Community Engagement
Technology Integration		Educational Technology	Technology Use/Integration	Education Technology
College, Career, Readiness			College and Career Readiness (CCR)	
Graduation/ Matriculation Rate			Graduation	
Attendance			Attendance	
	District Alignment	Education		Education
		School Improvement Planning		School Improvement Planning

Table 9: Comparison of topic lists generated across conditions in Stage 1. Entries are grouped to show thematic overlap and consolidation across all lists. Struckthrough entries indicate topics that annotators collectively decided to discard during the final discussion phase.

C Stage-2 Detailed Results:

We provide computation details for the metrics reported in Table 3. For the analysis, each paragraph-level field was encoded as a 14-dimensional binary vector: 13 dimensions correspond to the presence or absence of each topic from the Final Topic List, and the final slot indicates a “None” label (no topic assigned). These vectors were used for computing precision, recall, and agreement metrics.

Annotators	precision	recall
A1	0.57	0.5
A2	0.71	0.67
A3	0.77	0.47
A4	0.70	0.44
A5	0.65	0.65
Avg	0.68	0.55

Table 10: For each annotator in Stage 2, the precision and recall percentages of the AI-Only annotations over these documents when measured against the annotations of experts acting under the Human-Only condition. Also, the averages of these LLM precision and recall percentages.

Average Precision and Recall To evaluate how closely LLM-generated annotations align with human judgment, we compute precision and recall by comparing the LLM-assigned topics to those assigned by human annotators under each treatment condition (Human-Only, AI-Human, and AI-Human-Deanchoring).

Using the Human-Only condition as ground truth, we found that the LLM achieved an average precision of 0.68 and a recall of 0.55. This means that while 68% of LLM predictions aligned with expert judgments, nearly half of the expert-identified topics were not captured by the model. Thus, the LLM shows reasonable accuracy, but limited coverage in replicating full expert insight.

	Human-Only	AI-Human	AI-Human-Deanchoring
Average Annotation Speed (words/min)	73.75	71.15	89.91
Average Annotator Agreement with AI (%)	54.64	73.44	71.41

Table 11: Comparison of average annotation speed (words per minute) and average Human-AI agreement across the three conditions.

Average Annotation Speed To understand how LLM support affects efficiency, we calculated annotation speed in words per minute (wpm). For each document field, we divided the number of words by the time each annotator took to complete it, then averaged these speeds by condition. As shown in Table 11, annotators in the Human-Only condition averaged 73.75 wpm. This dipped slightly in the AI-Human condition to 71.15 wpm, but surprisingly jumped to 89.91 wpm in the AI-Human-Deanchoring condition—even though those annotators were explicitly warned about bias. The results suggest that having AI suggestions, even with cautionary prompts, may encourage annotators to move faster—possibly by relying on the AI’s suggestions rather than thinking through every decision from scratch.

Average Annotator Agreement with AI To assess how closely human annotators aligned with LLM-generated suggestions, we calculated the percentage of topic assignments that matched the AI-Only output. For each annotator–field pair, we compared the human-assigned topics to the AI’s and computed the overlap. These agreement scores were then averaged within each condition (see Table 11).

Agreement varied by condition. In the Human-Only setting—where annotators had no AI support—the average agreement with the AI was 54.64%. This jumped to 73.44% in the AI-Human condition, suggesting that access to AI suggestions substantially influenced annotator decisions. In the AI-Human-Deanchoring condition, agreement remained similarly high at 71.41%, even though annotators were explicitly warned about potential bias. This suggests that simply cautioning annotators may not be enough to counter the influence of LLM outputs.

Inter-Annotator Agreement. To assess how consistently annotators applied the topic labels, we used Cohen’s κ (Cohen, 1960), a standard measure for inter-rater agreement on categorical decisions. Because each document field was annotated by a pair of analysts within the same condition (see Table??), we were able to compute pairwise κ scores for each condition and then average them.

The results (Table 12) show that annotators aligned more closely when LLM suggestions were available. Agreement was highest in the AI-Human condition ($\kappa = 0.71$) and nearly as high in the AI-Human-Deanchoring setting ($\kappa = 0.69$). In

Agreement between	Human-Only	AI-Human	AI-Human-Deanchoring	Avg per Annotator
A1 and A4	0.48	0.72	0.79	0.66
A2 and A5	0.65	0.69	0.59	0.64
Avg per Condition	0.57	0.71	0.69	

Table 12: Agreement between annotator pairs across different treatment conditions. We report annotator agreement Cohen’s κ for each pair per setting. The average agreement per annotator pair is higher for the settings with LLM suggestions, implying towards a potential anchoring effect.

contrast, agreement dropped in the Human-Only condition ($\kappa = 0.57$), where annotators worked independently. These findings suggest that LLM support—regardless of deanchoring prompts—tends to guide annotators toward similar decisions, potentially reflecting a convergence effect around AI-generated suggestions.

Advances in Auto-Grading with Large Language Models: A Cross-Disciplinary Survey

Fredrick Eneye Tania-Amanda Nkoyo¹, Chukwuebuka Fortunate Ijezue¹, Maaz Amjad¹, Ahmad Imam Amjad², Sabur Butt³, Gerardo Castañeda-Garza³

¹Texas Tech University, Texas, USA

²The University of Punjab, Pakistan

³Tecnológico de Monterrey, Mexico

tafredri@ttu.edu, cijezue@ttu.edu, maaz.amjad@ttu.edu,
Ahmadimamjad@gmail.com, saburb@tec.mx, g.castaneda@tec.mx

Abstract

With the rise and widespread adoption of Large Language Models (LLMs) in recent years, extensive research has been conducted on their applications across various domains. One such domain is education, where a key area of interest for researchers is investigating the implementation and reliability of LLMs in grading student responses. This review paper examines studies on the use of LLMs in grading across six academic sub-fields: educational assessment, essay grading, natural sciences and technology, social sciences and humanities, computer science and engineering, and mathematics. It explores how different LLMs are applied in automated grading, the prompting techniques employed, the effectiveness of LLM-based grading for both structured and open-ended responses, and the patterns observed in grading performance. Additionally, this paper discusses the challenges associated with LLM-based grading systems, such as inconsistencies and the need for human oversight. By synthesizing existing research, this paper provides insights into the current capabilities of LLMs in academic assessment and serves as a foundation for future exploration in this area.

1 Introduction

Grading has traditionally been a manual process conducted by human teachers or graders, which can be time-intensive, laborious, and subject to inconsistencies due to individual judgment (Gnanaprakasam and Lourdusamy, 2024). To circumvent some of these issues, standardized examinations and rubrics are designed. Nonetheless, these may fail to detect variations in student ability or in learning styles (Gnanaprakasam and Lourdusamy, 2024). Furthermore, traditional grading methods fail to deliver tailored feedback at scale, further decreasing the value of exams as opportunities for personalized assessment (Haque et al., 2022).

Manual grading has significant mental and physical implications both for educators and students (Skaalvik and Skaalvik, 2017) and students (Hough, 2023). Due to its repetitive and time-consuming nature, it leads to physical and mental fatigue for educators. Previous research indicates that the stress associated with manual grading can also hinder educators' ability to focus on other critical aspects of teaching, such as lesson planning and student engagement (Hakanen et al., 2006). For students, the subjective nature of manual grading can introduce biases, which may negatively impact students' academic outcomes and their trust in the evaluation process (Wigfall, 2020). Delayed feedback from manual grading can leave students in prolonged uncertainty, which may increase their anxiety levels (England et al., 2019).

On the contrary, the rapid advancements in the field of artificial intelligence (AI), and the introduction of LLMs, capable of understanding and generating human-like text, have shifted this paradigm. LLM-based grading is any grading technique that leverages powerful Large Language Models to automate the evaluation of student responses, offering potential benefits in speed, consistency, and scalability. This shift is particularly relevant in educational settings where large volumes of assessments, such as essays and short answers, need efficient processing. Research conducted by Grandel et al. (2024) showed the ability of LLM-based grading techniques to reduce grading time by 81.2%. AI-automated grading could reduce the workload on educators, allowing them to spend more time teaching. Such systems could ensure consistency and objectivity in evaluations, reducing human biases and providing fair assessments for all students.

While individual studies have demonstrated LLM applications in specific educational domains, a comprehensive cross-disciplinary analysis is essential to understand broader patterns, identify transferable methodologies, and reveal domain-

specific challenges. Educational assessment varies significantly across disciplines—from objective STEM problem-solving to subjective humanities analysis—making it crucial to examine how LLMs perform across this spectrum. A cross-disciplinary perspective enables identification of universal best practices, domain-specific adaptations, and systematic gaps that single-domain studies cannot reveal. Moreover, such an approach allows for the synthesis of methodological insights that can inform both researchers and practitioners across diverse educational contexts.

This review paper surveys the current landscape of LLM-based assessment across six academic domains—educational assessment, essay grading, natural sciences and technology, social sciences and humanities, computer science and engineering, and mathematics. It synthesizes findings from 30 recent studies, analyzing how LLMs are applied in different assessment formats, the prompting strategies used, their alignment with human evaluators, and the contextual variables influencing performance. In doing so, this paper provides a cross-disciplinary framework for understanding the capabilities and limitations of LLM-based grading systems. It also highlights methodological trends, emerging implementation strategies, and the evolving role of human-AI collaboration in educational assessment. Overall, this paper provides a timely cross-disciplinary survey that will serve as a useful reference. It is well-scoped and captures key themes in LLM-based grading. Moreover, it brings to light current challenges and limitations in the area, such as rubric drift and LLM transparency issues.

2 Data Collection

We conducted a systematic literature review using the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology to identify relevant studies on the use of large language models (LLMs) in educational assessment. The search aimed to include published academic research between January 1, 2022, through January 14, 2025. The selection focused on works that addressed LLM use in grading, feedback generation, essay evaluation, short-answer marking, domain-specific assessments, and pedagogical implications.

Our review focuses on six academic domains—educational assessment, essay grading, natural sciences and technology, social sciences and

humanities, computer science and engineering, and mathematics—selected to represent the breadth of educational assessment contexts where LLMs are being applied. These domains were chosen to span the spectrum from highly structured (mathematics, computer science) to open-ended assessments (humanities), include both technical and non-technical fields, and represent different cognitive complexity levels as defined by Bloom’s taxonomy. This selection enables a comprehensive analysis of how LLM performance varies across assessment types, content domains, and evaluation criteria while maintaining sufficient depth within each domain.

Articles were gathered from multiple scholarly databases and repositories as detailed in Table 1, including Google Scholar, arXiv, IEEE Xplore, ACL Anthology, and ERIC (Education Resources Information Center). We also examined proceedings from key conferences, including ACL, EMNLP, EDM (Educational Data Mining), LAK (Learning Analytics and Knowledge), and AIED (Artificial Intelligence in Education). Keywords used in the searches included combinations of: “large language models,” “educational assessment,” “automated grading,” “essay scoring,” “student feedback,” “ChatGPT,” “GPT-4,” “short answer evaluation,” and “AI in education.”

Studies were selected based on predefined inclusion criteria: (1) empirical studies involving LLM applications in educational assessment, (2) published between 2022-2025, (3) sufficient detail on methodology and results, and (4) focus on grading, feedback, or evaluation tasks. The PRISMA flow diagram detailing the study selection process is presented in Figure 1.

The initial search yielded 104 articles and reports. After removing duplicates, irrelevant papers (e.g., not focused on education or assessment, theoretical works without application), we filtered 48 full-text articles assessed for eligibility. We excluded the remaining articles for insufficient empirical content, lack of focus on assessment, or being out of scope (e.g., general education technology without AI involvement). This rigorous selection process yielded 30 articles for the final review. The included studies comprised 19 peer-reviewed publications and 11 preprints, reflecting the rapidly evolving nature of this research area.

3 Variables of Study

To analyze LLM applications in educational assessment, we define a set of key variables that form the basis for comparing diverse implementations. These variables are grouped into four main categories. First, the assessment types include Multiple-Choice Questions (MCQs), Short-Answer Questions, Essay Assessments, Programming Assignments, Mathematics Assessments, and Handwritten Assessments. Second, the studies span a broad range of education levels—from Early and Primary education through Secondary, Undergraduate, Graduate, to Professional Education. Third, human annotators are classified into distinct groups: Expert Evaluators, Experienced Educators, Novice Evaluators, Field Practitioners, and Unspecified Graders, a categorization that is crucial for understanding how LLM outputs compare with human judgment. Finally, evaluation metrics employed across the studies include Cohen’s Kappa, Quadratic Weighted Kappa, Krippendorff’s Alpha, Pearson and Spearman Correlations, Accuracy, F1 Score, and Win Rate. Collectively, this framework is essential for identifying patterns and making meaningful cross-disciplinary comparisons of LLM-assisted assessment. Detailed tables for each variable category can be found in Tables 2, 3, 4, and 5 in the Appendix.

4 LLMs in Assessment

Large language models face significant challenges in educational assessment contexts, particularly when evaluating higher-order cognitive tasks and providing nuanced feedback comparable to human experts (Kasneji et al., 2023; Gnanaprakasam and Lourdusamy, 2024). However, researchers have developed innovative approaches to address these limitations, demonstrating increasingly promising results across diverse educational settings.

Early evaluations by Teckwani et al. (2024) in the physiological education domain revealed that LLMs, such as GPT-3.5, GPT-4o, and Gemini, achieved only moderate alignment with human graders (for example, Gemini reached 71% agreement with $r = 0.672$), whereas experienced faculty demonstrated superior consistency (80% agreement, $r = 0.936$). This divergence was especially pronounced on higher-order cognitive tasks, which has driven further research into methods for enhancing LLM assessment performance. To address these challenges, several studies have focused on

structured rubrics and frameworks (see Appendix A.4.2). For example, Morjaria et al. (2024) found that when ChatGPT-4 was paired with question-specific rubrics, score inflation was reduced and the correlation with human reviewers improved to between $r = 0.6$ and 0.7 . In parallel, Yuan and Hu (2024) observed that Llama-UKP models, when provided with well-defined assessment criteria (see Table 2), achieved high agreement with human evaluators (Spearman $\rho = 0.843$). These results underscore that explicit, rubric-based guidance consistently leads to more interpretable and reliable feedback in diverse educational contexts.

In addition to structured frameworks, advanced prompting strategies have emerged as critical tools for optimization (see Appendix A.1). The Reason-Act-Evaluate" (RAE) prompt introduced by Li et al. (2024) structures the assessment process into three clearly defined stages: reasoning about criteria, performing an assessment, and reviewing the outcome (see Appendix A.4.3). When applied to 1,235 student-generated texts, the RAE method achieved 76.5% accuracy and demonstrated strong alignment in dimensions such as logical reasoning ($\rho = 0.824$). This approach not only mirrors human grading practices but also significantly boosts the overall reliability of LLM outputs without necessarily relying on cutting-edge architectures.

Furthermore, the most promising results have been observed when LLMs are integrated into hybrid human-AI systems. Tools such as EvalGen, developed by Shankar et al. (2024b), combine LLM-generated assessments with human oversight to mitigate challenges like criteria drift (refer to Appendix A.4). Similar hybrid approaches proposed by Sinha et al. (2023), Khan et al. (2023), and the “Assisted RAE” method by Li et al. (2024) reinforce the idea that human-AI collaboration can enhance assessment consistency and integrity while reducing individual grader workload.

Finally, comparative model insights reveal that while newer LLMs often outperform older ones, the overall effectiveness of an LLM-based assessment system depends more on the quality of prompting and implementation strategy than simply on model recency. Open-source models like Llama-UKP, when used with robust methods, can perform comparably to proprietary systems (Yuan and Hu, 2024). Complimenting this, Li’s finding—that Assisted RAE achieved 76.5% accuracy—demonstrates that strategic prompt engineering can be just as influential as acquiring the latest

model updates (Li et al., 2024).

4.1 Essay Grading

On the widely-used ASAP dataset (Automated Student Assessment Prize, a collection of 17,043 student essays across eight prompts with expert human scores). (The Hewlett Foundation, 2012), performance varies significantly across model architectures and implementation approaches. See Appendix A.2 for a detailed description. Xiao et al. (2024)'s dual-process framework using LLaMA3-8B achieved Quadratic Weighted Kappa (QWK) scores of approximately 0.7, approaching state-of-the-art models (QWK = 0.79) while maintaining over 80% score consistency. Similarly, Tang et al. (2024) found that GPT-4 achieved moderate reliability (QWK=0.5677) with criteria-referenced prompts, though still below human reliability benchmarks (QWK=0.6573). In contrast, Kundu and Barbosa (2024)'s evaluation of ChatGPT on the same dataset showed weaker correlation with human scores ($r=0.21-0.23$), though Llama-3 models demonstrated 130–173% improvement over baseline metrics, highlighting the rapid evolution in open-source model capabilities for educational assessment.

Among the prompting methods, Jauhiainen and Garagorry Guerra (2024)'s implementation of verification-based chain-of-thought prompting (see Appendix A.1) with the RAG framework achieved remarkable consistency, with 68.7% of ChatGPT-4 grades remaining stable across multiple evaluations and 72.2% aligning closely with human assessments. This approach parallels Xiao et al. (2024)'s dual-process framework, which distinguishes between a "Fast Module" for rapid predictions and a "Slow Module" for detailed feedback when confidence is low—a design inspired by Kahneman's dual-processing theory (Kahneman, 2011). Both studies demonstrate how thoughtful prompt design can dramatically improve performance even without requiring the most advanced models, with Xiao's open-source implementation achieving a 35% win rate (see Table 5) when compared to GPT-4 explanations despite using the smaller LLaMA3-8B model. Tang et al. (2024) further established that lower temperature settings (0.0) consistently produced better human alignment across models, highlighting how parameter tuning complements prompt engineering in optimizing assessment quality.

Supporting our observation that LLMs in hu-

man grading workflows show particular promise, Xiao et al. (2024)'s human-AI experiments revealed that novice graders improved from QWK 0.53 to 0.66 (approaching expert-level performance of 0.71) when provided with LLM-generated feedback, while experts reached QWK 0.77 with AI assistance. These findings align with Farrokhnia et al. (2024)'s assertion that AI tools can effectively reduce teacher workload while maintaining assessment quality. The complementary relationship between human and AI evaluation extends beyond efficiency gains, with Kundu and Barbosa (2024) noting that humans and LLMs employ distinctly different evaluation criteria—humans prioritizing essay length ($r=0.74$) while LLMs focus more on technical elements like grammar—suggesting that hybrid approaches can provide more comprehensive assessment than either alone.

Interactive assessment frameworks represent an emerging frontier, moving beyond static grading toward dynamic, dialogue-based evaluation systems. Hong et al. (2024)'s CAELF (Contestable AI Evaluation with Logic and Feedback; see Appendix A.4) introduces a multi-agent framework that enables students to challenge grades through structured debate, with Teaching-Assistant Agents discussing essay quality while a Teacher Agent resolves conflicts using principles from computational argumentation (Dung, 1995). When tested on 500 critical thinking essays (Hugging Face, 2023), this approach improved interaction accuracy by 44.6% over GPT-4o while maintaining correct evaluations in 80-90% of cases. More importantly, the system admitted mistakes 10-20% more frequently than baselines, demonstrating improved metacognitive awareness. Human evaluators particularly praised the clarity and actionable nature of the feedback, aligning with advances in LLM-driven formative assessment (Dai et al., 2023).

4.2 Natural Sciences & Technology

In the natural sciences domain, Henkel et al. (2024a) demonstrated that GPT-4 achieved near-human performance on 1,710 K-12 short-answer questions from the Carousel dataset (Cohen's $\kappa = 0.70$ compared to human $\kappa = 0.75$), with metrics of 85% accuracy, 0.87 precision, and 0.85 recall. In contrast, GPT-3.5 only reached a κ of 0.45, highlighting rapid advancements between model generations. Similarly, Tobler (2024)'s GenAI-Based Smart Grading system attained strong alignment with human evaluators (Krippendorff's $\alpha = 0.818$,

95% CI [0.689, 0.926]) in university-level assessments. Further comparisons by [Latif and Zhai \(2024\)](#) revealed that a fine-tuned GPT-3.5-turbo outperformed BERT across six scientific tasks, particularly excelling in multi-class (10.6% improvement) and unbalanced multi-label scenarios. Meanwhile, [Wu et al. \(2024\)](#)'s work on the open-source Mixtral-8x7B-instruct model showed moderate rubric alignment (F1=0.752) and a scoring accuracy of 54.58%. Their "Full-shot + Holistic Rubrics" prompting strategy outperformed both human-created rubrics (50.41%) and non-rubric baselines (33.5%), underscoring the impact of structured prompting on assessment quality.

Notably, efficiency gains in science education are compelling. GPT-4 completed evaluations of 1,710 short-answer questions in approximately 2 hours, compared to 11 hours for manual grading ([Henkel et al., 2024a](#)), and [Tobler \(2024\)](#)'s system also demonstrated significant time savings in university-level assessments. Overall, these findings indicate that carefully structured, rubric-based prompts and advanced LLM architectures not only enhance performance but also offer substantial efficiency improvements in scientific assessments.

4.3 Social Sciences & Humanities

[Lundgren \(Lundgren, 2024\)](#) and [Kostic \(Kostic et al., 2024\)](#) evaluated GPT-4 in advanced humanities assessments using distinct approaches. Lundgren's study of master-level political science essays showed that GPT-4's mean scores (approximately 5.03–5.60) generally aligned with human scores (around 4.95), although interrater reliability was very low (Cohen's $\kappa \leq 0.18$, $\leq 35\%$ agreement). In contrast, Kostic's assessment of German-language business transfer assignments revealed that GPT-4 produced markedly different scores from human evaluators (e.g. 52/50/60 vs. an average human score of about 26). Furthermore, [Kooli and Yusuf \(Kooli and Yusuf, 2024\)](#) reported moderate positive correlations between LLM and human grading (Pearson $r = 0.46$, Spearman $r = 0.518$, $p = 0.008$) for open-ended exam responses, while [Pinto et al. \(Pinto et al., 2023\)](#) observed strong LLM performance on structured exam grading. These results suggest that LLMs tend to evaluate well-defined, bounded responses more reliably than extended analytical writing, which requires nuanced human interpretation.

In addition, GPT-4 appears to prioritize evaluation criteria differently from human graders by

favoring middle-range grades and language quality over the extremes preferred by human evaluators who emphasize analytical depth. Kostic also noted that human evaluators vary widely due to factors such as fatigue and subjectivity, a variability not observed in LLM scoring. Such complementary characteristics indicate the potential for hybrid approaches that integrate human expertise with the computational consistency of LLMs ([Williamson et al., 2012](#)).

4.4 Computer Science

[Xie et al. \(2024\)](#) developed a framework that employs LLMs for rubric generation, initial grading, and post-grading review. Their system iteratively refines rubrics using sampled student responses from the OS and Mohler datasets, and employs group comparisons to enhance assessment consistency. This approach parallels [Grandel et al. \(2024\)](#)'s GreAIter system, which achieved a grading accuracy of 98.21% while reducing grading time by 81.2% for programming assignments.

Performance evaluations across different LLM architectures show that both proprietary and open-source models can yield competitive outcomes. [Yousef et al. \(2025\)](#)'s BeGrading system, based on fine-tuned open-source LLMs, demonstrated only a 19% absolute difference relative to the benchmark Codestral model when grading programming assignments. Similarly, [Koutcheme et al. \(2024\)](#) and [Smolić et al. \(2024\)](#) found that models such as CodeLlama, Zephyr, GPT-3.5, and Gemini offer useful insights and perform comparably in providing feedback on programming assignments.

Targeted prompt engineering, like all other domains has also significantly impacted computer science. [Tian et al. \(2024\)](#)'s systematic evaluation of four prompting strategies revealed that few-shot-rubric prompting consistently outperformed zero-shot approaches, with strong agreement observed for criteria such as Greet Intent (QWK = 0.698) and Default Fallback Intent (QWK = 0.797). These findings are supported by [Duong and Meng \(2024\)](#), who demonstrated that combining GPT-4 with few-shot prompting and Retrieval Augmented Generation (RAG)¹ achieved the highest performance (Pearson correlation of 0.844).

¹See Appendix A.4 for details on RAG implementation.

4.5 Mathematics

In Mathematics, multi-agent systems represent a particularly promising direction, exemplified by (Chu et al., 2024)'s GradeOpt framework, which employs three specialized LLM-based agents—Grader, Reflector, and Refiner—working in concert to optimize mathematics assessment. When evaluated on a dataset of 1,218 teacher responses to five mathematics questions, the GPT-4o-powered system achieved impressive performance metrics (0.85 accuracy and 0.73 Kappa). Further testing on an expanded dataset of 6,541 responses demonstrated significant improvement on specific questions, enhancing accuracy from 0.70 to 0.78 and Kappa scores from 0.52 to 0.64. We also see prompting strategies significantly influencing LLM performance in mathematics assessment, with chain-of-thought approaches demonstrating particularly strong results. Henkel et al. (2024b)'s comprehensive evaluation using the AM-MORE dataset (53,000 question-answer pairs from African middle school students) compared six different grading methods ranging from simple string matching to sophisticated chain-of-thought prompting with GPT-4. The results showed that chain-of-thought prompting excelled particularly on challenging edge cases, achieving 92% accuracy where other methods struggled and boosting overall accuracy from 98.7% to 99.9%. This approach yielded impressive precision (0.97), recall (0.98), and F1 scores (0.98), demonstrating how well-designed prompting strategies can substantially enhance mathematics assessment quality. When implemented within a Bayesian Knowledge Tracing framework ($P(L_0) = 0.4$, $P(T) = 0.05$, $P(S) = 0.299$, $P(G) = 0.299$), these improvements translate to more accurate student mastery estimation, highlighting the practical educational value of such advancements. In mathematics assessment, GPT-4 in particular and its variants demonstrate particularly strong performance across multiple studies and assessment contexts, from GradeOpt's 0.85 accuracy on teacher responses to Henkel's 99.9% accuracy with chain-of-thought prompting on middle school mathematics. These results consistently outperform traditional NLP approaches like SBERT and RoBERTa as demonstrated in Chu's comparative evaluation. The performance advantage appears most pronounced when LLMs are implemented with sophisticated prompting strategies or multi-agent architectures, suggesting that contin-

ued advances in implementation methods may yield further improvements even with existing model architectures.

5 Discussion and Analysis

5.1 The Explainability Imperative

A critical consideration for the widespread adoption of LLM-based assessment is the fundamental need for explainable decisions in educational contexts. Unlike other AI applications, educational assessment directly impacts student learning, progression, and opportunities, making transparency not just desirable but essential. Students require clear explanations of their grades to understand learning gaps and improve performance, while educators need interpretable feedback to guide instructional decisions. The current "black box" nature of leading LLMs presents a significant barrier to educational adoption, as stakeholders cannot adequately justify or contest assessment decisions.

5.2 Patterns in LLM Assessment Performance

The reviewed studies reveal substantial variations in LLM assessment performance across academic disciplines. Figure 2 shows that mathematics and general education yield high human-LLM agreement rates (0.74 and 0.72, respectively), whereas humanities assessments exhibit notably lower alignment (0.46)—a pattern that mirrors our observation that structured formats (see Table 2) offer clearer evaluation criteria than open-ended tasks.

GPT-4 consistently outperforms earlier models in well-structured contexts (Henkel et al., 2024a; Chu et al., 2024; Henkel et al., 2024b), yet its reliability diminishes on complex, subjective tasks. For example, in political science essays, Lundgren (2024) observed that despite similar mean scores, GPT-4 showed very low interrater reliability (Cohen's $\kappa \leq 0.18$, $\leq 35\%$ agreement). Likewise, Kostic et al. (2024) reported that GPT-4 produced scores that differed dramatically from human evaluators in business administration assessments.

Human-LLM agreement studies consistently report moderate alignment: Jauhainen and Garagorry Guerra (2024) found that 72.2% of GPT-4 grades differ by at most one grade from human scores, while Teckwani et al. (2024) noted 71% agreement between LLMs and human graders compared to 80% among humans. Tobler et al. (2024) achieved strong alignment (Krippendorff's $\alpha = 0.818$), though with notable qualitative differences

in rubric interpretation. Comparative evaluations further reveal that, while GPT-4 attains high reliability with criteria-referenced prompts (QWK = 0.5677; Tang et al. (2024)) it still falls slightly short of human benchmarks (QWK = 0.6573). Similarly, Xiao et al. (2024) demonstrated that LLaMA3-8B achieved QWK scores near 0.7 with 80% score consistency, and Morjaria et al. (2024) reported moderate to good correlations ($r = 0.6\text{--}0.7$) in medical education, despite discrepancies in 65–80% of cases.

Figure 3 reveals that single LLM approaches dominate current research (50%), while emerging alternatives such as multi-agent frameworks (10%) and chain-of-thought implementations (6.7%) show superior performance. Overall, these findings suggest that although the gap between human and LLM assessment is narrowing in structured domains, significant differences persist in evaluating complex, open-ended tasks due to varying evaluation approaches and priorities.

5.3 Methodological Approaches and Their Effectiveness

The literature reveals an evolution in prompting techniques, with more sophisticated approaches consistently outperforming simpler implementations across diverse educational contexts. As shown in Figure 4, semi-automated (0.90) and chain-of-thought approaches (0.81) demonstrate the highest human-LLM agreement rates, substantially outperforming single LLM implementations (0.53). These findings align with our categorization of prompting strategies in A.1, where we distinguish between simple zero-shot implementations and more advanced approaches like chain-of-thought. Chain-of-thought and few-shot prompting strategies have proven significantly more effective than zero-shot implementations Wu et al. (2024); Tian et al. (2024); Henkel et al. (2024b) across multiple disciplines as explained in Section 4. Multi-agent frameworks Hong et al. (2024); Chu et al. (2024); Xie et al. (2024) represent another promising methodological direction, allowing for more sophisticated assessment processes that mimic human evaluation workflows, as described in A.4.

Similarly, context-aware approaches that incorporate domain-specific knowledge show particular promise for enhancing assessment quality. Retrieval-augmented generation (RAG) Duong and Meng (2024); Jauhainen and Garagorry Guerra (2024), as detailed in A.4 has emerged as an effective

technique for contextualizing assessments with relevant educational materials. While many people have totally relied on AI to score, we also see many Hybrid human-AI approaches. These approaches yield optimal results in educational assessment by leveraging the complementary strengths of both human evaluators and LLMs. As noted by Xiao et al. (2024), positioning LLMs as assistants rather than replacements enhances overall evaluation quality and efficiency. Kundu and Barbosa (2024) observed that humans and LLMs apply different evaluation criteria—humans prioritizing essay length (with $r = 0.74$) while LLMs focus on technical elements like grammar—suggesting that a combined approach offers a more comprehensive assessment. This complementarity is evident across disciplines; for instance, Morjaria et al. (2024) reported that GPT-4 showed moderate to good correlation with human assessors ($r = 0.6\text{--}0.7$) in medical education, yet discrepancies persisted, and Teckwani et al. (2024) further reinforced the importance of human oversight by finding that human graders demonstrated 80% agreement compared to 71% for LLMs, particularly on higher-order cognitive tasks.

Figure 5 reveals important relationships between assessment types and frameworks, with certain combinations demonstrating particular prevalence. Single LLM approaches dominate essay (3 studies) and short-answer assessment (3 studies), while more specialized frameworks like chain-of-thought appear primarily with mathematical problem solving. These patterns suggest domain-specific optimization of LLM implementation strategies, aligned with our categorization of assessment formats in 2.

6 Conclusion

This review shows that LLM applications in educational assessment are advancing rapidly across various disciplines. Our analysis of 30 studies suggests that these models can help reduce the grading workload while still maintaining quality, particularly in structured contexts—GPT-4, for instance, is already nearing human-level performance in mathematics and science assessments. Innovations like chain-of-thought prompting, multi-agent frameworks, and retrieval-augmented generation are proving to be game changers for improving assessment accuracy.

However, challenges remain. LLMs continue

to struggle with nuanced, subjective evaluations in the humanities and social sciences, and rubric adherence is inconsistent. Technical hurdles, such as processing handwritten responses and handling complex programming tasks, further complicate the picture. Overall, the evidence supports a hybrid human-AI approach: LLMs are most effective when they serve as helpful assistants—automating routine tasks and generating detailed feedback—while human experts handle the more complex evaluations.

While this review focuses specifically on educational assessment, the use of LLMs as evaluators (“LLM as a judge”) is a rapidly growing area across multiple domains including legal document review, content moderation, and research evaluation. Our findings regarding prompting strategies, human-AI collaboration, and reliability challenges likely have broader applicability beyond education, suggesting opportunities for cross-domain learning and methodological transfer.

Looking forward, future research should emphasize domain-specific fine-tuning, standardize prompt engineering practices, and explore multimodal assessment strategies. Moreover, more classroom-based validation studies are needed to assess the long-term impact on learning outcomes. Despite the rapid progress in LLM technology, it is clear that human oversight remains essential for achieving high-quality educational assessment.

7 Limitations

7.1 Methodological Limitations

This review, while comprehensive within its scope, has several methodological limitations that should be acknowledged. Our analysis is based on *limited sample size and generalizability concerns*, as the review includes 30 studies, which may limit the generalizability of our findings, particularly regarding framework performance comparisons shown in Figure 3. Some framework categories are represented by only 1-2 studies, making it difficult to draw robust conclusions about their relative effectiveness. The small sample size is partly due to the nascent nature of LLM applications in educational assessment, with most research emerging only after 2022. Future reviews with larger sample sizes will be needed to validate these preliminary patterns and provide more statistically robust comparisons across framework types.

A significant issue affecting our analysis is

methodological heterogeneity across the reviewed studies. The studies exhibit significant methodological diversity, using different datasets, evaluation metrics, experimental protocols, and LLM configurations. This heterogeneity limits direct comparability and complicates the generalization of findings across studies. For instance, studies within the same discipline often use different datasets (e.g., some essay grading studies use ASAP while others use proprietary datasets), making it challenging to attribute performance differences to framework choices versus dataset characteristics. Additionally, generative AI systems employ various decoding strategies (beam search, temperature settings, top-p sampling) that can significantly impact output quality and consistency, yet these technical parameters are inconsistently reported across studies.

Our organizational approach presents another methodological consideration. While our discipline-based organization provides domain-specific insights valuable for understanding how LLMs perform across different educational contexts, an alternative methodological organization (e.g., by prompting strategies, assessment types, or hybrid architectures) might have enabled different analytical perspectives and cross-cutting insights. This organizational choice may limit the visibility of methodological patterns that transcend disciplinary boundaries. Future reviews could explore cross-cutting methodological themes to complement the domain-specific patterns we identify.

The *under-representation of K-12 studies* in our review likely reflects both limited published research in this educational level and potential search strategy limitations. K-12 educational technology adoption often faces greater institutional barriers, ethical considerations, and regulatory requirements than higher education, potentially slowing research publication in this area. Additionally, our keyword strategy may have inadvertently favored higher education terminology, though we attempted to include broad terms like “educational assessment” and “K-12.” This limitation suggests that our findings may be more applicable to higher education contexts, with K-12 applications requiring additional targeted research.

Another concern affecting the quality of our analysis is *publication quality variability*. Over one-third of the reviewed studies (11 out of 30, or 37%) are preprints that have not undergone formal peer review. While preprints provide valuable insights into cutting-edge research and emerging trends in

LLM-based educational assessment, their inclusion introduces potential quality variability to our analysis. Preprints may contain methodological limitations, incomplete evaluations, or preliminary findings that could change during the peer review process. This limitation is particularly relevant given the rapidly evolving nature of LLM technology, where researchers often share findings quickly through preprint servers to keep pace with technological advances.

7.2 Challenges from the Literature

Several recurring challenges emerge from the literature that must be addressed before widespread educational adoption of LLM assessment systems can occur. A prominent issue is Rubric adherence problems. While [Kostic et al. \(2024\)](#) report poor adherence to assessment criteria in business evaluations despite explicit rubrics, [Tobler \(2024\)](#) observed that AI sometimes adheres more strictly to rubrics than humans, indicating divergent interpretations (see Appendix A.4.2).

Another critical limitation is the *inadequate description of human grader characteristics*. Approximately 40% of studies classify human evaluators as “Unspecified Graders” (see Table 4), making it difficult to contextualize performance metrics and understand the influence of grader expertise—as exemplified by [Xiao et al. \(2024\)](#)’s finding that novice graders scored significantly lower (QWK of 0.53) compared to experts (QWK of 0.71).

A further challenge is the persistence of *grading inconsistencies* across domains. For example, [Lundgren \(2024\)](#) found that GPT-4 exhibits a central tendency bias (favoring middle grades) in political science essays, while [Kooli and Yusuf \(2024\)](#) reported that ChatGPT is more conservative in social science assessments. Similarly, [Smolić et al. \(2024\)](#) noted discrepancies between LLM-provided numerical grades and human standards in programming, highlighting challenges in aligning qualitative feedback with quantitative accuracy.

Significant *technical limitations* also remain for processing specialized content and complex assessment scenarios. [Liu et al. \(2024\)](#) encountered OCR issues in handwritten mathematics, with false positives averaging 27%, and model architecture continues to affect reliability—larger models like GPT-4 consistently outperform smaller ones, although improvements in open-source models and fine-tuning are narrowing this gap. Another concern is the “*black box*” nature of commercial LLMs, which

raises issues of transparency and explainability in educational assessment. The proprietary models (e.g., GPT-3.5/4) offer little insight into their internal decision-making processes, complicating the justification of evaluation decisions. This also leads to *critical governance questions*, as institutions risk disruptions if vendor-controlled systems are modified or discontinued. While promising open-source alternatives ([Yousef et al., 2025](#); [Koutcheme et al., 2024](#)) offer more transparent solutions, they require substantial technical capacity to implement and maintain.

There is also a notable limitation in the *scarcity of real-world, classroom-based implementations*, especially in K-12 contexts. Most studies are controlled experiments, raising concerns about ecological validity and practical challenges. Moreover, there is an *imbalanced focus on educational levels*, with over 60% of studies focusing on higher education while early childhood and primary education remain underexplored. This is particularly problematic given the distinct developmental, pedagogical, and ethical requirements for younger learners, where *ethical and privacy considerations* are especially pronounced. Collectively, these challenges call for further research into standardized methods, transparent AI, and real-world strategies to bridge the gap between experimental promise and practical assessment.

References

- Carousel Learning. 2024. [Carousel short answer dataset](#). Carousel Learning Platform.
- Yucheng Chu, Hang Li, Kaiqi Yang, Harry Shomer, Hui Liu, Yasemin Copur-Gencturk, and Jiliang Tang. 2024. A llm-powered automatic grading framework with human-level guidelines optimization. *arXiv preprint arXiv:2410.02165*.
- 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.
- Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357.
- Ta Nguyen Binh Duong and Chai Yi Meng. 2024. Automatic grading of short answers using large language models in software engineering courses. In *2024 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–10. IEEE.

- B. J. England, J. R. Brigati, E. E. Schussler, and M. M. Chen. 2019. [Student anxiety and perception of difficulty impact performance and persistence in introductory biology courses](#). *CBE—Life Sciences Education*, 18(2):ar21.
- Mohammadreza Farrokhnia, Seyyed Kazem Banihashem, Omid Noroozi, and Arjen Wals. 2024. A swot analysis of chatgpt: Implications for educational practice and research. *Innovations in education and teaching international*, 61(3):460–474.
- Johnbenetic Gnanaprakasam and Ravi Lourdasamy. 2024. The role of ai in automating grading: Enhancing feedback and efficiency. In *Artificial Intelligence and Education-Shaping the Future of Learning*. IntechOpen.
- Skyler Grandel, Douglas C Schmidt, and Kevin Leach. 2024. Applying large language models to enhance the assessment of parallel functional programming assignments. In *Proceedings of the 1st International Workshop on Large Language Models for Code*, pages 102–110.
- Jari J Hakanen, Arnold B Bakker, and Wilmar B Schaufeli. 2006. Burnout and work engagement among teachers. *Journal of school psychology*, 43(6):495–513.
- Sakib Haque, Zachary Eberhart, Aakash Bansal, and Collin McMillan. 2022. Semantic similarity metrics for evaluating source code summarization. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*, pages 36–47.
- Owen Henkel, Libby Hills, Adam Boxer, Bill Roberts, and Zach Levonian. 2024a. 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*, pages 300–304.
- Owen Henkel, Hannah Horne-Robinson, Maria Dyshel, Nabil Ch, Baptiste Moreau-Pernet, and Ralph Abood. 2024b. Learning to love edge cases in formative math assessment: Using the ammore dataset and chain-of-thought prompting to improve grading accuracy. *arXiv preprint arXiv:2409.17904*.
- Shengxin Hong, Chang Cai, Sixuan Du, Haiyue Feng, Siyuan Liu, and Xiuyi Fan. 2024. " my grade is wrong!": A contestable ai framework for interactive feedback in evaluating student essays. *arXiv preprint arXiv:2409.07453*.
- Lory Hough. 2023. [The problem with grading](#). *Ed. Magazine*. Accessed: 2025-03-16.
- Hugging Face. 2023. [Critical thinking essays dataset](#). Hugging Face Datasets Hub.
- Jussi S Jauhiainen and Agustín Garagorry Guerra. 2024. Generative ai in education: Chatgpt-4 in evaluating students' written responses. *Innovations in Education and Teaching International*, pages 1–18.
- Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York, NY, USA.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.
- Rehan Ahmed Khan, Masood Jawaid, Aymen Rehan Khan, and Madiha Sajjad. 2023. Chatgpt-reshaping medical education and clinical management. *Pakistan journal of medical sciences*, 39(2):605.
- Chokri Kooli and Nadia Yusuf. 2024. Transforming educational assessment: Insights into the use of chatgpt and large language models in grading. *International Journal of Human-Computer Interaction*, pages 1–12.
- Milan Kostic, Hans Friedrich Witschel, Knut Hinkelmann, and Maja Spahic-Bogdanovic. 2024. Llms in automated essay evaluation: A case study. In *Proceedings of the AAAI Symposium Series*, volume 3, pages 143–147.
- 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.
- Anindita Kundu and Denilson Barbosa. 2024. Are large language models good essay graders? *arXiv preprint arXiv:2409.13120*.
- Ehsan Latif and Xiaoming Zhai. 2024. Fine-tuning chatgpt for automatic scoring. *Computers and Education: Artificial Intelligence*, 6:100210.
- Kangkang Li, Chengyang Qian, and Xianmin Yang. 2024. Evaluating the quality of student-generated content in learnersourcing: A large language model based approach. *Education and Information Technologies*, 30:2331–2360.
- Tianyi Liu, Julia Chatain, Laura Kobel-Keller, Gerd Kortemeyer, Thomas Willwacher, and Mrinmaya Sachan. 2024. Ai-assisted automated short answer grading of handwritten university level mathematics exams. *arXiv preprint arXiv:2408.11728*.
- Magnus Lundgren. 2024. Large language models in student assessment: Comparing chatgpt and human graders. *arXiv preprint arXiv:2406.16510*.
- 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)*.

- Leo Morjaria, Levi Burns, Keyna Bracken, Anthony J Levinson, Quang N Ngo, Mark Lee, and Matthew Sibbald. 2024. Examining the efficacy of chatgpt in marking short-answer assessments in an undergraduate medical program. *International Medical Education*, 3(1):32–43.
- Gustavo Pinto, Isadora Cardoso-Pereira, Danilo Monteiro, Danilo Lucena, Alberto Souza, and Kiev Gama. 2023. Large language models for education: Grading open-ended questions using chatgpt. In *Proceedings of the XXXVII brazilian symposium on software engineering*, pages 293–302.
- Shreya Shankar, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, JD Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G Parameswaran, and Eugene Wu. 2024a. Spade: Synthesizing data quality assertions for large language model pipelines. *arXiv preprint arXiv:2401.03038*.
- Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024b. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14.
- Ranwir K Sinha, Asitava Deb Roy, Nikhil Kumar, Himel Mondal, and Ranwir Sinha. 2023. Applicability of chatgpt in assisting to solve higher order problems in pathology. *Cureus*, 15(2).
- Einar M Skaalvik and Sidsel Skaalvik. 2017. Dimensions of teacher burnout: Relations with potential stressors at school. *Social Psychology of Education*, 20:775–790.
- Ema Smolić, Marko Pavelić, Bartol Boras, Igor Mekterović, and Tomislav Jaguš. 2024. Llm generative ai and students’ exam code evaluation: Qualitative and quantitative analysis. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pages 1261–1266. IEEE.
- Xiaoyi Tang, Hongwei Chen, Daoyu Lin, and Kexin Li. 2024. Harnessing llms for multi-dimensional writing assessment: Reliability and alignment with human judgments. *Heliyon*, 10(14).
- Swapna Haresh Teckwani, Amanda Huee-Ping Wong, Nathasha Vihangi Luke, and Ivan Cherh Chiet Low. 2024. Accuracy and reliability of large language models in assessing learning outcomes achievement across cognitive domains. *Advances in Physiology Education*, 48(4):904–914.
- The Hewlett Foundation. 2012. [Automated student assessment prize \(asap\) dataset](#). Kaggle.
- Xiaoyi Tian, Amogh Mannekote, Carly E Solomon, Yukyeong Song, Christine Fry Wise, Tom Mcklin, Joanne Barrett, Kristy Elizabeth Boyer, and Maya Israel. 2024. Examining llm prompting strategies for automatic evaluation of learner-created computational artifacts. In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 698–706.
- Samuel Tobler. 2024. Smart grading: A generative ai-based tool for knowledge-grounded answer evaluation in educational assessments. *MethodsX*, 12:102531.
- Catrin Wigfall. 2020. [Grading standards do impact student achievement](#). Accessed: 2025-03-16.
- 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.
- 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. *arXiv preprint arXiv:2407.18328*.
- Changrong Xiao, Wenxing Ma, Qingping Song, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Qi Fu. 2024. Human-ai collaborative essay scoring: A dual-process framework with llms. *arXiv preprint arXiv:2401.06431*.
- Wenjing Xie, Juxin Niu, Chun Jason Xue, and Nan Guan. 2024. Grade like a human: Rethinking automated assessment with large language models. *arXiv preprint arXiv:2405.19694*.
- Mina Yousef, Kareem Mohamed, Walaa Medhat, Ensaf Hussein Mohamed, Ghada Khoriba, and Tamer Arafa. 2025. Begrading: large language models for enhanced feedback in programming education. *Neural Computing and Applications*, 37(2):1027–1040.
- Bo Yuan and Jiazi Hu. 2024. An exploration of higher education course evaluation by large language models. *arXiv preprint arXiv:2411.02455*.

A Terminology and Definitions

This appendix provides comprehensive definitions for key terms, methodologies, and frameworks referenced throughout this review, offering detailed context beyond the condensed definitions in the main text.

A.1 Prompting Strategies

A.1.1 Zero-shot prompting

Direct instruction to the LLM to perform an assessment task without providing examples or demonstrations. The model relies entirely on its pre-training knowledge to understand the assessment criteria and generate appropriate evaluations. [Tang et al. \(2024\)](#) examined this approach for essay evaluation, finding it achieved lower reliability

(QWK=0.4321) compared to few-shot approaches. Zero-shot prompting represents the simplest implementation but typically under-performs more sophisticated strategies, particularly for complex or domain-specific assessments.

A.1.2 Few-shot prompting

Providing the LLM with a limited number (typically 3-6) of example question-answer pairs and their corresponding evaluations before asking it to assess new responses. This approach establishes a pattern for the model to follow when generating its own assessments. [Tian et al. \(2024\)](#) demonstrated that few-shot-rubric prompting consistently outperformed zero-shot approaches when assessing chatbot projects, with particularly strong performance in structured dimensions like Greet Intent (QWK=0.698) and Default Fallback Intent (QWK=0.797). [Duong and Meng \(2024\)](#) found that GPT-4 with 6 examples achieved a Pearson correlation of 0.694 with human graders, substantially outperforming simpler implementations.

A.1.3 Chain-of-thought prompting

Guiding the LLM to articulate step-by-step reasoning before providing a final assessment, mimicking human cognitive processes in evaluation. This approach is particularly effective for mathematical and logical evaluations requiring multi-step reasoning. [Henkel et al. \(2024b\)](#) used this method on the AMMORE dataset of 53,000 question-answer pairs from African middle school students, showing it increased mathematics assessment accuracy from 98.7% to 99.9% and was especially effective for complex edge cases (92% accuracy where other methods struggled). While more computationally intensive than simpler prompting methods, chain-of-thought approaches consistently demonstrate superior performance for complex assessment tasks requiring logical reasoning.

A.1.4 Reason-Act-Evaluate (RAE) prompting

A structured three-stage process where the LLM first reasons about assessment criteria (contemplating evaluation dimensions and standards), then performs the actual assessment (applying these criteria to the student response), and finally reviews its own assessment for accuracy, consistency, and adherence to rubrics. [Li et al. \(2024\)](#) developed this approach for evaluating student-generated content, achieving 76.5% accuracy across 1,235 articles with particularly strong performance in structured dimensions like logical reasoning ($\rho = 0.824$).

This technique incorporates meta-cognitive awareness into the assessment process, enabling self-correction and improved reliability.

A.1.5 Rubric-guided prompting

Explicitly incorporating detailed assessment rubrics into LLM prompts, providing structured evaluation criteria that guide the model's judgment. This approach improves alignment with human evaluation standards by making assessment criteria explicit rather than implied. [Morjaria et al. \(2024\)](#) found this approach significantly reduced score inflation tendencies when using ChatGPT-4 to evaluate medical students' short-answer assessments, achieving moderate to good correlation ($r=0.6-0.7$) with human assessors. Similarly, [Yuan and Hu \(2024\)](#) demonstrated that rubric incorporation enabled Llama-UKP models to achieve remarkable correlation with human evaluators (Spearman: 0.843) when assessing higher education courses.

A.2 Dataset

A.2.1 ASAP dataset

The Automated Student Assessment Prize dataset, released by the Hewlett Foundation in 2012 ([The Hewlett Foundation, 2012](#)), containing 17,043 student essays across eight distinct prompts with expert human scores. Each prompt represents a different essay type (e.g., persuasive, source-based, narrative) and grade level (ranging from grade 7 to 10), with varying length requirements and scoring scales. This comprehensive collection has become the standard benchmark for automated essay scoring systems, enabling direct comparison of different approaches. Studies by [Xiao et al. \(2024\)](#), [Tang et al. \(2024\)](#), and [Kundu and Barbosa \(2024\)](#) used this dataset to evaluate LLM essay assessment capabilities, with [Xiao et al. \(2024\)](#)'s implementation achieving QWK scores of approximately 0.7, approaching state-of-the-art performance (QWK 0.79).

A.2.2 ASAP++ dataset

An extension of the original ASAP dataset developed by [Mathias and Bhattacharyya \(2018\)](#) that enriches the essays with additional attribute scores beyond the holistic ratings in the original dataset. These attributes include content, organization, word choice, sentence fluency, conventions, and prompt adherence. [Kundu and Barbosa \(2024\)](#) used this enhanced dataset to evaluate LLM assessment capabilities across multiple dimensions of

writing quality, providing more nuanced analysis of model performance on different aspects of essay evaluation.

A.2.3 Carousel dataset

A collection of 1,710 K-12 short-answer questions from science and history subjects developed by Carousel Learning (Carousel Learning, 2024). The dataset includes multiple student responses to each question along with expert human evaluations based on detailed rubrics. Questions span multiple grade levels and subject areas, providing a diverse testbed for short-answer assessment capabilities. Henkel et al. (2024a) used this dataset to evaluate GPT-4's performance on K-12 short-answer grading, finding near-human performance (Cohen's $\kappa = 0.70$ compared to human $\kappa = 0.75$).

A.2.4 AMMORE dataset

The African Middle-school Math Open Response Evaluation dataset contains 53,000 question-answer pairs from African middle school students across multiple mathematical topics. This comprehensive collection includes diverse response formats and challenging edge cases that test the limits of automated assessment capabilities, including unconventional solution methods and partial understanding demonstrations. Henkel et al. (2024b) used this dataset to evaluate various assessment approaches, finding that chain-of-thought prompting achieved 99.9% overall accuracy and 92% accuracy on challenging edge cases where simpler methods struggled.

A.2.5 Mohler dataset

A computer science short-answer dataset containing 2,273 student responses to technical questions with expert human grades. This dataset features specialized computer science content requiring domain-specific knowledge for accurate assessment, including algorithm descriptions, theoretical explanations, and applied problem-solving. Xie et al. (2024) and Duong and Meng (2024) used this dataset to evaluate LLM performance on computer science assessment, with Duong and Meng (2024) achieving a Pearson correlation of 0.694 using GPT-4 with few-shot prompting.

A.2.6 OS dataset

A dataset of operating systems concept questions and student responses used by Xie et al. (2024) to evaluate their multi-agent assessment system.

This specialized collection focuses on technical computer science concepts and includes varied response types requiring domain-specific knowledge for accurate evaluation. The dataset exemplifies the challenges of assessing technical subject matter where specialized terminology and conceptual precision are essential for accurate evaluation.

A.3 Framework Definitions

- **Mixed-initiative:** Systems combining human and AI decision-making with dynamic role allocation
- **OCR+LLM:** Optical Character Recognition integrated with Large Language Models for handwritten content
- **Semi-automated:** Human-AI collaborative systems where AI provides initial assessment subject to human review
- **Multi-agent:** Multiple LLM instances with specialized roles working collaboratively

A.4 Specialized Concepts

A.4.1 Criteria drift

The phenomenon is where evaluation standards evolve or shift during the assessment process, potentially compromising consistency and fairness. This can occur with both human and LLM evaluators and represents a significant challenge for maintaining reliable assessment standards. Shankar et al. (2024a) identified this as a fundamental challenge in LLM assessment, where initial evaluation criteria may be applied differently to later responses. Criteria drift manifests in several forms:

- **Standard inflation/deflation:** Gradual shifting of grading standards to become more lenient or strict over time.
- **Criteria reinterpretation:** Subtle changes in how specific rubric elements are interpreted across different responses.
- **Priority shifting:** Changes in the relative importance assigned to different evaluation criteria during the assessment process.
- **Context effects:** Earlier responses influencing the evaluation of later responses through comparative judgment rather than fixed standards.

Addressing criteria drift requires explicit metacognitive awareness and structured review processes, which multi-agent LLM frameworks like [Chu et al. \(2024\)](#)'s GradeOpt implement through specialized roles such as the "Reflector" agent dedicated to consistency monitoring.

A.4.2 Rubric-based approach

Assessment methodologies that employ structured evaluation frameworks with explicitly defined criteria and performance levels to ensure consistent, transparent evaluation. In LLM assessment, rubric-based approaches involve providing models with these structured frameworks to guide evaluation. Key elements include:

- **Dimension specification:** Clearly identified aspects of performance to be evaluated (e.g., content coverage, organizational structure, technical accuracy, language use).
- **Performance descriptors:** Explicit descriptions of what constitutes different quality levels for each dimension, typically ranging from excellent to unsatisfactory.
- **Weighting schemes:** Optional specifications regarding the relative importance of different dimensions in the overall assessment.
- **Scoring mechanics:** Clear instructions on how to convert qualitative judgments into numerical scores, ensuring consistent quantification of performance.

Studies by [Morjaria et al. \(2024\)](#), [Wu et al. \(2024\)](#), and [Yuan and Hu \(2024\)](#) demonstrated that incorporating detailed rubrics significantly improved LLM assessment alignment with human evaluation, particularly for complex responses requiring multi-dimensional evaluation. [Morjaria et al. \(2024\)](#) specifically found that rubric incorporation reduced ChatGPT-4's tendency toward score inflation in medical education contexts.

A.4.3 Assisted RAE approach

An enhancement to the basic Reason-Act-Evaluate framework developed by [Li et al. \(2024\)](#) that incorporates metadata analysis and additional contextual information to improve assessment quality. This approach augments the three-stage RAE process (reasoning about criteria, performing assessment, evaluating quality) with supplementary information

about the assessment context, student characteristics, or relevant educational standards. The assisted version achieved 76.5% accuracy when evaluating student-generated content across 1,235 articles, with particularly strong performance in structured dimensions like logical reasoning ($\rho = 0.824$).

A.4.4 CAELF framework

Contestable AI Evaluation with Logic and Feedback, a multi-agent framework developed by [Hong et al. \(2024\)](#) that enables students to challenge AI-generated grades through structured debate. The system employs teaching assistant agents for initial evaluation and discussion of contested grades, while a teacher agent resolves conflicts using principles from computational argumentation theory ([Dung, 1995](#)). When tested on 500 critical thinking essays, this approach improved interaction accuracy by 44.6% over GPT-4o alone, maintained correct evaluations in 80-90% of cases, and admitted mistakes 10-20% more frequently than baselines, demonstrating improved metacognitive awareness.

A.4.5 Retrieval-Augmented Generation (RAG)

Enhancing LLM evaluation by retrieving and incorporating relevant reference materials from external sources to contextualize the assessment. This approach integrates domain-specific knowledge beyond the model's training data, improving performance on specialized subjects. [Duong and Meng \(2024\)](#) applied this method to software engineering course assessment, dramatically improving Pearson correlation from 0.694 to 0.844 by incorporating course materials into the evaluation process. RAG implementations are particularly valuable for domain-specific assessments where specialized knowledge or context is essential for accurate evaluation.

A.4.6 Multi-agent frameworks

Using multiple specialized LLM instances that perform different aspects of the assessment process in collaboration, mimicking human evaluation workflows with distinct roles. These frameworks typically include components like initial graders, reviewers, and arbitrators that communicate to produce a refined assessment. [Hong et al. \(2024\)](#)'s CAELF framework exemplifies this approach, employing teaching assistant agents for initial evaluation and a teacher agent to resolve conflicts, improving interaction accuracy by 44.6% over single-agent approaches. Similarly, [Chu et al. \(2024\)](#)'s

GradeOpt employed three distinct agents—grader, reflector, and refiner—working collaboratively to achieve 0.85 accuracy and 0.73 Kappa in mathematics assessment. While more complex to implement, multi-agent frameworks consistently demonstrate superior performance, particularly for nuanced assessment tasks requiring multiple perspectives.

A.4.7 Automated Short Answer Grading (ASAG)

A field focused on using computational methods to automatically evaluate student responses to short-answer questions. ASAG systems typically analyze the semantic content of responses against reference answers or rubrics to determine correctness, completeness, and relevance. LLM-based ASAG frameworks like GradeOpt (Chu et al., 2024) represent advanced approaches that can evaluate nuanced understanding beyond simple keyword matching.

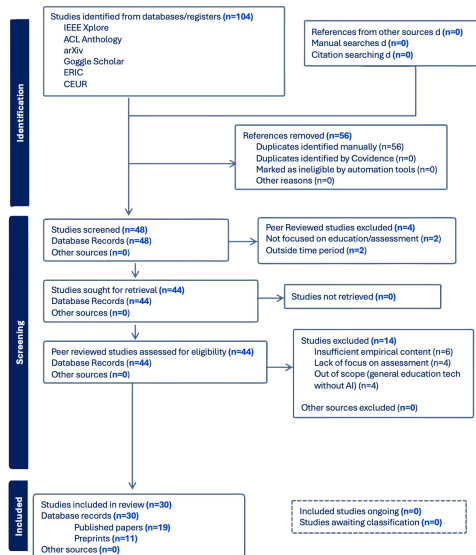


Figure 1: PRISMA flow diagram showing the study selection process.

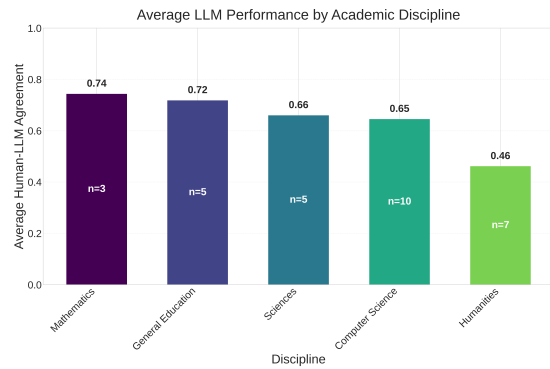


Figure 2: Average LLM Performance by Academic Discipline.

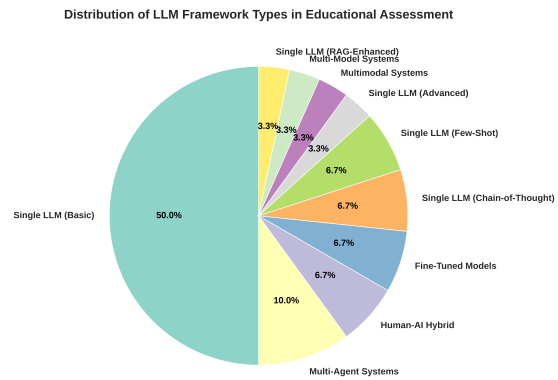


Figure 3: Distribution of LLM Framework Types in Educational Assessment (see Appendix A.3).

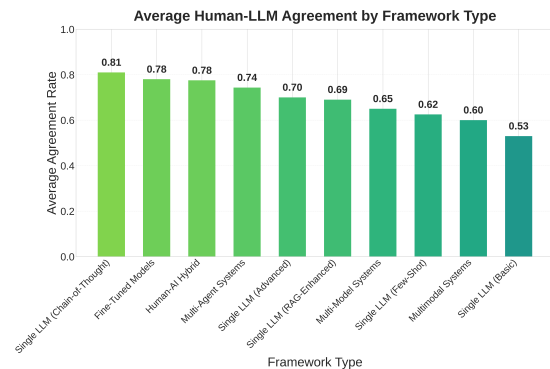


Figure 4: Average Human-LLM Agreement by Framework Type.

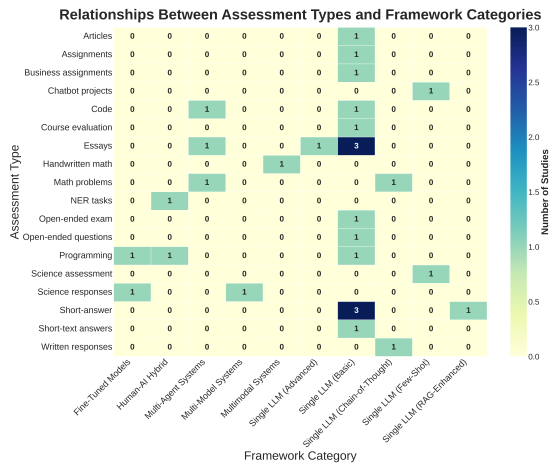


Figure 5: Relationships Between Assessment Types and Frameworks. Cell values represent the number of studies using each assessment type-framework combination (0 = no studies, 1 = one study, 2 = two studies, etc.).

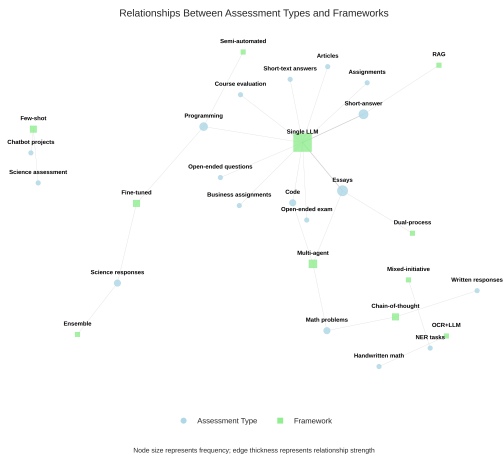


Figure 6: Network Visualization of Assessment Types and Frameworks Relationships.

Table 1: Search sources and terms used to extract peer-reviewed scientific literature on large language models in educational assessment (January 1, 2022 – January 14, 2025).

Source	Date of Search	Search Terms
Google Scholar	Jan 14, 2025	("large language models" OR "educational assessment" OR "automated grading" OR "essay scoring" OR "student feedback" OR "ChatGPT" OR "GPT-4" OR "short answer evaluation" OR "AI in education")
arXiv	Jan 14, 2025	("large language models" OR "educational assessment" OR "automated grading" OR "essay scoring" OR "student feedback" OR "ChatGPT" OR "GPT-4" OR "short answer evaluation" OR "AI in education")
IEEE Xplore	Jan 14, 2025	("large language models" OR "educational assessment" OR "automated grading" OR "essay scoring" OR "student feedback" OR "ChatGPT" OR "GPT-4" OR "short answer evaluation" OR "AI in education")
ACL Anthology	Jan 14, 2025	("large language models" OR "educational assessment" OR "automated grading" OR "essay scoring" OR "student feedback" OR "ChatGPT" OR "GPT-4" OR "short answer evaluation" OR "AI in education")
ERIC (Education Resources Info Center)	Jan 14, 2025	("large language models" OR "educational assessment" OR "automated grading" OR "essay scoring" OR "student feedback" OR "ChatGPT" OR "GPT-4" OR "short answer evaluation" OR "AI in education")
CEUR	Jan 14, 2025	("large language models" OR "educational assessment" OR "automated grading" OR "essay scoring" OR "student feedback" OR "ChatGPT" OR "GPT-4" OR "short answer evaluation" OR "AI in education")

Table 2: Assessment Types in LLM Evaluation Studies.

Assessment Type	Description	Typical Evaluation Method
Multiple-Choice Questions (MCQs)	Structured questions with predefined answer options where students select from available choices.	Binary correctness evaluation; typically automated using answer keys
Short-Answer Questions	Brief text responses (typically 1-5 sentences) addressing specific, bounded questions with relatively constrained correct answers.	Rubric-based evaluation against expected key concepts or knowledge points
Essay Assessments	Extended written responses (typically >300 words) requiring development of arguments, analysis, or synthesis of information.	Multi-dimensional rubrics evaluating content quality, structure, argumentation, and language use
Programming Assignments	Code writing tasks requiring functional implementation of algorithms or solutions to computational problems.	Evaluation of correctness, efficiency, readability, and adherence to programming standards
Mathematics Assessments	Problems requiring mathematical reasoning, calculation, and demonstration of procedural or conceptual understanding.	Step-by-step evaluation of solution process, correctness, and mathematical reasoning
Handwritten Assessments	Written responses composed by hand rather than digitally, requiring OCR processing before LLM evaluation.	Content evaluation following digitization; may involve image processing and character recognition

Table 3: Education Levels in LLM Evaluation Studies.

Category	Typical Age Range	Description	Examples in Studies
Early Education	Ages 3-5	Pre-primary education including kindergarten and preparatory programs	Limited representation in current studies
Primary Education	Ages 6-10	Elementary school (grades 1-5 in many systems)	Wu et al. (2024), Henkel et al. (2024a)
Secondary Education	Ages 11-18	Middle and high school (grades 6-12 in many systems)	Henkel et al. (2024a), Latif and Zhai (2024)
Undergraduate Education	Ages 18-22	Bachelor’s degree programs and equivalent tertiary education	Yuan and Hu (2024), Tobler (2024), Kooli and Yusuf (2024)
Graduate Education	Ages 22+	Master’s and doctoral programs	Lundgren (2024), Morjaria et al. (2024)
Professional Education	Various (typically 18+)	Specialized training for specific professions (medical, engineering, etc.)	Morjaria et al. (2024), Sinha et al. (2023)

Table 4: Human Annotator Categories in LLM Evaluation Studies.

Category	Definition	Typical Characteristics
Expert Evaluators	Individuals with advanced qualifications and substantial experience in the subject matter and assessment context	PhD or equivalent qualification; 5+ years teaching/evaluation experience; specialized domain knowledge
Experienced Educators	Teachers or instructors with formal teaching qualifications and moderate experience	Master’s degree or equivalent; 2-5 years teaching experience; formal pedagogical training
Novice Evaluators	Individuals with basic subject knowledge but limited assessment experience	Bachelor’s degree or equivalent; <2 years assessment experience; may include teaching assistants or student peers
Field Practitioners	Domain experts who may lack formal education qualifications but possess practical expertise	Industry experience; professional certifications; variable teaching experience
Unspecified Graders	Studies where human grader qualifications are not explicitly described	Unknown qualifications and experience levels; represents a methodological limitation in some studies

Table 5: Evaluation Metrics in LLM Assessment Studies.

Metric	Description	Typical Interpretation
Cohen’s Kappa (κ)	Measures interrater reliability between two raters, accounting for agreement occurring by chance. Scale from -1 to 1, with 1 representing perfect agreement.	< 0.40: Poor agreement 0.40 – 0.75: Fair to good > 0.75: Excellent agreement
Quadratic Weighted Kappa (QWK)	Extension of Cohen’s Kappa that assigns different weights to disagreements based on their severity. Common in essay scoring evaluation.	Similar to Cohen’s Kappa, but with increased sensitivity to disagreement magnitude
Krippendorff’s Alpha (α)	Reliability coefficient suitable for multiple raters and various measurement levels. Ranges from 0 to 1.	< 0.67: Insufficient 0.67 – 0.80: Tentative > 0.80: Reliable
Pearson Correlation (r)	Measures linear correlation between two variables. Ranges from -1 to 1.	< 0.40: Weak correlation 0.40 – 0.70: Moderate correlation > 0.70: Strong correlation
Spearman Correlation (ρ)	Measures monotonic relationships between ranked variables. Useful for ordinal data like grades.	Similar to Pearson, but for ranked data
Accuracy	Percentage of correctly identified instances. Simple measure for classification tasks.	Context-dependent; higher is better
F1 Score	Harmonic mean of precision and recall. Balances false positives and false negatives.	0 to 1 scale; higher is better
Win Rate	Percentage of instances where the LLM’s assessment is preferred over alternatives in comparative evaluations.	Context-dependent; used primarily in comparative studies

Table 6: Summary of LLM Educational Assessment Research.

Reference	Discipline / Subject	Data	Data Availability	Techniques	Results
Teckwani et al. (2024)	General Education	117 assignments aligned with Bloom’s taxonomy	Not mentioned	LLM evaluation (GPT-3.5, GPT-4o, Gemini)	LLMs: moderate consistency (Gemini: 71%, $r = 0.672$); Human: superior reliability (80% agreement, $r = 0.936$). It was found that LLMs struggled with higher-order tasks; poor human alignment ($\leq 44\%$)
Morjaria et al. (2024)	Medical Education	Medical students’ short-answer assessments	Not mentioned	ChatGPT-4 as grading assistant	Moderate to good correlation with humans ($r = 0.6-0.7$); Score discrepancies in 65–80% of cases. Including rubrics reduced ChatGPT’s score inflation tendency
Yuan and Hu (2024)	Higher Education	100 Chinese university courses	Not mentioned	GPT-4o, Kimi, and Llama models	Llama-UKP had strong correlation with human evaluations (Spearman: 0.843)
Li et al. (2024)	Educational Content	1,235 student articles	Not mentioned	“Reason-Act-Evaluate” prompt with metadata analysis	76.5% accuracy. Strong correlation with expert evaluations in structured dimensions (logic: $\rho = 0.824$)
Shankar et al. (2024b)	General (NLP)	Medical transcripts and product descriptions	Not mentioned	EvalGen tool with GPT-4	Criteria drift identified. Furthermore, revealed interdependence of criteria and outputs
Xiao et al. (2024)	Essay Grading	ASAP dataset and private Chinese dataset	ASAP: Publicly available	Dual-process framework with LLaMA3-8B	QWK scores (~ 0.7) close to SOTA (QWK 0.79); $>80\%$ score consistency. Novices improved from QWK 0.53 to 0.66 with AI assistance
Hong et al. (2024)	Essay Grading	500 critical thinking essays	Publicly available (Hugging Face, 2023)	CAELF multi-agent framework	Improved interaction accuracy by 44.6% over GPT-4o. Maintained correct evaluations in 80–90% of cases
Kundu and Barbosa (2024)	Essay Grading	ASAP and ASAP++ datasets	Publicly available	ChatGPT and Llama models	Weak correlation with human scores (ChatGPT: $r = 0.21-0.23$). It was found that LLMs excel in error detection but prioritize different criteria than humans
Jauhiainen and Garagorry Guerra (2024)	General Education	54 student responses	Not mentioned	ChatGPT-4 with verification-based chain-of-thought	68.7% grade consistency; 72.2% alignment with humans. Discrepancies in the model are addressable through prompt refinement
Tang et al. (2024)	Essay Grading	ASAP dataset (1,730 essays)	Publicly available	GPT-3.5, GPT-4, Claude 2	GPT-4: highest reliability (QWK = 0.5677). Lower temperature settings (0.0) produced better human alignment

Table 6: Summary of LLM Educational Assessment Research (continued).

Reference	Discipline/Subject	Data	Data Availability	Techniques	Results
Henkel et al. (2024a)	K-12 Science/History	1,710 short-answer questions (Carousel dataset)	Publicly available	GPT-4 and GPT-3.5	GPT-4: near-human performance (Cohen's $\kappa = 0.70$ vs. human $\kappa = 0.75$). 85% precision, 0.87 precision, 0.85 recall; automated grading required 2 hours vs. 11 hours manually
Wu et al. (2024)	Physics (Middle School)	12 physics science assessment items	Not mentioned	Mixtral-8x7B-instruct with few-shot prompting	Best configuration achieved 54.58% scoring accuracy. Strong correlation between human-aligned rubrics and accurate grading
Tobler (2024)	General Education	29 university students' responses	Consent required	GenAI-Based Smart Grading with GPT-4	Strong alignment with human grading ($\alpha = 0.818$). Based on results from the study, AI exhibited stricter adherence to rubrics
Latif and Zhai (2024)	Science Education	2,600 middle/high school responses	Not mentioned	Fine-tuned GPT-3.5-turbo vs. BERT	GPT-3.5: mean precision of 0.915 vs. BERT: 0.838. GPT-3.5 showed strength in multi-class tasks (10.6% improvement)
Lundgren (2024)	Political Science	60 master-level essays	Not mentioned	GPT-4 with four prompt types	Low interrater reliability (Cohen's $\kappa \leq 0.18$). GPT-4 favored middle grades; detailed prompts didn't improve accuracy
Kostic et al. (2024)	Business Administration	German-language business assignments	Not mentioned	GPT-4 with three prompt variations	Unreliable grades (e.g., overscoring). This study revealed that the automated system displayed poor rubric adherence, and is inadequate for nuanced assessment
Kooli and Yusuf (2024)	Social Science	25 open-ended exam responses	Not mentioned	ChatGPT vs. human grader	Moderate positive correlation (Pearson $r = 0.46$). ChatGPT found to be more conservative and variable than humans
Xie et al. (2024)	Computer Science	OS and Mohler datasets	Not mentioned	Multi-agent system for rubric generation	Improved grading consistency. However, challenges in achieving complete fairness and rubric precision
Yousef et al. (2025)	Programming Education	Python and Java assignments	Not mentioned	BeGrading system with fine-tuned LLMs	19% absolute difference rate. Fine-tuning small models improved performance
Koutcheme et al. (2024)	Programming Education	Programming assignments	Not mentioned	CodeLlama and Zephyr	Zephyr models performed similarly to proprietary models. Open-source LLMs can offer meaningful student feedback

Table 6: Summary of LLM Educational Assessment Research (continued).

Reference	Discipline / Subject	Data	Data Availability	Techniques	Results
Smolić et al. (2024)	Programming Education	Student code submissions	Not mentioned	GPT-3.5 and Gemini	Useful insights for code review; numerical grades inconsistent with human standards
Schneider et al. (2023)	Computer Science	Short-text answers from university courses	Not mentioned	ChatGPT-3.5	Inconsistent grading; struggled with contextual understanding and course-specific knowledge
Duong and Meng (2024)	Software Engineering	Mohler Dataset (2,273 answers) and SE Dataset (421 answers)	Not mentioned	Embedding-based and completion-based methods	GPT-4 with 6 examples: Pearson correlation of 0.694. GPT-4 superior to GPT-3.5 but at higher cost
Grandel et al. (2024)	Programming Education	Programming assignments	Not mentioned	GreAIter semi-automated system with ChatGPT-4	98.21% grading accuracy; reduced grading time by 81.2%
Tian et al. (2024)	AI Education	75 chatbot projects	Not mentioned	GPT-4 with four prompting strategies	Good performance in some dimensions (QWK=0.698). Few-shot-rubric prompting outperformed zero-shot
Pinto et al. (2023)	Software Engineering	Responses to open-ended questions	Not mentioned	ChatGPT	Aligned with expert evaluations; good at identifying misunderstandings
Gao et al. (2023)	Mechanical Engineering	Quiz dataset (70 students) and Activity dataset (85–95 students)	Not mentioned	7 NLP models (BERT, T5, etc.)	PromCSE excelled in binary tasks. NLP models struggle with precision and complex questions
Chu et al. (2024)	Mathematics	1,218 teacher responses and 6,541 teacher responses	Not mentioned	GradeOpt multi-agent framework with GPT-4o	0.85 accuracy and 0.73 Kappa. More effective than traditional methods
Liu et al. (2024)	University Mathematics	Handwritten calculus exam (54 students)	Consent required	GPT-4 with Mathpix and GPT-4V for OCR	Accuracy: 0.59 to 0.62. Whole-page OCR outperformed answer-box methods
Henkel et al. (2024b)	Middle School Mathematics	AMMORE dataset (53,000 question-answer pairs)	Publicly available	Chain-of-thought prompting and LLMs	92% accuracy on edge cases; 99.9% overall accuracy. Chain-of-thought prompting excelled but required more processing time

Unsupervised Sentence Readability Estimation Based on Parallel Corpora for Text Simplification

Rina Miyata*

Ehime University
miyata@ai.cs.ehime-u.ac.jp

Toru Urakawa

The Asahi Shimbun Company
urakawa-t@asahi.com

Hideaki Tamori

The Asahi Shimbun Company
tamori-h@asahi.com

Tomoyuki Kajiwara

Ehime University / The University of Osaka
kajiwara@cs.ehime-u.ac.jp

Abstract

We train a relative sentence readability estimator from a corpus without absolute sentence readability. Since sentence readability depends on the reader’s knowledge, objective and absolute readability assessments require costly annotation by experts. Therefore, few corpora have absolute sentence readability, while parallel corpora for text simplification with relative sentence readability between two sentences are available for many languages. With multilingual applications in mind, we propose a method to estimate relative sentence readability based on parallel corpora for text simplification. Experimental results on ranking a set of English sentences by readability show that our method outperforms existing unsupervised methods and is comparable to supervised methods based on absolute sentence readability.

1 Introduction

Readability estimation of text, such as words, sentences, and documents, is applied to assist in text recommendation and simplification for a wide range of readers, including children (Xu et al., 2015), language learners (Xia et al., 2016), and people with cognitive disabilities (Yaneva et al., 2017), according to their language abilities. We work on readability estimation for sentences, which are the main units in the text simplification task (Alva-Manchego et al., 2020).

Since sentence readability depends on the reader’s knowledge, objective and absolute readability assessments require costly annotation by experts. Therefore, corpora annotated with absolute readability are limited to a scale of $1k$ to

$10k$ sentences even in English (Stajner et al., 2017; Arase et al., 2022), and are rarely available in other languages. This low-resource problem hinders research and development of high-quality supervised sentence readability estimation.

In this study, we train a relative sentence readability estimator based on labeled corpora for relative sentence readability, which are more accessible than those with absolute sentence readability labels. Our proposed method estimates which of two given sentences is more readable based on pairs of complex sentences and simpler sentences in parallel corpora for text simplification. The estimator is then applied to pairwise comparisons of a given set of sentences to rank them in terms of readability.

Experimental results on ranking a set of English sentences by readability show that the proposed method outperforms existing unsupervised methods. In addition, our proposed method achieved performance comparable to supervised methods that consider absolute sentence readability.

2 Related Work

For estimating text readability, supervised methods (Vajjala and Lučić, 2018; Deutsch et al., 2020) have been proposed that consider readability indices, linguistic features, and language model scores. Since they are based on corpora annotated with absolute sentence readability, they can not apply to languages without labeled corpora available.

Unsupervised methods such as FKGL (Kincaid et al., 1975) and other readability metrics and ranking methods based on relative readability estimation (Tanaka-Ishii et al., 2010) have been proposed. However, they are targeted at documents and are not applicable to sentences.

*Work done during an internship at The Asahi Shimbun Company.

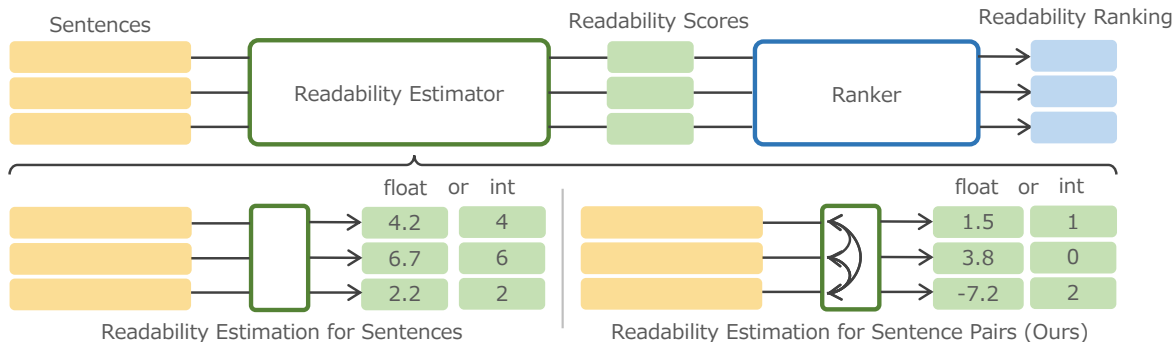


Figure 1: System overview

3 Method

We first train a relative sentence readability estimator that estimates which of two given sentences is more readable, based on parallel corpora for text simplification. The estimator is then applied to pairwise comparisons of a given set of sentences to rank them in terms of readability. The overall system process consists of estimating readability by using the Readability Estimator and then ranking them, as shown in the upper part of Figure 1.

3.1 Relative Sentence Readability Estimator

Our sentence readability estimator is based on fine-tuning pre-trained masked language models (Devlin et al., 2019). From sentence pairs in parallel corpora for text simplification, we create input sequences of “[CLS] complex sentence [SEP] simple sentence” with special tokens indicating the beginning and sentence boundaries. Note that in 50% of the input sequences, the positions of complex and simple sentences are swapped. We train a binary classifier with this dataset to estimate which of two given sentences is more readable.

3.2 Readability Ranking

We rank each sentence in a given set of shuffled sentences by readability using a pairwise comparison method. In other words, the relative sentence readability is estimated for all combinations of two sentences in a given set of sentences, as shown in the bottom right-hand corner of Figure 1. The readability of a sentence is given as an integer, if there is a tie, it depends on the order of input. Finally, we obtain a ranking according to the probability that each sentence is estimated to be more readable.

	Train	Valid	Test
Newsela	385,270	42,323	43,171
CEFR-SP	-	-	17,676

Table 1: Corpus size

4 Experiments

In this section, we experiment with ranking a set of English sentences by readability. Following previous studies of document readability rankings, we evaluated rankings according to four metrics: normalized discounted cumulative gain (NDCG), Spearman’s correlation (ρ), Kendall’s correlation (τ), and ranking accuracy (RA).

4.1 Experimental Setup

Datasets As shown in Table 1, a relative sentence readability estimator was trained on a training set of 385k sentence pairs and a validation set of 42k sentence pairs from the parallel corpus for text simplification, Newsela¹ (Xu et al., 2015; Jiang et al., 2020). A set of 43k sentence pairs for evaluation was used to construct a set of sentences for readability ranking.² We also constructed a set of sentences from the CEFR-SP³ (Arase et al., 2022), an English corpus with absolute sentence readability. Note that since the CEFR-SP is not a parallel corpus, it is a set of non-synonymous sentences, unlike Newsela. The CEFR-SP has six levels of readability labels for each of the 17k sentences, and we randomly selected one sentence at each level to obtain a set

¹<https://github.com/chaojiang06/wiki-auto>

²Newsela is a parallel corpus consisting of English news articles manually simplified into four levels. In this experiment, sets of synonymous sentences consisting of different simplifications for the same source sentences were ranked in terms of their readability.

³<https://github.com/yukiar/CEFR-SP>

of sentences. Finally, the set of sentences for evaluation from Newsela totals 4, 478 pairs of five level sentences and one from CEFR-SP totals 165 pairs of six level sentences.

Model For our sentence readability estimator, we employed BERT⁴ (Devlin et al., 2019) as a pre-trained model. We used batch size of 128 sentence pairs, AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 5×10^{-5} . We employed early stopping for fine-tuning with a patience of 3 epochs using a cross-entropy loss in the validation set.

4.2 Baseline models

Baseline unsupervised models We employed two comparative methods of unsupervised sentence readability estimation: define RSRS (Martinc et al., 2021) based on language model scores⁵ and methods based on in-context learning of large language models (LLM). The overall system process follows the upper part of Figure 1, as in our method. In addition, RSRS and LLM estimate the readability of each sentence in the set, as shown in the bottom left-hand corner of Figure 1. In this case, RSRS is a floating, and LLM is an integer.

For the LLM-based method, we used LLaMA⁵ (Touvron et al., 2023) in two settings, 0-shot and 10-shot. We used the prompts in Figure 2 for experiment, which we modified for sentence readability estimation from the prompts used in a previous study (Wang et al., 2024) working on document readability estimation. 0-shot, in which no examples are presented in the prompt (the “example” portion of Figure 2), and 10-shot, in which 10 examples are presented at each readability level. These examples were randomly selected from valid set from Newsela.

Baseline supervised models We employed two types of baselines for supervised sentence readability estimation: the Pointwise method, which imputes sentences, and the Pairwise method, which imputes sentence pairs. The overall system process follows the upper part of Figure 1, as in our method. These are sentence readability estimation models based on masked language models as in the proposed method, but they were trained using absolute

⁴<https://huggingface.co/google-bert/bert-base-uncased>

⁵<https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

System Prompt:

Evaluate the readability of the text using the following eleven levels (reading difficulty):

[score: 2]: Most Easy

[score: 12]: Most Difficult

Based on the provided text examples, assign a readability score to new text and display it in the following format: "[score: X]"

User Input:

Text: {example 1}

[score: 2]

...

Text: {example n}

[score: 12]

New text:

Text: "{}"

Figure 2: Prompt for LLM-based readability estimation.

sentence readability labels in the Newsela corpus,⁶ unlike the proposed method. The pointwise method is a regression model that estimates the readability of input sentences using masked language models and obtains a readability ranking by the readability of each sentence. This baseline estimates the readability of each sentence in the set, as shown in the bottom left-hand corner of Figure 1. In this case, pointwise is a floating.

The pairwise method inputs two sentences as in the proposed method, but unlike the proposed method, it is a regression model that estimates the difference in readability between two given sentences. The pairwise method can provide a binary classification of which sentence is more readable according to whether the output score is positive or negative, resulting in a readability ranking as in the proposed method.

4.3 Results

Readability ranking on Newsela The left side of Table 2 shows the experimental results of readability ranking for a set of synonymous sentences in Newsela⁷. Our method consistently achieved the

⁶There are only a few corpora with sentence readability. So, following previous studies on text simplification (Scarton and Specia, 2018; Nishihara et al., 2019; Yanamoto et al., 2022), the readability of a sentence is defined as the readability of a document containing that sentence. However, but we understand that this is not the best approach.

⁷In this experiment, we use Newsela to enable evaluation, but our method does not use readability labels.

	Supervised	Newsela (Parallel)				CEFR-SP (Non-Parallel)			
		NDCG	ρ	τ	RA	NDCG	ρ	τ	RA
RSRS	-	0.913	0.402	0.341	0.081	0.851	0.082	0.060	0.000
LLM (0-shot)	-	0.888	0.207	0.178	0.041	0.861	0.034	0.027	0.000
Ours	-	0.985	0.865	0.799	0.421	0.958	0.749	0.619	0.048
Pointwise	✓	0.980	0.841	0.769	0.369	0.949	0.661	0.529	0.012
Pairwise	✓	0.986	0.874	0.811	0.438	0.961	0.755	0.621	0.048
LLM (10-shot)	*	0.953	0.644	0.550	0.130	0.967	0.764	0.636	0.073

Table 2: Experimental results of sentence readability estimation. For each setting, unsupervised and supervised, the highest performance is highlighted in bold. * is a few-shot in-context learning.

best performance among the unsupervised methods in the upper rows. The fact that the pairwise method performed better than the pointwise method among the supervised methods suggests that it is important to consider the relationship between sentences for relative readability estimation. Although the supervised pairwise method showed the best performance, our proposed method in an unsupervised manner also achieved comparable performance. Furthermore, the proposed method outperforms the supervised pointwise method and the LLM-based method in the few-shot setting, revealing its effectiveness.

Readability ranking on CEFR-SP The right side of Table 2 shows the experimental results of readability ranking for a set of non-synonymous sentences in CEFR-SP. Similar to the experimental results on Newsela, the proposed method achieved the best performance among the unsupervised methods in the upper rows. However, experiments with non-synonymous sentence sets showed significantly lower RA overall. In comparison with the supervised methods, the proposed method outperforms the pointwise method and is comparable to the pairwise method, again similar to the experimental results on Newsela. In CEFR-SP, the LLM-based method with the few-shot setting outperformed the other supervised methods, achieving the best performance.

4.4 Analysis

We analyse in detail an experiment on readability ranking for synonymous sentence sets in Newsela.

Is relative sentence readability estimation easier the larger the difference in readability between sentence pairs? → **Yes.** To clarify this, we append experiments. Table 3 shows the accuracy

Difference in readability	Accuracy
1	0.759
2	0.886
3	0.954
4	0.990

Table 3: Analysis of the impact of differences in readability of sentence pairs on readability estimation.

results of the readability estimation by splitting the sentence pairs in different levels of readability. The results of this analysis show that as the difference in readability increases (more levels of simplification), the accuracy of relative readability estimation improves. As expected, we can conclude that the larger the difference in readability between sentence pairs, the easier the relative readability estimation is. In specific examples, sentence pairs with small differences, such as “Sub-Saharan Africa has benefited from **high** oil and other commodities **prices**, which have started to decline sharply. → Sub-Saharan Africa has benefited from **high prices for** oil and other commodities, which have started to decline sharply.”, which is a one-level simplification, have a small difference in readability, and it is difficult to determine the latter sentence is simpler. On the other hand, sentence pairs with large differences, such as “Any artifacts linked to an emperor would bring tremendous pride to Mexico. → Finding remains of those leaders would make Mexico proud.”, This is a four-level simplification, has a large difference in readability, and it is easy to determine the latter sentence is simpler. In fact, our method failed to estimate the readability in the top example and succeeded in the bottom one.

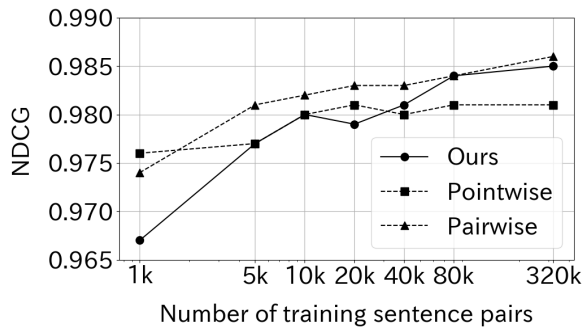


Figure 3: Analysis of the impact of training data size on readability estimation.

How many sentence pairs of parallel corpus for training text simplification make the proposed method effective? → 5k sentences pairs. To clarify this, we append experiments. Figure 3 shows the results of evaluating the quality of the readability ranking (NDCG) while reducing the training data of 385k sentence pairs from 320k to 1k sentence pairs. The results of this analysis show that the text simplification parallel corpus for training our method performs better than the unsupervised sentence readability estimation of RSRS and LLM, NDCG = 0.967 for 1k sentence pairs only. As a text simplification parallel corpus of this scale is available in several languages including Japanese, so the method is promising for the multilingual deployment of sentence readability estimation. And if we can prepare a text simplification parallel corpus consisting of 5k sentence pairs, to reach comparable performance with supervised sentence readability estimation.

5 Conclusion

In this study, we approach unsupervised sentence readability estimation, which does not use absolute sentence readability data. We train a relative sentence readability estimator that predicts which of two given sentences is more simple, using a text simplification parallel corpus, in our method. Then, we derived a readability ranking for the sentence set by pairwise comparisons. Experimental results in English show that our method outperforms previous unsupervised sentence readability estimation for both synonymous and non-synonymous sentence sets, and achieves performance comparable to supervised methods trained with absolute sentence readability.

Limitations

Although the proposed method was designed with multilingual applications in mind, the experiments in this paper are limited only to English. There is no guarantee that performance consistent with this experiment will be achieved in other languages. As mentioned in Section 1, corpora annotated with sentence readability are scarce, and annotating them is very expensive, therefore, it is not easy to actually experiment with non-English languages.

References

- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. [Data-Driven Sentence Simplification: Survey and Benchmark](#). *Computational Linguistics*, 46(1):135–187.
- Yuki Arase, Satoru Uchida, and Tomoyuki Kajiwara. 2022. [CEFR-Based Sentence Difficulty Annotation and Assessment](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6206–6219.
- 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.
- 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.
- Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. [Neural CRF Model for Sentence Alignment in Text Simplification](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960.
- J. Peter Kincaid, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. [Derivation of New Readability Formulas \(Automated Readability Index, Fog Count and Flesch Reading Ease Formula\) for Navy Enlisted Personnel](#). *Technical report, Defense Technical Information Center Document*.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled Weight Decay Regularization](#). In *Proceedings of the Seventh International Conference on Learning Representations*.
- Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. [Supervised and Unsupervised Neural Approaches to Text Readability](#). *Computational Linguistics*, 47(1):141–179.

- Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. [Controllable Text Simplification with Lexical Constraint Loss](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266.
- Carolina Scarton and Lucia Specia. 2018. [Learning Simplifications for Specific Target Audiences](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 712–718.
- Sanja Stajner, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2017. [Automatic Assessment of Absolute Sentence Complexity](#). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4096–4102.
- Kumiko Tanaka-Ishii, Satoshi Tezuka, and Hiroshi Terada. 2010. [Sorting Texts by Readability](#). *Computational Linguistics*, 36(2):203–227.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [LLaMA: Open and Efficient Foundation Language Models](#). *arXiv:2302.13971*.
- Sowmya Vajjala and Ivana Lučić. 2018. [OneStopEnglish Corpus: A New Corpus for Automatic Readability Assessment and Text Simplification](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304.
- Ziyang Wang, Sanwoo Lee, Hsiu-Yuan Huang, and Yunfang Wu. 2024. [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*, pages 280–295.
- 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.
- 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.
- Daiki Yanamoto, Tomoki Ikawa, Tomoyuki Kajiwara, Takashi Ninomiya, Satoru Uchida, and Yuki Arase. 2022. [Controllable Text Simplification with Deep Reinforcement Learning](#). In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 398–404.
- Victoria Yaneva, Constantin Orăsan, Richard Evans, and Omid Rohanian. 2017. [Combining Multiple Corpora for Readability Assessment for People with Cognitive Disabilities](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 121–132.

From End-Users to Co-Designers: Lessons from Teachers

Martina Galletti

Sony Computer Science Laboratories - Paris , France
University of Rome "La Sapienza, Italy
martina.galletti@sony.com

Valeria Cesaroni

University of Perugia, Italy
valeria.cesaroni@dottorandi.unipg.it

Abstract

This study presents a teacher-centred evaluation of an AI-powered reading comprehension tool, developed to support learners with language-based difficulties for English and Italian. Drawing on the Social Acceptance of Technology (SAT) framework, we investigate technical usability and the pedagogical, ethical, and contextual dimensions of AI integration in classrooms. We explore how teachers perceive the platform's alignment with inclusive pedagogies, instructional workflows, and professional values through a mixed-methods approach, including questionnaires and focus groups with educators. Findings revealed a shift from initial curiosity to critical, practice-informed reflection, with trust, transparency, and adaptability emerging as central concerns. The study contributes a replicable evaluation framework and highlights the importance of engaging teachers as co-designers in developing educational technologies.

1 Introduction

As Natural Language Processing (NLP) continues to advance, its applications in education are expanding rapidly—from intelligent tutoring systems to automated writing feedback and reading support (Su et al., 2023; Özer, 2024; Zawacki-Richter et al., 2019). These AI-powered tools promise to transform instruction (Maity and Deroy, 2024), yet a key question remains: How do they perform in real classrooms with real teachers and students? Do they align with the practical realities and pedagogical expectations of educators, ensuring both usability and instructional relevance? (Cesaroni et al., 2024) Many systems are developed in controlled settings with limited educator input (Celik et al., 2022; Cukurova and Luckin, 2018; Luckin and Cukurova, 2019), often overlooking pedagogical realities and learning science principles (Luckin and Cukurova, 2019; Cesaroni et al., 2025).

Addressing this disconnect requires greater attention to the roles educators play, not just as passive users, but as active contributors throughout the AI development cycle. Indeed, teachers have already played various roles in educational AI research (Celik et al., 2022). They served as models for AI training through classroom data (Su et al., 2014; Kelly et al., 2018), shared professional development information to improve predictive systems (Alzahrani and Alzahrani, 2025; Yoo and Rho, 2020), and provided student data to support AI-driven interventions (Bonneton-Botté et al., 2020; Nikiforos et al., 2020). They have also validated AI outputs by grading work and defining evaluation criteria (Huang et al., 2010; Yuan et al., 2020), influenced pedagogical alignment through instructional material selection (Dalvean and Enkhbayar, 2018; Fitzgerald et al., 2015), and in some cases, offered technical feedback on system design (Burstein et al., 2004). Despite these contributions, their role as evaluators who shape AI integration in classroom contexts remains largely underexamined.

Building on this foundation, the paper presents an evaluation framework of an AI-powered reading comprehension interface using a framework that places educators at the centre of AI integration. Although applied to a single interface in this study, the framework is generalizable to the evaluation of AI technologies across diverse educational contexts. This framework draws on the SAT model to examine the pedagogical, ethical, and practical dimensions of AI adoption in education. Through a mixed-methods approach involving questionnaires and focus groups, we not only assess how teachers perceive the system but also explore how their insights can shape more effective, inclusive, and ethically grounded AI implementation in real classroom settings.

Our findings highlight the value of participatory design, showing that teachers act as co-designers and evaluators, not just users. Their acceptance of

AI tools relies on alignment with pedagogical values, transparency, and autonomy. While they saw promise in promoting inclusion and differentiated instruction, they also pointed to needed improvements in clarity, layout, and customisation. These insights call for ethically grounded, teacher-centred approaches and further research through long-term classroom use, evaluation and broader educator involvement.

2 Background

2.1 Existing Reading Comprehension Interfaces

Reading comprehension interfaces aim to support users in understanding and engaging with complex textual material. Unlike general reading tools, these systems are designed to go beyond passive reading by incorporating interactive features such as question answering, summarisation, sentence simplification, and semantic annotations. Existing tools focus on general users and surface-level comprehension, lacking therapeutic intent, multilingual support, and personalisation.

One of the most notable efforts in this domain is the Semantic Reader Project (Lo et al., 2023), which augments scientific documents with context-aware explanations, definitions, and citation-level summaries to help readers quickly identify core ideas. Similarly, systems like SciReader (Head et al., 2021) employ semantic highlighting, definitions on hover, and automatic summarisation to assist users, particularly researchers, in navigating dense academic material. However, these tools are not tailored to students with reading comprehension deficits or learning disorders.

Another promising direction involves gaze-driven sentence simplification interfaces, such as the work of Higasa et al. (2023), which are particularly relevant for language learners or readers with cognitive impairments. These systems use real-time eye-tracking data to detect reading difficulty and apply NLP techniques to simplify complex sentences. However, while useful as assistive technologies, they do not provide structured activities aimed at rehabilitating underlying comprehension deficits.

Complementing these assistive tools are educational systems like 3D Readers (3dR) and CACSR (Kim et al., 2006), which take a more interactive and instructional approach to enhancing reading comprehension. 3D Readers allow

users to engage with texts through either verbal strategies (such as question generation) or visual strategies (like manipulating images), with immediate feedback provided to support learning (Johnson-Glenberg, 2007). Similarly, CACSR offers personalised instruction using techniques like visual imagery, graphic organisers, mnemonics, self-questioning, and summarization (Stetter and Hughes, 2011), also incorporating real-time feedback to support continuous assessment (Kim et al., 2006). Despite their effectiveness in educational settings, these systems are not designed with therapeutic goals in mind.

Moreover, existing systems are designed for English-language users. There appears to be only one known system available in Italian that supports integrated telerehabilitation: RIDInet¹. The platform offers activity modules like the Cloze Application, which trains reading comprehension through multiple-choice tasks. However, RIDInet does not offer targeted exercises for developing word-level literal understanding, nor does it support the integration of prior knowledge with new textual input.

2.2 Assessment of AI-Powered Educational Technologies

Evaluating AI-powered educational tools poses a methodological challenge due to the lack of frameworks integrating pedagogical, psychological, and social dimensions of technology adoption. Existing models, such as the Technology Acceptance Model (TAM) (Davis, 1989) and the Unified Theory of Acceptance and Use of Technology (UTAUT) (Venkatesh et al., 2003) offer robust tools for analysing perceptions of usefulness, usability, and behavioural intention. However, these frameworks emphasise generic constructs (e.g., efficiency, ease of use), but often neglect education-specific factors such as alignment with instructional goals, teacher–student dynamics, and pedagogical adaptability.

The Technological Pedagogical Content Knowledge (TPACK) framework (Koehler and Mishra, 2009) addresses the integration of technology into pedagogy, but primarily with a formative intent. It delineates the competencies required to design learning experiences that effectively combine technological tools with pedagogical strategies and disciplinary knowledge. However, it lacks evaluative tools for real-world adoption, omitting con-

¹<https://www.anastasis.it/ridinet/>

cerns like ethical implications, institutional fit, and teacher autonomy. For instance, a teacher may possess TPACK proficiency in using an NLP tool yet refrain from adopting it due to ethical concerns (e.g., algorithmic bias) or practical constraints (e.g., misalignment with classroom workflows) — factors that lie outside the scope of TPACK.

To address this gap, this study adopts a mixed-methods approach guided by the SAT framework (Occhipinti et al., 2023) to explore the use of AI tools in educational settings. Unlike traditional models that primarily assess individual user experience or usability, the SAT framework views technology as part of a broader socio-technical system. By focusing on four interrelated dimensions (User Experience, Value Impact, and Trust), SAT enables an assessment that extends beyond subjective usability to encompass ethical, cultural, and contextual factors. In the context of schools, the study uses SAT to design questionnaires and focus groups that examine how a software for the teaching of reading comprehension aligns with pedagogical values, affects teacher relationships and institutional structures, and impacts trust. Special emphasis is placed on the Value Impact and Trust dimensions, which help uncover educators’ perspectives on issues like inclusion, transparency, autonomy, and coherence with teaching practices.

3 System Description

In this paper, we present a novel and enhanced version of ARTIS (Galletti et al., 2023, 2024). ARTIS is a web-based educational tool designed to support reading comprehension for primary school students, with a particular focus on learners with reading difficulties or language-based learning disorders up to 11 years old (Galletti et al., 2023). The system integrates a multimodal approach to text comprehension by combining visual, auditory, and interactive components. The interface supports multilingual content (Italian and English), making it adaptable for bilingual contexts or second-language learners. While some of the assistive features were previously introduced in Galletti et al. (2024), this version introduces new rehabilitative features, an enhanced administrative dashboard, as well as updated design and graphics.

The design of the platform’s features is grounded in the psycholinguistic model of reading comprehension proposed by Kintsch and van Dijk (Kintsch and Van Dijk, 1978; Galletti et al., 2023). This

model outlines three levels of text comprehension. First, there is surface representation, which involves recognizing words and grammar (i.e. lexical and morphosyntactic understanding). Second, there is propositional representation, where readers connect ideas into meaningful sequences and structures. Finally, there is the mental model construction, where readers combine what the text says with their background knowledge.

Following Kintsch and Van Dijk’s model, our interface includes different modules and exercises targeting different comprehension levels: lexical understanding, propositional structuring, and mental model integration, each progressively supporting deeper text processing. In the next subsection, we describe each module and the algorithms behind its core functionalities².

3.1 Assistive Features

Upon logging into the platform, students can access a digital library interface that displays a collection of illustrated literary and informational texts. Each title is visually represented with a stylised image and a short textual excerpt to support browsing and engagement. Once a text is selected, the reading interface presents the full passage along with assistive features such as read-aloud audio, using the Google text-to-speech API, synchronized text highlighting, and pace controls (e.g., play, pause, and speed adjustment). These supports are designed to aid comprehension while still requiring the child to engage actively with the text.

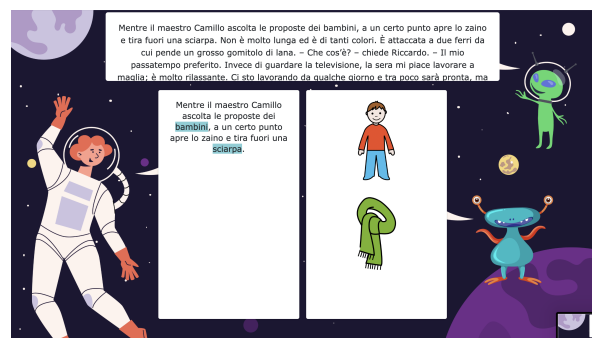


Figure 1: Keywords like “*bambini*” (“children”) and “*sciarpa*” (“scarf”) are highlighted and paired with pictograms to support understanding.

In a subsequent step, the text is presented sentence by sentence thanks to the spaCy Sentencizer³ and key terms within the passage are visually highlighted and linked to pictograms that

²A recorded demonstration of our proof of concept is available at this [link](#).

³<https://spacy.io/api/sentencizer>

illustrate their meaning, as in Figure 1. Secondly, keywords are extracted using a fine-tuned version of Keybert (Grootendorst, 2020). To ensure accuracy and prevent misleading outputs, the extracted keywords were manually reviewed by speech and language therapists as described in Galletti et al. (2023). Once the keywords were extracted from the sentences, after lemmatisation, we used the Arasaac API⁴ to link them to pictograms.

In a third step, unfamiliar or specific terms are also supported with definitions and example sentences drawn directly from the source passage, as in Figure 2. These terms are selected either manually by the operator or automatically extracted as detailed in Galletti et al. (2023). Users can interact with the words to hear their definitions, view pictograms illustrating their meanings, and see them embedded in the text, supporting multi-sensory learning and strengthening decoding skills. For definitions, we used *gpt-3.5-turbo-0125*⁵.

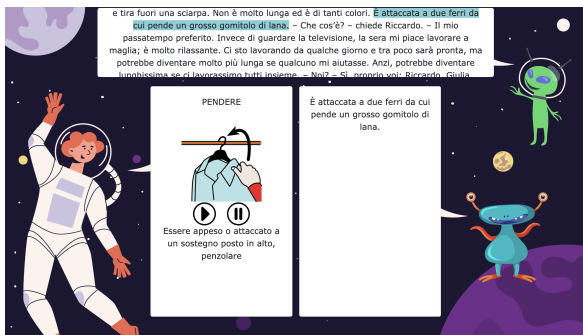


Figure 2: Vocabulary’s support a not common Italian verb “*pendere*” (“to hang” or “to dangle”). The left panel shows the pictogram associated with it, a definition, and audio playback buttons to support its comprehension.

3.2 Rehabilitative Features

A variety of comprehension and language-focused exercises are included to deepen semantic processing and support inference-making skills. These activities comprise: (1) “*Leggi e rispondi*” (“Read and Respond”), where students answer comprehension questions generated by *gpt-3.5-turbo-0125* and manually validated by speech and language therapists; and (2) “*Trova le parole chiave*” (“Find the Keywords”), which engages learners in identifying key terms within the text. Keywords are generated using a fine-tuned version of KeyBERT, as

⁴<https://arasaac.org/>

⁵<https://platform.openai.com/docs/models/gpt-3-5-turbo>

described in Galletti et al. (2023); and (3) “*Trova la rete semantica*” (“Build the Semantic Network”), which prompts students to connect words with semantically related concepts, as illustrated in Figure 3. Both the related terms and distractors for this task were generated using *gpt-3.5-turbo-0125*. To generate related and unrelated words, we used two prompts: one asked for a list of synonyms with definitions, and the other for non-synonyms that are semantically unrelated, both formatted as JSON arrays with appropriate keys.

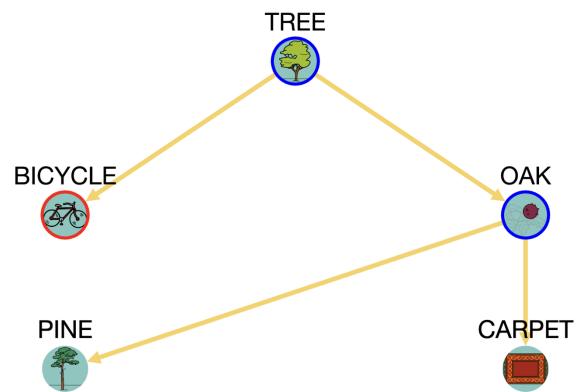


Figure 3: Example of a semantic network exercise. The user starts from the word “TREE” and must choose which of the two presented options is semantically related. If the correct option is selected, the network expands and presents two new options.

3.3 Administrative Dashboard

Finally, the interface includes an administrative dashboard that gives therapists full control over all aspects of the content generated by the AI algorithms. This dashboard allows for the management of texts, such as editing existing content, inserting new texts, or regenerating AI-related components, as well as the organisation and customisation of exercises. When a new text is added, the dashboard displays a preview of each AI-generated component—such as keywords, pictograms, sentences, and questions—for validation. This enables the educator to verify or adjust these associations before the material is presented to the learner. Additionally, the dashboard supports the enrollment of students and the assignment of personalised texts, avatars and exercises, helping to tailor the learning experience to each user. It also collects the data and makes it available for downloading to the therapist.

Dimension	Description
Pedagogical Appropriateness	Evaluates alignment with inclusive and AAC (Augmentative and Alternative Communication)-based pedagogy, focusing on scaffolding, shared meaning-making, and student autonomy.
Inclusive Potential	Assesses support for diverse learners and compatibility with Universal Design for Learning (UDL) and cooperative learning strategies.
Teacher Readiness	Explores how hands-on experience influenced openness to AI, highlighting competence gaps and training needs.
Trust	Investigates confidence in AI-driven features and the balance between automation and teacher control, including transparency and ethical oversight.
Expectation Shift	Compares pre- and post-use attitudes to identify shifts in teachers' perceptions of AI in education.

Table 1: The five key dimensions which guided the focus group.

4 Methods

The evaluation framework is a two-phase approach: a questionnaire which assessed teachers' general attitudes and readiness toward AI, followed by a focus group conducted after hands-on interaction with the platform. This structure allowed us to compare abstract views of AI with teachers' hands-on experiences, highlighting how their perceptions align with pedagogical values, ethical concerns, and practical adoption barriers. This section outlines the questionnaire and focus group design; results are presented in Section 5 followed by their discussion in Section 6.

4.1 Questionnaire's Design

The questionnaire explored teachers' knowledge, perceptions, and attitudes toward digital and AI technologies in education. Its structure follows the four dimensions of the SAT framework: user experience, social disruptiveness, value alignment, and trust. As part of our contribution, the questionnaire is openly available to the community at this [link](#), while the detailed results are available [here](#).

Following an initial section collecting teacher details such as years of experience and subject area, Section 2 explored teachers' general approach to technology use. It included items on comfort with technology in personal and professional contexts, habits for staying updated on tools, and frequency of using digital resources for planning. Participants also rated the importance of technology in teaching and their interest in new tools. The section ended with a multiple-choice item on perceived barriers to tech integration, based on established research (Ertmer et al., 2012).

Section 3 focused on teachers' awareness and use of digital tools designed to support reading comprehension, drawing on research into the educational technology ecosystem (Tondeur et al., 2017). Participants first have to identify any relevant software or platforms they know, such as simplified reading tools, text-to-speech apps, or concept map-

ping software. They then need to indicate which they had used in teaching or planning, and report any reading comprehension technologies available at their schools.

Section 4 explores teachers' perceptions of AI in education, assessing their knowledge, expectations, trust, ethical concerns, and professional agency—defined as the capacity to shape one's practice within institutional and technological constraints (Biesta et al., 2015; Toom et al., 2015). A mixed-format design with Likert-scale and open-ended items enables a mixed-methods analysis, aligning with best practices in educational technology research (Ponce and Pagán-Maldonado, 2015). This last section is divided into three subsection focusing respectively on (A) *Trust and Risk/Benefit Perception* - using items adapted from the Propensity to Trust in AI scale (Mcknight et al., 2011), (B) *Ethical Awareness*, drew on critical AI literacy frameworks (Veldhuis et al., 2024) and (C) *Teacher Agency and Involvement*, with items informed by the TPACK framework (Mishra and Koehler, 2006) and research on teacher agency (Leijen et al., 2024). This subsection examines the perception of students' interest in AI, teachers' views on the importance of ethical and pedagogical training, and expectations for future integration and involvement in AI-related decisions.

4.2 Focus Group's Design

The focus group was intentionally designed to capture authentic, practice-informed insights from educators by combining experiential use of the platform with structured group reflection. Following the initial questionnaire, participating teachers engaged in a one-hour, hands-on session with the ARTIS platform. To preserve the ecological validity of the study, no prior exposure or formal training was provided. Instead, participants received minimal onboarding and quick-start instructions, allowing for natural, intuitive engagement with the interface.

The exploratory session encouraged teachers to

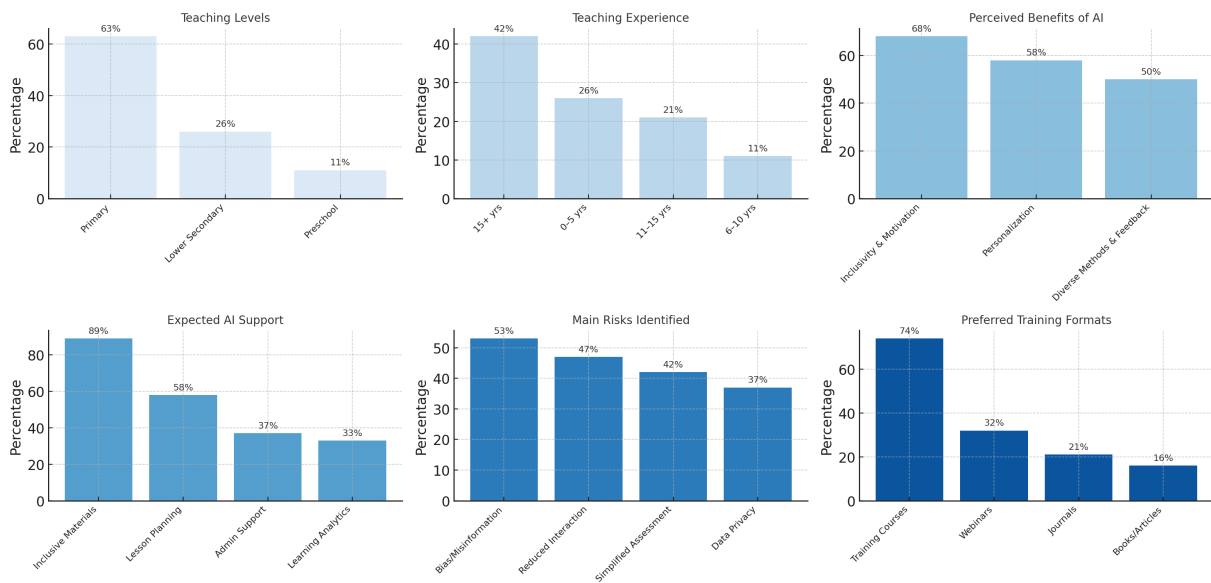


Figure 4: Summary of questionnaire responses from 19 teachers on AI in education, including teaching demographics, perceived benefits and risks, expected support, and preferred training formats.

freely navigate the platform, selecting from a variety of texts. Participants were asked to engage with materials spanning multiple educational levels to ensure a range of instructional contexts were represented. Additionally, a subset of four teachers was invited to interact with English-language texts, enabling the evaluation of bilingual and second-language accessibility features.

Immediately after the interaction phase, a structured focus group gathered in-depth reflections on five key dimensions of AI acceptance and pedagogical fit: (I) *Pedagogical Appropriateness*, (II) *Inclusive Potential*, (III) *Teacher Readiness*, (IV) *Trust*, and (V) *Reconfiguration of Expectations*—based on sociotechnical, ethical, and inclusive education frameworks and guided the analysis of teacher responses. Precise details on this key dimension can be found in Table 1.

5 Results

We recruited 19 teachers from the *Istituto Comprensivo di Narni Scalo (Italy)* to participate in this study. The sample included 12 primary school teachers, 5 lower secondary teachers, and 2 preschool educators. Although the ARTIS platform is primarily designed for literacy development and may have limited direct applicability in early childhood education, preschool teachers were intentionally included to examine how attitudes toward AI integration vary across educational levels. This inclusive approach captured diverse perspectives

and enabled comparison of teachers' readiness for AI across contexts.

5.1 Questionnaire Results

When asked about digital tools used to support reading comprehension, teachers cited a mix of general productivity platforms, such as [Google Workspace](#) and [Canva](#), alongside more specialized tools like [Genially](#), [Popplet](#), and [Arasaac](#) for augmentative and alternative communication. A few teachers had explored newer AI-powered tools such as [NotebookLM](#) and [Napkin.AI](#). Participants were optimistic about AI's educational benefits, linking it to inclusivity, student motivation, personalized learning, diverse instructional formats, and instant feedback. Teachers saw AI as useful for creating inclusive materials, supporting planning and content design, reducing administrative tasks, and improving the use of learning analytics to guide instruction.

At the same time, teachers were mindful of the risks associated with AI in education. Concerns included the possibility of algorithmic bias, misinformation, and diminished teacher autonomy. Others raised ethical issues such as the reduction of student-teacher interaction, oversimplified forms of assessment, and the potential misuse of student data. In terms of professional development, participants showed a clear preference for structured training opportunities focused on the ethical and social dimensions of AI. Webinars, journals, and

other professional resources were also seen as helpful, though less commonly preferred. Overall, the findings suggested a cautiously optimistic attitude among teachers. While they see promise in the pedagogical affordances of AI, especially about inclusion and personalization, they also emphasize the need for thoughtful implementation, transparency, and support through well-designed training. Aggregated results are shown in Figure 4 and detailed results are available [here](#).

5.2 Focus group results

The following paragraphs present findings from the focus group discussions, organized according to the five evaluative dimensions illustrated in Table 1.

Pedagogical Appropriateness Participants raised thoughtful concerns about the platform's alignment with inclusive pedagogical practices. A key point of critique centered on the sequencing of support features: currently, visual and linguistic scaffolds, such as keywords and pictograms, are presented before students actively engage with the text through the exercise. While well-intentioned, this design was perceived by many as potentially limiting student autonomy and interpretive effort. Several educators proposed reversing this order, suggesting that scaffolds introduced during or after initial engagement would better support active meaning-making.

Inclusive Potential Teachers generally recognized the platform's value in supporting differentiated instruction, especially for students with language-based or cognitive challenges. However, concerns were raised about the semantic precision and clarity of the pictograms, as well as the overall visual layout, both of which were seen as crucial to accessibility. Importantly, educators emphasized that the platform should not replace teacher-student interaction but instead enhance it, particularly through collaborative practices like co-selecting keywords and interpreting texts.

Teacher Readiness While initial questionnaire responses reflected a generally positive orientation toward AI in education, the post-use discussions revealed more grounded, experience-based perspectives. Teachers emphasized the importance of training that goes beyond technical operation to include pedagogical integration. They acknowledged the platform's potential to support differentiated learning and streamline resources, but stressed that its

success would depend on its adaptability to real classroom contexts and its alignment with established instructional workflows.

Trust Trust in the platform emerged as closely tied to the degree of teacher agency and system transparency. However, this did not translate into a blanket rejection of the technology. Rather, educators identified specific areas for improvement, calling for enhanced user control and clearer communication about how AI-driven choices are made.

Expectations Shift Educators moved from abstract curiosity and cautious optimism to a more critical, practice-informed perspective. Their experiences prompted a clear set of priorities for the future development of AI in education: (1) Flexibility over rigidity – AI tools must be adaptable to diverse classroom contexts; (2) Transparency over opacity – teachers need to understand and shape how AI-driven decisions are made; (3) Support over substitution – technology should amplify, not replace, human interaction and pedagogical creativity. While initial enthusiasm was tempered by practical limitations, participants remained confident in the potential of AI-supported learning environments—particularly when such tools are designed to complement teacher expertise and foster meaningful student engagement. Importantly, the findings underscore the value of involving educators not merely as users, but as co-designers and evaluators in the development process.

6 Discussion

This study contributes to a growing body of research on the integration of AI in education by offering a practice-informed, teacher-centered perspective grounded in the SAT framework (Occhipinti et al., 2023). Our findings show how teachers' acceptance of AI tools is not static nor solely based on usability, but shaped dynamically through hands-on engagement, educational values, and pedagogical alignment. Methodologically, this study aligns with recent calls for participatory and iterative approaches to AI design in education (Luckin and Cukurova, 2019; Mouta et al., 2024).

Rather than viewing acceptance as a fixed variable to be assessed retrospectively (Celik et al., 2022), our approach positions teachers as formative agents, whose experiences, critiques, and creativity are integral to the ethical and effective development of technology. In line with Zawacki-Richter

et al. (2019), who underline the scarcity of qualitative studies that capture educators' voices in AI research, our work emphasizes the importance of interpretive approaches that explore how teachers make sense of AI tools in concrete pedagogical settings. By combining questionnaires with in-depth focus groups, we were able to reveal not only general trends in acceptance but also the nuanced ways in which teachers negotiated the role of AI in their practice. Specifically, participants moved from general skepticism to targeted suggestions, such as reversing scaffold sequencing or refining pictogram clarity, demonstrating a shift from rejection to co-design.

While 89% of teachers viewed AI as potentially beneficial for inclusive education (questionnaire), the focus group exposed significant caveats. Teachers emphasized that inclusion cannot be achieved through technical affordances alone but requires alignment with pedagogical routines and accessibility standards. These findings echo critiques of UDL frameworks when implemented in a top-down, compliance-oriented manner (Edyburn, 2010). Instead, participants advocated for adaptive interfaces, customizable visuals, and collaborative practices, such as co-selection of keywords, that preserve student-teacher interaction and interpretive autonomy. This underscores the need to reconceptualize inclusivity as a dynamic co-construction rather than a static feature.

Our findings also nuance assumptions in the literature about AI adoption motives. Their insistence on retaining control over scaffolding and keyword selection reflects a broader commitment to maintaining instructional intentionality. Similarly, the shift from interest in formal AI training (74%) to a focus on pedagogical and ethical guidance suggests that professional development should go beyond technical skills to include critical and ethical perspectives (Perrotta and Selwyn, 2020). By engaging teachers not as end-users but as evaluators and co-evaluators, this study contributes a replicable model for socio-technical evaluation and advances the debate on how educational technologies can be made aligned with the realities of classroom practice.

Finally, these insights are also shaped by the cultural and institutional context. The participating teachers, embedded within the Italian education system, approached AI-mediated feedback through pedagogical norms distinct from those observed in studies conducted in other countries. For ex-

ample, while Anglo-American frameworks often emphasize data-driven personalization and performance metrics (Shum and Luckin, 2019; Selwyn, 2019), Italian educators tended to prioritize dialogic, relational approaches to learning and a strong emphasis on formative assessment as a collective rather than individualistic practice (Moretti et al., 2015; Pastore, 2020). Teachers highlighted the importance of maintaining pedagogical intentionality and student-teacher interaction, reflecting a broader educational tradition in Italy that values interpretive autonomy and humanistic principles (Viteritti, 2009). These values shaped how teachers perceived AI tools—not simply as assistive technologies, but as agents that must harmonize with existing curricular structures, ethical responsibilities, and institutional logics. This cultural lens helps explain why some technological affordances, such as automated scaffolding or visual simplifications, were met with ambivalence unless they could be flexibly adapted to local pedagogical aims. Cross-cultural comparisons are thus essential to avoid universalist assumptions in AI design and to ensure that integration strategies remain sensitive to educational diversity (Zhang, 2025).

6.1 Actionable steps

The findings confirm that when engaged early in the development process, teacher expertise plays a pivotal role in surfacing abstract concerns and translating them into actionable design feedback (UNESCO, 2021). Rather than perceiving resistance to AI as rooted in negative attitudes, the study highlights the value of participatory engagement, where teachers act as co-designers. In particular, initial concerns centred around algorithmic opacity and perceived lack of control (Mcknight et al., 2011), mirroring broader critiques of AI as black-box systems that conceal decision-making logic (Burrell, 2016). Participants consistently called for increased transparency, interpretability, and human oversight—features that contribute to what is often termed “calibrated trust” (Zhang et al., 2020), where users remain critically engaged while feeling empowered to understand and shape system outcomes.

The next phase of interface development started to operationalize the teachers' inputs into concrete design interventions. These included (1) co-designed interface modifications with customizable elements (e.g., reversible scaffold sequencing, teacher-defined, configurable visual aids), (2) im-

plementing adjustable transparency layers (e.g., explainable feedback rationales), (3) develop teacher-facing toggles for AI assistance, allowing instructors to choose when and how AI contributes during a session (e.g., real-time suggestion, post-activity reflection), (4) include a cultural/contextual alignment layer in system design documentation: capture assumptions embedded in educational norms (e.g., Italian vs. others), design localized variants where necessary and plan for comparative studies across national systems to test transferability.

7 Conclusions & Future Work

This study is among the first to systematically evaluate the ethical and pedagogical acceptability of AI in real educational contexts using a framework explicitly oriented toward sociotechnical reflection. It highlights the importance of involving teachers not just as end-users, but as active co-designers and evaluators—positioning them as key contributors in shaping how AI tools are developed and integrated into educational settings. Rather than viewing acceptance as a fixed variable to be assessed retrospectively (Celik et al., 2022), our approach positions teachers as formative agents, whose experiences, critiques, and creativity are integral to the ethical and effective development of technology.

Using the SAT framework, we captured nuanced perceptions of an AI-supported reading platform, revealing that teachers' acceptance depends on more than functionality: it is conditional on alignment with pedagogical values, transparency, and support for professional autonomy. In sum, the question is not only whether AI works, but whether it works with and for teachers, in alignment with the values and practices that define education. The shift from abstract optimism to context-sensitive critique underscores the importance of participatory, ethically grounded approaches to AI design.

Future work should also explore long-term classroom use to track evolving practices, involve a more diverse sample of educators, in a second school, and examine how training and co-design processes influence ethical and pedagogical alignment of AI technologies, further validating the SAT model for educational settings. Moreover, insights from the current evaluation framework will inform the next development cycle of the ARTIS interface, ensuring that future iterations are responsive to teacher feedback.

Limitations

While the study offers valuable insights and contributes meaningfully to the design of educational AI tools, certain limitations also highlight important avenues for future exploration. The teacher sample, though rich in contextual relevance, was relatively small and specific, which may affect the broader applicability of the findings. However, this focused scope allowed for in-depth engagement and formative feedback that can directly inform future iterations. Participants interacted with a prototype version of ARTIS, and some of their observations likely reflect temporary interface or usability elements rather than underlying structural challenges, providing useful direction for refinement.

Moreover, the study's temporal scope was limited to a single session, offering a snapshot of initial impressions rather than longitudinal insights. Nonetheless, this approach effectively captured early responses and surfaced key priorities for longer-term implementation. Similarly, although emerging needs for teacher training were identified during the focus group, the study did not incorporate formal training programs. This opens a promising path for future research to investigate how targeted support mechanisms influence adoption and pedagogical integration over time.

An additional consideration relates to the use of the SAT framework. While SAT served as a valuable and critically-informed structure for guiding the study, it remains a relatively new model, especially in educational contexts. In this study, we adapted a combination of existing scales and custom-developed items to reflect the SAT's four dimensions. While this tailoring ensured contextual relevance, it may reduce replication and comparability across studies. Advancing this work will require the development and validation of standardized SAT-based instruments to foster wider methodological consistency.

Despite these limitations, the study yields important design implications. Educational AI tools should enable flexibility and personalization, promote transparency to build trust, and support—rather than supplant—educators' pedagogical creativity.

References

- 3D Readers Software. <https://www.sbir.gov/node/334161>. Accessed: 10/04/2025.
- Abdulaziz Alzahrani and Amal Alzahrani. 2025. Comprendiendo la adopción de chatgpt en universidades: el impacto del tpack y utaut2 en los docentes understanding chatgpt adoption in universities: the impact of faculty tpack and utaut2. *RIED-Revista Iberoamericana de Educación a Distancia*, 28(1).
- Gert Biesta, Mark Priestley, and Sarah Robinson. 2015. The role of beliefs in teacher agency. *Teachers and teaching*, 21(6):624–640.
- N. Bonneton-Botté, G. Beucher, and F. Ollivier. 2020. Can tablet apps support the learning of handwriting? an investigation of learning outcomes in kindergarten classroom. *Computers & Education*, 144:103706.
- Jenna Burrell. 2016. How the machine ‘thinks’: Understanding opacity in machine learning algorithms. *Big data & society*, 3(1):2053951715622512.
- J. Burstein, M. Chodorow, and C. Leacock. 2004. Automated essay evaluation: The criterion online writing service. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 23–26. Association for Computational Linguistics.
- Ismail Celik, Muhterem Dindar, Hanni Muukkonen, and Sanna Järvelä. 2022. The promises and challenges of artificial intelligence for teachers: A systematic review of research. *TechTrends*, 66(4):616–630.
- Valeria Cesaroni, Martina Galletti, Eleonora Pasqua, Daniele Nardi, and 1 others. 2024. Towards trustworthy ai in inclusive education: A co-creation approach rooted in ecological frameworks. In *Ital-IA 2024: 4th National Conference on Artificial Intelligence, organized by CINI*.
- Valeria Cesaroni, Eleonora Pasqua, Piercosma Bisconti, and Martina Galletti. 2025. A participatory strategy for ai ethics in education and rehabilitation grounded in the capability approach. *arXiv preprint arXiv:2505.15466*.
- Mutlu Cukurova and Rose Luckin. 2018. Measuring the impact of emerging technologies in education: A pragmatic approach.
- Michael Dalvean and Galbadrakh Enkhbayar. 2018. Assessing the readability of fiction: A corpus analysis and readability ranking of 200 english fiction texts. *Linguistic Research*, 35:137–170.
- Fred D Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340.
- Dave L Edyburn. 2010. Would you recognize universal design for learning if you saw it? ten propositions for new directions for the second decade of udl. *Learning Disability Quarterly*, 33(1):33–41.
- Peggy A Ertmer, Anne T Ottenbreit-Leftwich, Olgun Sadik, Emine Sendurur, and Polat Sendurur. 2012. Teacher beliefs and technology integration practices: A critical relationship. *Computers & education*, 59(2):423–435.
- Jill Fitzgerald, Jeff Elmore, Heather Koons, Elfrieda H Hiebert, Kimberly Bowen, Eleanor E Sanford-Moore, and A Jackson Stenner. 2015. Important text characteristics for early-grades text complexity. *Journal of Educational Psychology*, 107(1):4.
- Martina Galletti, Eleonora Pasqua, Francesca Bianchi, Manuela Calanca, Francesca Padovani, Daniele Nardi, and Donatella Tomaiuolo. 2023. A reading comprehension interface for students with learning disorders. In *Companion Publication of the 25th International Conference on Multimodal Interaction*, pages 282–287.
- Martina Galletti, Eleonora Pasqua, Manuela Calanca, Caterina Marchesi, Donatella Tomaiuolo, Daniele Nardi, and 1 others. 2024. Artis: a digital interface to promote the rehabilitation of text comprehension difficulties through artificial intelligence. In *Ital-IA 2024: 4th National Conference on Artificial Intelligence, organized by CINI, May 29-30, 2024, Naples, Italy*.
- Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.
- Andrew Head, Kyle Lo, Waleed Ammar, and 1 others. 2021. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.
- Taichi Higasa, Keitaro Tanaka, Qi Feng, and Shigeo Morishima. 2023. Gaze-driven sentence simplification for language learners: Enhancing comprehension and readability. *arXiv preprint arXiv:2310.00355*.
- R. H. Huang, J. M. Spector, and J. F. Yang. 2010. Ict literacy and the development of digital learning resources. *Educational Technology Research and Development*, 58(2):191–204.
- Mina C Johnson-Glenberg. 2007. *Web-based reading comprehension instruction: Three studies of 3D-readers*. Erlbaum.
- 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.
- Ae-Hwa Kim, Sharon Vaughn, Janette K Klingner, Althea L Woodruff, Colleen Klein Reutebuch, and Kamiar Kouzekanani. 2006. Improving the reading comprehension of middle school students with disabilities through computer-assisted collaborative strategic reading. *Remedial and special education*, 27(4):235–249.

- Walter Kintsch and Teun A Van Dijk. 1978. Toward a model of text comprehension and production. *Psychological review*, 85(5):363.
- Matthew Koehler and Punya Mishra. 2009. What is technological pedagogical content knowledge (tpack)? *Contemporary issues in technology and teacher education*, 9(1):60–70.
- Äli Leijen, Margus Pedaste, and Aleksandar Baucal. 2024. A new psychometrically validated questionnaire for assessing teacher agency in eight dimensions across pre-service and in-service teachers. In *Frontiers in Education*, volume 9, page 1336401. Frontiers Media SA.
- Kyle Lo, Joseph Chee Chang, Andrew Head, and 1 others. 2023. The semantic reader project: Augmenting scholarly documents through ai-powered interactive reading interfaces. *arXiv preprint arXiv:2303.14334*.
- Rosemary Luckin and Mutlu Cukurova. 2019. Designing educational technologies in the age of ai: A learning sciences-driven approach. *British Journal of Educational Technology*, 50(6):2824–2838.
- Subhankar Maity and Aniket Deroy. 2024. Generative ai and its impact on personalized intelligent tutoring systems. *arXiv preprint arXiv:2410.10650*.
- D Harrison Mcknight, Michelle Carter, Jason Bennett Thatcher, and Paul F Clay. 2011. Trust in a specific technology: An investigation of its components and measures. *ACM Transactions on management information systems (TMIS)*, 2(2):1–25.
- Punya Mishra and Matthew J Koehler. 2006. Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record*, 108(6):1017–1054.
- Giovanni Moretti, Arianna Giuliani, and ARIANNA LODOVICA Morini. 2015. Flexible and dialogic instructional strategies and formative feedback: an observational research on the efficacy of assessment practices in italian high schools. In *ICERI2015 Proceedings*, pages 8229–8236. IATED.
- Ana Mouta, Ana María Pinto-Llorente, and Eva María Torrecilla-Sánchez. 2024. Uncovering blind spots in education ethics: Insights from a systematic literature review on artificial intelligence in education. *International Journal of Artificial Intelligence in Education*, 34(3):1166–1205.
- S. Nikiforos, S. Tzanavaris, and K. L. Kermanidis. 2020. [Automatic detection of learners’ aggressive behavior in a virtual learning community using machine learning techniques](#). *Education and Information Technologies*, 25(4):3293–3313.
- Carmela Occhipinti, Antonio Carnevale, Luigi Briguglio, Andrea Iannone, and Piercosma Bisconti. 2023. Sat: a methodology to assess the social acceptance of innovative ai-based technologies. *Journal of Information, Communication and Ethics in Society*, 21(1):94–111.
- Mahmut Özer. 2024. Potential benefits and risks of artificial intelligence in education. *Bartın University Journal of Faculty of Education*, 13(2):232–244.
- Serafina Pastore. 2020. How do italian teacher trainees conceive assessment?. *International Journal of Instruction*, 13(4):215–230.
- Carlo Perrotta and Neil Selwyn. 2020. Deep learning goes to school: Toward a relational understanding of ai in education. *Learning, Media and Technology*, 45(3):251–269.
- Omar A Ponce and Nellie Pagán-Maldonado. 2015. Mixed methods research in education: Capturing the complexity of the profession. *International journal of educational excellence*, 1(1):111–135.
- Neil Selwyn. 2019. *Should robots replace teachers?: AI and the future of education*. John Wiley & Sons.
- SJ Buckingham Shum and Rosemary Luckin. 2019. Learning analytics and ai: Politics, pedagogy and practices. *British journal of educational technology*, 50(6):2785–2793.
- Maria Earman Stetter and Marie Tejero Hughes. 2011. Computer assisted instruction to promote comprehension in students with learning disabilities. *International Journal of Special Education*, 26(1):88–100.
- Jiahong Su, Davy Tsz Kit Ng, and Samuel Kai Wah Chu. 2023. Artificial intelligence (ai) literacy in early childhood education: The challenges and opportunities. *Computers and Education: Artificial Intelligence*, 4:100124.
- Yen-Ning Su, Chia-Cheng Hsu, Hsin-Chin Chen, Kuo-Kuang Huang, and Yueh-Min Huang. 2014. Developing a sensor-based learning concentration detection system. *Engineering Computations*, 31(2):216–230.
- Jo Tondeur, Johan Van Braak, Peggy A Ertmer, and Anne Ottenbreit-Leftwich. 2017. Understanding the relationship between teachers’ pedagogical beliefs and technology use in education: A systematic review of qualitative evidence. *Educational technology research and development*, 65:555–575.
- Auli Toom, Kirsi Pyhältö, and Frances O’Connell Rust. 2015. Teachers’ professional agency in contradictory times. *Teachers and teaching*, 21(6):615–623.
- UNESCO. 2021. *AI and education: Guidance for policy-makers*.
- Annemiek Veldhuis, Priscilla Y Lo, Sadhbh Kenny, and Alissa N Antle. 2024. Critical artificial intelligence literacy: A scoping review and framework synthesis. *International Journal of Child-Computer Interaction*, page 100708.
- Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. 2003. User acceptance of information technology: Toward a unified view. *MIS quarterly*, pages 425–478.

- Assunta Viteritti. 2009. A cinderella or a princess? the italian school between practices and reforms. *Italian Journal of Sociology of Education*, 1(Italian Journal of Sociology of Education 1/3):10–32.
- Jin Eun Yoo and Minjeong Rho. 2020. Exploration of predictors for korean teacher job satisfaction via a machine learning technique, group mnet. *Frontiers in psychology*, 11:441.
- H. Yuan, Y. Wang, and Y. Liu. 2020. [Automated essay scoring based on two-stage learning](#). *IEEE Access*, 8:21906–21915.
- Olaf Zawacki-Richter, Victoria I Marín, Melissa Bond, and Franziska Gouverneur. 2019. Systematic review of research on artificial intelligence applications in higher education—where are the educators? *International journal of educational technology in higher education*, 16(1):1–27.
- Li Zhang. 2025. Educational technology in a cross-cultural perspective: Applications and challenges of generative ai tools in language education in sino-foreign cooperative programs. *Journal of Humanities, Arts and Social Science*, 9(3).
- Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 295–305.

LLMs in alliance with Edit-based Models: Advancing In-Context Learning for Grammatical Error Correction by Specific Example Selection

Alexey Sorokin

MSU Institute for Artificial Intelligence Yandex

Regina Nasyrova

MSU Institute for Artificial Intelligence

Abstract

We show that fewshot Grammatical Error Correction might be improved by using an encoder-based sequence labeling model, such as GECTOR, to select similar examples. We demonstrate this on three Russian GEC corpora and English BEA corpus. The effect is the most significant for the new LORuGEC corpus and reaches up to 5-10% F0.5-score depending on the model. The corpus is released in our paper and contains 348 train and 612 test examples. The corpus is designed for diagnostic purposes and is also equipped with writing rules' annotations.

These annotations allow to further improve few-shot error correction by contrastive tuning of GECTOR-like encoder on rule classification task. This holds for a broad class of large language models. The best results are obtained with 5-shot YandexGPT-5 Pro model, achieving F0.5-score of 83%.

1 Introduction

The task of Grammatical Error Correction (GEC) may be defined in two ways, depending on whether the main objective is to make the sentence *grammatical*, i.e. applying minimal edits until it is grammatically correct, or *fluent*, namely transforming a sentence, likely substantially, so that it sounds natural yet saves the initial meaning (Coyne et al., 2023). In the era of Large Language Models (LLMs), researchers studied their ability in both settings (see more in the Section 2.1) and concluded that LLMs outperform mainstream GEC models in the latter objective (Coyne et al., 2023), demonstrating more freedom and creativity in sentence modifications. However, this asset becomes a burden in the former setting, where LLMs' 'generous' edits are treated as overcorrections.

A reasonable thing to do to make LLMs predict more reliable corrections as well as leverage their fluency and language knowledge is to apply them

in few-shot settings which proved to be valuable in many other NLP tasks, e.g. Machine Translation and Question Answering (Brown et al., 2020). As supported by Fang et al. (2023); Loem et al. (2023), in-context examples indeed enhance the quality and consistency of LLMs' corrections. However, the research of in-context learning in GEC pays little attention to example selection, the rare exception being Tang et al. (2024), using a syntactic structure similarity metric to select in-context examples.

We argue that sentences containing the errors of the same kind as the target ones may be much more beneficial as in-context examples rather than randomly selected ones. To prove this hypothesis, we present a novel approach to Grammatical Error Correction which makes use of a task-specific sequence labeling model (Omelianchuk et al., 2020) and retrieval-based few-shot learning. The sequence labeling model was trained to predict token-level edits, required to transform the source text into the grammatically correct one. We employ it to encode tokens in a sentence and choose the embeddings of the most likely edits as the representation of a sentence. After that, we use the retriever to select the closest sentence representations to the target one. As a result, the sentences corresponding to the selection are used as in-context demonstrations.

However, we assume that the notion of "errors of the same kind" may require an extension, involving the similarity of not only the edit but also the general pattern behind it. Since the same edits may occur in diverse contexts (e.g. comma insertions may be required before certain conjunctions or between subordinate clauses), the sentence with the same edit may not be informative enough. The model would not comprehend the utility of the given demonstration because it is unclear what it should pay attention to when sentences are completely distinct, apart from the edit.

That is why, we collect a new Linguistically Oriented Rule-annotated GEC dataset for Russian –

LORuGEC, which consists of sentences representing the rules of Russian grammar that are considered to be complicated both for L1 learners and large language models. These errors are also under-represented in the existing Russian GEC corpora, so we expect that the effect of in-context demonstrations would be the most prominent for this corpus.

We conduct experiments on Russian GEC datasets in zero-shot and few-shot (1-shot and 5-shot) settings. For the few-shot setting we study random example selection and retrieval-based selection with the GECTOR-like pretrained encoder. We additionally tune the retriever to select sentences related to the same rule. We choose several LLMs for testing and also present the results of their finetuned versions where possible.

Our main contribution is as follows:

- Novel GEC dataset for Russian, where sentences are also annotated for rules which are violated in them. The methodology of its collection makes it a challenging benchmark for LLMs, as it includes previously underrepresented cases.
- We are the first to apply the GECTOR-like(Omelianchuk et al., 2020) model for few-shot examples retrieval in grammatical error correction. The proposed approach yields considerably higher scores on LORuGEC dataset than random selection of examples for all models, supporting the impact of demonstrations’ quality and design on the performance of LLMs.
- Contrastive tuning of the retriever on related data additionally improves the quality of corrections on LORuGEC.
- The proposed method may compete with LLMs’ finetuning, especially if the training data is not large in size.

We make our data¹ and code² freely available.

2 Related work

2.1 Using LLMs for Grammatical Error Correction

Large Language Models gained prominence over the recent years as helpful tools for most Natural Language Processing tasks (Brown et al., 2020;

¹<https://github.com/ReginaNasyrova/LORuGEC>

²<https://github.com/AlexeySorokin/LORuGEC>

DeepSeek-AI et al., 2025). Their abilities were also tested on the Grammatical Error Correction task. Wu et al. (2023); Fang et al. (2023) show that ChatGPT³ performs worse, than commercial and conventional GEC models for English, being less prone to under-correction and mis-correction, but generating more fluent corrections, hence over-correcting, which is penalized severely by conventional metrics designed to evaluate minimal edits. Moreover, ChatGPT shows promising results for Multilingual GEC (Fang et al., 2023).

A more detailed analysis with fine-grained prompt and hyperparameter search was done in Coyne et al. (2023). They found that low temperature and suitable prompts increase the reliability of corrections produced by GPT-3.5(Ouyang et al., 2022) and GPT-4(OpenAI, 2023). Loem et al. (2023) proceed to research prompt-based methods for GEC, discovering that GPT-3(Brown et al., 2020) is much less prompt-sensitive and inconsistent, when supported with in-context examples.

Fang et al. (2023); Loem et al. (2023) propose that the investigation on the effect of example quality and design may be beneficial. An instance of it is introduced in Tang et al. (2024), where sentences with the same syntactically incorrect structure are adopted as in-context examples, significantly outperforming randomly selected ones. Advancing the choice of in-context examples, Robotian et al. (2025) propose Retrieval-Augmented Generation within In-Context Learning approach to improve Generative Error Correction in speech recognition systems. Other works also consider LLMs’ instruction tuning and ensembling for GEC (Kaneko and Okazaki, 2023; Omelianchuk et al., 2024).

2.2 In-context learning for LLMs

Our work is an example of the so-called retrieval-based few-shot learning, where demonstration samples are selected according to some similarity measure between vectors. A review of retrieval-based in-context learning is presented in Xu et al. (2024). The early examples of this approach include Rubin et al. (2022) where retrieval-based selection of demonstrations was shown to improve performance for three sequence-to-sequence learning tasks. The authors also demonstrated that one may reach further gains by training the retriever to select examples that maximize the correct output probability. Margatina et al. (2023) verified the positive role

³<https://openai.com/index/chatgpt/>

of similarity between test and in-context examples on a diverse range of models and tasks including classification and multiple choice datasets. [Nori et al. \(2023\)](#) demonstrated that using KNN-based few-shot example selection allows to adapt general models to medical domain without special tuning.

2.3 GEC corpora for Russian

There are three available Russian GEC datasets: RULEC-GEC([Rozovskaya and Roth, 2019](#)), RU-Lang8([Trinh and Rozovskaya, 2021](#)) and GERA([Sorokin and Nasyrova, 2025](#)). The first one represents a subset of the Russian Learner Corpus of Academic Writing (RULEC)([Alsufieva et al., 2012](#)), containing essays of the US students who were either learning Russian as a foreign language or heritage speakers. The authors comprised a list of 23 error type labels that cover (morpho)syntactic, lexical and spelling errors.

The RU-Lang8 Dataset constitutes a subset of the Lang-8 Corpus([Mizumoto et al., 2012](#)) learner corpus, based on the language learning website⁴. Most texts in RU-Lang8 are much shorter, being small paragraphs or learners’ questions. Unlike RULEC-GEC, RU-Lang8 has a more coarse-grained annotation, with error type labels representing operations of token replacement, deletion, insertion and change in word order.

As opposed to both datasets, GERA is based on Russian school texts and was annotated in line with a much more fine-grained label inventory, i.e. grammatical error types cover a broader list of parts of speech and grammatical categories, and there are different types of lexical and spelling errors depending on the erroneous construction.

2.4 Linguistically motivated data for GEC

Usually GEC corpora are based on real-world learner data, not a predefined error taxonomy. A partial example of error-driven approach was [Volodina et al. \(2021\)](#), where the four principal error types from existing data were selected to be included in the dataset. Similarly to LORuGEC, most examples in their corpus contain exactly one error.

More frequently, error taxonomies are used for collecting linguistic acceptability data. The most well-known example of such corpora are COLA([Warstadt et al., 2019](#)) and BLIMP([Warstadt et al., 2020](#)) for English. One may even convert a BLIMP-like dataset of minimal pairs to GEC for-

mat, by using the ungrammatical element of the pair as the source and the grammatical one – as the target, this approach was adopted in [Volodina et al. \(2021\)](#) for Swedish and [Jentoft and Samuel \(2023\)](#) for Norwegian. Concerning Russian language, BLIMP-like datasets of minimal pairs were introduced in the recent works of [Graschenkov et al. \(2024\)](#) and [Taktasheva et al. \(2024\)](#).

3 LORuGEC: Corpus description

3.1 Motivation and data collection

Most existing GEC corpora consist of L2 learners’ data. Even corpora based on native learners’ data mostly reflect the real-world error distribution, underrepresenting complicated grammatical rules. Concerning the Russian language, existing corpora, such as RULEC-GEC([Rozovskaya and Roth, 2019](#)), RU-Lang8([Trinh and Rozovskaya, 2021](#)) and GERA([Sorokin and Nasyrova, 2025](#)), contain very few examples of complex, “school-book” rules, making these corpora suboptimal for use in educational applications. Our primary goal is to fill this gap and collect a corpus of complex cases that represent the rules which are considered difficult for Russian L1 learners. The second goal of our project is to study, which rules present the highest complexity for modern LLMs in the task of Grammatical Error Correction.

Given our research goals, we organize the data collection and annotation process as follows:

1. Firstly, one of the paper authors (a bachelor in Linguistics) collected an initial set of about 10 rules that are known as difficult for Russian high school students. These rules covered various fields of writing, mostly punctuation, grammar and spelling. The list of rules was checked by another author of the paper and verified using several Russian grammar books.
2. For each of the selected rules, the annotators, which were students with linguistic backgrounds and Russian native speakers, were asked to collect up to 15 examples belonging to these rules. Since the collected examples were intended to be used for LLM benchmarking, several precautions were taken, which were expressed in the instruction (see more in [Appendix A](#)), as follows:
 - Preferably, choose sentences from different sources.

⁴<https://lang-8.com/>

- Avoid using quotations from fiction.
 - Refrain from selecting commonplace examples.
3. The collected examples were corrupted to simulate the common mistakes corresponding to particular rules. For example, if the rule governs the use of comma between the conjuncts, the comma was either deleted in the contexts where it was required or inserted when it must not be used. If there are multiple ways to introduce errors, the examples should cover them all. For instance, clauses with participles in Russian should be surrounded by commas, so possible corruptions included deletion of both commas, only the preceding comma or only the following one.
 4. The collected examples were passed through the YandexGPT3 Pro⁵ model. The goal of this stage was to identify complex sentences and make the dataset more challenging by including analogous examples.
 5. After successfully completing the data collection for the initial set of 10 rules, the annotators were allowed to select the subsequent rules themselves. They were instructed to consult grammar reference books and cover all fields of written language, such as punctuation, spelling, grammar (in the narrow sense) and lexis. The process was supervised by the principal annotator (one of the authors) who checked the selection of rules and example cases, as well as their annotation. Since the source sentences were created by targeted manual corruption, the correct sentence was known in advance, thus reducing the correction ambiguity. The principal annotator additionally analyzed 100 random samples and found no disagreement with the annotators.

3.2 Data sources

While selecting the rules, annotators and authors used various resources, such as grammar reference books, teacher manuals and educational websites based on them, we refer to **B** for the full list of data sources. The textbooks that were used comply with Russian educational standards, some of them are specially approved by the Russian Academy of Sciences, for example, (Valgina et al., 2009).

⁵<https://yandex.cloud/ru/docs/foundation-models/concepts/yandexgpt/models>

3.3 Rules Description and Statistics

We gathered 48 rules from 4 grammar sections. The majority of them represent punctuation and spelling. We present the comprehensive list of rules in Appendix C.

We collected 960 pairs of sentences (an average of 20 sentences per rule), which were split into validation and test subsets so that for each rule at least 9 sentences or approximately two thirds of collected sentences would be allocated to the test partition. Consequently, the size of the test subset is twice as large as the size of the validation one (see Table 1). Additionally, unlike the latter, only the test subset includes initially correct sentences (for hypercorrection considerations). See more on the data format in Appendix D.

3.4 Comparison with other GEC corpora for Russian

Comparing to existing Russian GEC corpora, such as RULEC-GEC (Rozovskaya and Roth, 2019), RU-Lang8 (Trinh and Rozovskaya, 2021) and GERA (Sorokin and Nasyrova, 2025), our data differs in several aspects:

- To the best of our knowledge, that is the only Russian GEC corpus where all the errors are matched with corresponding grammar rules instead of error type.
- Our corpus is deliberately created for evaluation and diagnostic purposes. Therefore, it has no training subset and is much smaller than other corpora (see Table 2). We do not want LLMs to acquire new capabilities on the validation set of our corpus, but rather to reveal the knowledge they already have.

On the other hand, almost all sentences in our corpus contain errors and are supposed to be challenging in contrast to other GEC data.

- Since corpus examples were created via corruption, for the vast majority of mistakes there is only one possible correction, increasing the trustworthiness of evaluation scores.
- As shown in Table 3, LORuGEC has the highest fraction of pattern-based errors covered by a rule-based generator. These errors include punctuation errors, word form changes, deletion, insertion or replacement of closed word categories (prepositions, conjunctions and pronouns), spelling errors, etc. Despite this, the

Sample	Sentences	Correct source sentences	Sentences for complex rules (%)	Tokens
Validation	348	0	250 (71.84)	5,579
Test	612	31	419 (68.46)	10,131

Table 1: Statistics on the validation and test samples of LORuGEC.

Sample	Sentences	Tokens
RULEC-GEC	12,480	206,258
RU-Lang8	4,412	54,741
GERA	6,681	119,068
LORuGEC	960	15,710

Table 2: Quantitative comparison of GEC datasets for Russian.

corpus	P	R	F0.5	uncov., %
RULEC-GEC	50.4	32.6	45.5	42.0
RU-Lang8	60.8	37.9	54.2	48.8
GERA	74.3	47.0	66.6	33.7
LORuGEC	45.1	17.7	34.4	21.9

Table 3: Comparison of GEC model performance and difficult fraction (uncov., %) for different Russian GEC corpora. The model is Qwen2.5-7B finetuned on the concatenation of Russian GEC data.

GEC model finetuned on the concatenation of 3 Russian GEC corpora (see Section 5 for details) has much lower scores on LORuGEC than on other corpora. This implies that the main problem on LORuGEC is not to generate the suggestion but to discriminate between correct and incorrect variants.

4 Similar example retrieval

4.1 Approach description

We suppose that large language models may lack knowledge about specific Russian grammar rules. This information might be injected during inference via in-context example selection. A natural solution might be to select examples that belong to the same rule, i.e. resembling not only the required correction, but also the grammatical reasoning behind it. However, this restricts the method to a predefined bounded set of rules that prevents the model from real-world usage.

Our approach is to use an embedder to select training examples similar to the given test sentence. We want this embedder to reflect grammatical sim-

ilarity. That is not the case for standard sentence embedders that assign similar vector representations to semantically similar sentences. To be used for similar examples retrieval, the embedder should be pretrained on a grammar-related task.

We decide to select the famous GECTOR model offered by (Omelianchuk et al., 2020). Their approach does not treat GEC as a Machine Translation task but reduces it to sequence labeling, taking into account the fact that most tokens in a sentence remain unchanged after the correction. GECTOR classifier, which is built upon a pretrained encoder⁶, predicts the no-operation label KEEP for such tokens. In other cases, labels represent

- **elementary edit operations**, such as DELETE, REPLACEWITH_<TOKEN> (e.g., replace the current word with the word *on*) or INSERT_<TOKEN>, where <TOKEN> may refer to not only words, but also punctuation marks.
- **grammatical transformations** which mostly have to do with inflection (e.g., GRAM\$SING, meaning ‘put the current word in the singular form instead of plural’).

Although the latter labels, the so-called G-labels, do not exactly correspond to rules of writing, mistakes from the same rule class often obtain the same label. Since the hidden states of encoder models reflect the similarity in their label space, this similarity is also related to rule similarity.

4.2 Implementation details

Although the retrieval based on embedding similarity is very common and is extensively used, e.g., in Retrieval-Augmented Generation (RAG), the adaptation of GECTOR to retrieval has several details. Firstly, as GECTOR operates on token level, it does not assign meaningful representation to the [CLS] token usually used for retrieval. We represent the sentence with the hidden states from the final encoder layer and select up to 3 hidden states

⁶<https://huggingface.co/ai-forever/ruRoberta-large> in our case.

corresponding to the most probable error positions. The probability of an error is predicted by the GECTOR model itself, using $1 - p(\text{KEEP})$, where KEEP is the no-edit label of the GECTOR model.

Since the original GECTOR model uses obsolete Python libraries and the sets of G-labels differ significantly between English and Russian, we reimplement the model by ourselves using HuggingFace Transformers⁷ library. The details of its training are available in Appendix F.

4.3 Retriever finetuning

We suppose that pretraining on external data empowers the model with the basic information about grammatical error patterns, but the model might not have enough knowledge about rare or dataset-specific rules. Therefore, we propose to finetune the retriever on the task of rule classification using contrastive learning. The tuning is performed on the validation part of our dataset. The training objective is a standard triplet loss

$$L(h, h^+, h^-) = \max\left(\frac{\rho(h, h^+) - \rho(h, h^-) + \alpha}{t}, 0\right),$$

where ρ is the distance function (e.g., cosine), α is the margin and t is the temperature. We always use as h^+ the closest example with the same class label and as h^- – the closest example with another class label. In terms of contrastive learning literature, we use hard positives and hard negatives without in-batch negatives.

We retrieve the closest positive and negative examples once in epoch. After completing the epoch we recalculate the triples using the updated embedder. Further details are given in Appendix F.

5 Model evaluation

In this section we evaluate several LLMs on our corpus⁸. We select two open-source models: the open-source *Qwen-2.5 7B Instruct*⁹(Yang et al., 2024) and *yandex/YandexGPT-5-Lite-8B-instruct*¹⁰ as well as closed-source *YandexGPT-5*

⁷<https://huggingface.co/docs/transformers/index>

⁸We restrict our attention to LLMs by two reasons: first, one of our goals is to study few-shot learning approach. Second, in contrast to English, LLMs outperform other approaches, such as encoder-decoder or GECTOR-like, on available Russian data.

⁹<https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>

¹⁰<https://huggingface.co/yandex/YandexGPT-5-Lite-8B-instruct>

*Pro*¹¹. The latter two models are selected because they were largely trained on Russian data and the first one is chosen due to its excellent multilingual abilities. We evaluate additional models, such as *LLama3-8B-Instruct*(Meta, 2024) and *GPT4o-2024-05-13*(OpenAI, 2023), in Appendix G.1.

We compare several settings:

1. zero-shot prompt-based application of LLM. The prompt is provided in Appendix E.1.
2. few-shot prompt-based application of LLMs with different selection of in-context examples: random, the general purpose e5-base-multilingual¹²(Wang et al., 2024) embedder, pretrained GECTOR and GECTOR with contrastive finetuning).
3. finetuning open-source LLMs on external Russian GEC data: RULEC-GEC, RU-Lang8 and GERA(Sorokin and Nasyrova, 2025).
4. further training of the finetuned LLMs on the validation part of our corpus.
5. LORA-based training of open-source LLMs only on the validation part of our corpus.

The hyperparameters of the finetuning are available in Table 9.

As is commonly done, we score the tokenized model outputs with M2scorer(Dahlmeier et al., 2013) and report precision, recall and F0.5 score, using F0.5 as the main metric. The results are given in Table 4. We make the following conclusions:

1. Finetuning on external GEC data is detrimental for LORuGEC. Since LORuGEC types of errors are rare in general GEC corpora, the finetuned model decides not to correct them, hence, its recall dramatically reduces.
2. With a single exception, the GECTOR retriever performs better than the random one, proving our first hypothesis: **during pre-training on general GEC data, the encoder learns the representations for error types**. Moreover, these representations are helpful even for rare classes of errors that the LLM was not able to learn. In contrast, the general-purpose e5-base-multilingual embedder produces much smaller improvements.

¹¹<https://yandex.cloud/ru/docs/foundation-models/concepts/yandexgpt/models>

¹²<https://huggingface.co/intfloat/multilingual-e5-base>

Setup	Qwen2.5-7B			YandexGPT5-Lite			YandexGPT5-Pro		
	P	R	F0.5	P	R	F0.5	P	R	F0.5
zero-shot	43.3	34.0	41.0	66.4	51.0	62.6	76.5	66.7	74.3
1-shot, random	44.4	28.6	40.0	67.8	48.6	62.8	78.3	71.0	76.7
5-shot, random	47.2	30.2	42.4	68.5	56.3	65.6	83.9	79.2	83.0
1-shot, e5-base	44.6	29.5	40.5	69.4	49.4	64.2	81.6	69.7	78.9
5-shot, e5-base	47.0	31.8	42.9	68.8	56.8	66.0	81.8	72.2	79.7
1-shot, GECTOR	50.2	35.8	46.5	69.9	53.9	66.0	81.9	72.8	79.9
5-shot, GECTOR	54.3	41.7	51.2	70.0	62.4	68.3	82.7	76.7	81.4
1-shot, GECTOR+FT	52.7	39.8	49.5	71.2	56.7	67.7	83.0	76.3	81.6
5-shot, GECTOR+FT	59.3	46.2	56.1	73.1	65.5	71.4	83.5	78.1	82.3
ext. finetuning	45.1	17.7	34.4	67.0	35.4	56.9	NA		
ext.+LORuGEC finetuning	50.1	37.9	47.1	77.4	73.6	76.6	NA		
LORuGEC LORA finetuning	48.6	42.6	47.3	74.1	72.6	73.8	NA		

Table 4: Comparison of different LLMs on the LORuGEC test set in zero-shot, few-shot and finetuning modes. Ext. finetuning refers to training on the concatenation of other Russian GEC corpora. The best metric inside the same approach (e.g., 1-shot) is presented in italics and the best overall metric – in bold.

- Contrastive finetuning of the embedder is also helpful: the 1-shot GECTOR+FT retrieval almost matches the performance of 5-shot GECTOR retrieval. This proves our second hypothesis: **In-domain contrastive tuning of the retriever improves the quality of few-shot error correction**. This also proves the usefulness of rule annotation that distinguishes our corpus from general GEC data.
- The models of the YandexGPT-5 family handle “schoolbook” errors from LORuGEC much better than Qwen-2.5 does. The details of their training are not available, however, it is likely that they saw more high-quality Russian data than the multilingual Qwen model.

5.1 Detailed results and examples

In Table 5 we also report the results per category for different error types. For both compared models punctuation errors are the easiest and the lexical ones – the hardest. A plausible explanation of this fact is that punctuation rules are the most strict, mostly binary (whether to use the comma or not) and rely on separate tokens, while the lexical rules are more vague and usually deal with more options.

When training the embedder, we use the retrieval quality as an intrinsic quality metric: the more often the embedder retrieves examples that belong to the same rule, the better it is. We observe that this internal metric correlates well with error correction quality, as shown in Table 6.

We provide illustrative examples of retrieved

Category	Qwen2.5-7B			YandexGPT5-Pro		
	P	R	F0.5	P	R	F0.5
Grammar	50.0	36.5	46.6	86.3	69.8	82.4
Lexis	46.7	22.6	38.5	85.0	54.8	76.6
Punct.	66.2	53.6	63.0	85.7	83.3	85.2
Spelling	55.2	44.9	52.8	80.9	77.4	80.2

Table 5: Per-category scores of 5-shot learning, GECTOR+FT retriever for Qwen2.5-7B and YandexGPT5-Pro models.

samples together with corresponding model outputs in Figures 3 and 4.

5.2 Results for other corpora

The results on the introduced LORuGEC corpus prove the utility of our approach on a rule-oriented corpus. We wonder whether GECTOR-based demonstration selection improves results for general GEC corpora as well. To verify it, we compare three types of few-shot example selection (random, GECTOR and GECTOR+FT) on three available corpora: RULEC-GEC, RU-Lang8 and GERA. The results for the first two corpora are provided in Table 7, the results for GERA are in Table 13.

We again observe the advantage of GECTOR-based examples over random samples. Finetuning of GECTOR retriever on LORuGEC data does not have a clear positive effect probably due to the difference in error distribution between corpora. Due to larger sizes of these corpora, few-shot learning is not able to outperform full finetuning, but demon-

Retriever	acc.	top-5 recall	Qwen2.5-7B F0.5		YandexGPT5-Pro F0.5	
			1-shot	5-shot	1-shot	5-shot
random	2.3	10.3	40.0	42.4	76.7	83.0
GECTOR	31.7	49.3	46.5	51.2	79.9	81.4
GECTOR+FT	55.9	72.2	49.5	56.1	81.6	82.3

Table 6: Correlation between retrieval and GEC metrics for different retrievers. Accuracy is the percentage of cases when the most closest example belongs to the same rule and recall-5 – the fraction of cases when such examples occur among top 5 closest examples.

Setup	Qwen-2.5 7B Instruct						YandexGPT-5 Lite 8B Instruct					
	RULEC-GEC			RU-Lang8			RULEC-GEC			RU-Lang8		
	P	R	F0.5	P	R	F0.5	P	R	F0.5	P	R	F0.5
zero-shot	38.2	39.3	38.4	48.9	39.2	46.6	41.7	42.6	41.9	53.8	41.9	50.9
random, 1-shot	40.7	37.8	40.1	50.4	37.1	47.1	43.5	41.9	43.2	55.1	42.5	52.0
random, 5-shot	42.4	37.9	41.4	51.6	38.3	48.2	43.7	45.1	44.0	55.4	47.5	53.6
gector, 1-shot	<i>41.8</i>	37.6	<i>40.9</i>	<i>53.7</i>	38.8	<i>49.8</i>	45.0	42.5	44.5	56.9	43.5	53.6
gector, 5-shot	43.9	37.1	42.4	55.4	40.2	51.5	46.0	45.4	45.9	57.2	48.3	55.2
gector+FT, 1-shot	41.7	37.2	40.7	52.6	38.1	48.8	<i>45.4</i>	42.2	<i>44.7</i>	<i>57.1</i>	43.7	53.8
gector+FT, 5-shot	<i>44.7</i>	38.1	43.2	55.3	40.7	<i>51.6</i>	<i>46.1</i>	45.8	<i>46.0</i>	56.0	47.7	54.1
finetuning	52.2	31.2	46.0	61.7	37.2	54.5	57.3	38.9	52.4	66.3	48.5	61.8
prev. SOTA	70.5	29.1	54.8 ²	73.7	27.3	55.0 ¹	70.5	29.1	54.8 ²	73.7	27.3	55.0 ¹

Table 7: Comparison of different few-shot example selection methods on RULEC-GEC and RU-Lang8 corpora. The best metric inside the same approach (e.g., 1-shot) is presented in italics and the best overall metric – in bold. ¹ refers to Sorokin (2022) and ² to Sorokin and Nasyrova (2025)

strates higher recall in 3 of 4 experiments.

We also apply our approach to English BEA corpus, see Appendix G.3 for details. There GECTOR-based example selection leads to a small (about 1.5% F0.5 score) but consistent improvement.

6 Discussion and conclusions

In our study we make two principal contributions:

1. We release a new LORuGEC corpus, which differs from existing Russian GEC corpora in data sources, difficulty and typology of errors and, most importantly, the presence of rule labels. This annotation makes our corpus more suitable for L1 educational applications, such as school writing assistants.
2. We compare several methods of in-context learning on our data and discover that retrieval-based demonstration selection significantly outperforms random choice. The retrieval leverages the encoder-based GECTOR model. Contrastive finetuning of this encoder to predict rule labels further improves correction quality.

Since our data has a distinct error distribution, we also check the second result on other corpora. We observe that GECTOR-based in-context examples retrieval is beneficial over random selection. This confirms that our approach effectively works for general GEC data, at least for Russian.

As a future work, we plan to extend our corpus in terms of size and errors number. We have already collected a small pool of sentences with multiple errors, which require additional verification. To reduce annotation burden, we also experimented with example generation. We found that LLM may effectively generate 5 examples to the required rules: 23 out of 25 samples were correct, however, they were shorter and less variable than the manually collected ones, thus further investigation is needed.

We also believe our approach to be viable in domains where task-induced similarity differs from surface meaning similarity. For example, in code retrieval similar programs are not the ones using the same variable names but the ones using the same algorithms. So we hope to investigate the usefulness of our approach in other fields.

7 Limitations

1. As for any LLM-based method, our results are prompt-dependent. In particular, our prompts were optimized towards YandexGPT models and might be suboptimal for the models from other families or for later versions of YandexGPT. However, we did not find any major differences in results when slightly modifying the prompt.
2. For now we evaluate our approach only on Russian and the results may differ for other languages. However, the approach itself has no language-specific details.
3. The LORuGEC corpus is rather small in size compared to other GEC corpora, thus the result may change after collecting more analogous data. We addressed this question in the Conclusions section.
4. Though in principle contrastive tuning works with multiple example labels, we were unable to successfully extend our approach to the multilabel case.

8 Ethics considerations

Our work is based on Large Language Models. We acknowledge that such models might be used in a harmful or malicious manner, however, we utilize them only for scientific purposes. Nevertheless, if a retrieved fewshot sample includes an unsafe generation, that may bias the model towards undesirable behaviour. Thus generalizing our method to datasets containing such examples requires additional precautions.

All of the students who participated in the creation of the dataset earned credit hours as a result. The students were informed about the goals of the work and gave their content for dataset publication.

Acknowledgments

References

- Anna Alsufieva, Olesya Kisselev, and Sandra Freels. 2012. [Results 2012: Using flagship data to develop a Russian learner corpus of academic writing](#). *Russian Language Journal*, 62:79–105.
- Svetlana Berezina and Nikolaj Borisov. 2017. *Russkij yazyk v sxemax i tablicax (in Russian)*. Eksmo, Moskva.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Christopher Bryant, Mariano Felice, 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.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the Performance of GPT-3.5 and GPT-4 in Grammatical Error Correction. *arXiv preprint arXiv:2303.14342*.

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.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanxia Zhao, Wen Liu,

- Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is ChatGPT a Highly Fluent Grammatical Error Correction System? A Comprehensive Evaluation. *arXiv preprint arXiv:2304.01746*.
- Pavel Grashchenkov, Lada Pasko, Kseniia Studenikina, and Mikhail Tikhomirov. 2024. [Russian parametric corpus ruparam \(in Russian\)](#). *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 24(6):991–998.
- Matias Jentoft and David Samuel. 2023. NoCoLA: The norwegian corpus of linguistic acceptability. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 610–617.
- 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.
- Elena Kochneva. 1983. *Slovar' sochetaemosti slova russkogo yazyka (in Russian)*. Russkij yazyk, Moskva.
- 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. *arXiv preprint arXiv:2305.18156*.
- Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. [Active learning principles for in-context learning with large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5011–5034, Singapore. Association for Computational Linguistics.
- AI Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2(5):6.
- Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. [The effect of learner corpus size in grammatical error correction of esl writings](#). In *Proceedings of COLING 2012: Posters*, pages 863–872.
- Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*.
- 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.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzshanskyi, 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. *arXiv preprint arXiv:2404.14914*.
- OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- 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.
- Amin Robatian, Mohammad Hajipour, Mohammad Reza Peyghan, Fatemeh Rajabi, Sajjad Amini, Shahrokh Ghaemmaghami, and Iman Gholampour. 2025. GEC-RAG: Improving Generative Error Correction via Retrieval-Augmented Generation for Automatic Speech Recognition Systems. *arXiv preprint arXiv:2501.10734*.
- 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.
- Ditmar Rozenal'. 1997. *Spravochnik po pravopisaniyu i stilistike (in Russian)*. Komplekt, SPB.
- 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.
- Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.
- Elena Simakova. 2016. *Russkij yazyk: Novyj polnyj spravochnik dlya podgotovki k EGE' (in Russian)*. AST: Astrel', Moskva.
- 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.
- Alexey Sorokin and Regina Nasyrova. 2025. **Gera: A corpus of russian school texts annotated for grammatical error correction**. In *Analysis of Images, Social Networks and Texts*, pages 148–163, Cham. Springer Nature Switzerland.
- Ekaterina Taktasheva, Maxim Bazhukov, Kirill Koncha, Alena Fenogenova, Ekaterina Artemova, and Vladislav Mikhailov. 2024. **Rublimp: Russian benchmark of linguistic minimal pairs**. *arXiv preprint arXiv:2406.19232*.
- Chenming Tang, Fanyi Qu, and Yunfang Wu. 2024. Ungrammatical-syntax-based in-context example selection for grammatical error correction. *arXiv preprint arXiv:2403.19283*.
- Viet Anh Trinh and Alla Rozovskaya. 2021. **New dataset and strong baselines for the grammatical error correction of Russian**. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4103–4111, Online. Association for Computational Linguistics.
- Nina Valgina, Nataliya Es'kova, Ol'ga Ivanova, Svetlana Kuz'mina, Vladimir Lopatin, and Lyudmila Chel'cova. 2009. *Pravila russkoj orfografii i punktuacii. Polnyj akademicheskij spravochnik (in Russian)*. AST, Moskva.
- Nina Valgina, Ditmar Rozenal', and Margarita Fomina. 2002. *Sovremennyy russkij yazyk: Uchebnik (in Russian)*. Logos, Moskva.
- Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 2021. **DaLaj – a dataset for linguistic acceptability judgments for Swedish**. In *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 28–37, Online. LiU Electronic Press.
- 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*.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. **BLiMP: A benchmark of linguistic minimal pairs for English**. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.
- Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. **Neural network acceptability judgments**. *Transactions of the Association for Computational Linguistics*, 7:625–641.
- Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. ChatGPT or Grammarly? Evaluating ChatGPT on Grammatical Error Correction Benchmark. *arXiv preprint arXiv:2303.13648*.
- Xin Xu, Yue Liu, Panupong Pasupat, Mehran Kazemi, et al. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

A Annotation Instruction

Выберите грамматический справочник по русскому языку, затем составьте набор правил.

Для каждого правила найдите 15 примеров (предложений). Предложения должны быть из разных источников и желательно не из художественной литературы. Примеры также не должны быть тривиальными.

Добавьте в предложения нарушения той нормы, которую Вы исследуете. Если есть несколько способов допустить ошибку в правиле, отразите это в собранных примерах.

Для каждого правила протестируйте YandexGPT 3 Pro на его примерах. Если модель не справилась хотя бы в одном примере, то проанализируйте, что отличает сложные предложения, и соберите еще 5-10 сложных примеров.

(Select a reference book for Russian, after that choose the rules for consideration.

For each rule find 15 example sentences that are preferably from different sources and not trivial, avoid using examples from fiction.

Add errors to the sentences based on the rule under consideration. If there are several ways of making a mistake in a rule, this should be reflected in the collected set of sentences for it.

For each rule test the YandexGPT 3 Pro on its sentences. If there are any imperfections in the

model's corrections, analyse what distinguishes complicated sentences and gather 5-10 more complex examples.)

B Educational sources of the rules

- High school Unified State Exam preparation books: (Berezina and Borisov, 2017) (Simakova, 2016)
- Academic handbook on spelling and punctuation: (Valgina et al., 2009), <http://orthographia.ru/>
- Handbook on the contemporary Russian language: (Valgina et al., 2002), <https://pedlib.ru/Books/6/0262/>
- Handbook on spelling and stylistics: (Rozenal', 1997), <https://rosental-book.ru/>
- Dictionary of Russian collocations: (Kochneva, 1983)
- Educational web-sources: <https://orfogrammka.ru/>, <https://gramota.ru/biblioteka/spravochniki/>, <http://old-rozenal.ru/>, <https://grammatika-rus.ru/>, https://licey.net/free/4-russkii_yazyk/, <https://www.yaklass.ru/p/russky-yazik/>

C Rules of Russian grammar in LORuGEC

• Grammar

- 1 Incorrect expression of government
- 2 Declension of cardinal numerals
- 3 Declension of numerals *poltora* ('one and a half.NOM'), *poltory* ('one and a half.GEN'), *poltorasta* ('a hundred and fifty.NOM')
- 4 Agreement between the participle and the word it defines

• Punctuation

- 5 Commas in idiomatic expressions
- 6 Commas between homogeneous subordinate clauses
- 7 Commas between subordinate and main clauses
- 8 Commas between the two conjunctions
- 9-11 Commas before the conjunction *kak* ('as'): 3 instances

- 12 Sentences with homogeneous parts
- 13 Converbs after conjunctions
- 14 Clauses related to the personal pronoun
- 15 Clauses that are distant from the word they define
- 16 Punctuation in meaningful (indecomposable) expressions
- 17 Linking words and constructions
- 18 Recurring conjunctions
- 19 Dashes in sentences with no conjunctions
- 20 Dashes between the subject and the predicate
- 21 Dashes in case of appositions

• Semantics

- 22 Collocations
- 23 Pleonasms

• Spelling

- 24 *n* and *nn* in the suffixes of adjectives
- 25 Vowels in the suffixes of participles
- 26 Noun suffixes *on'k*, *en'k*
- 27 Suffixes *ic*, *ec* in neuter nouns
- 28 Suffixes *ek*, *ik*
- 29 Adjective suffixes *insk*, *ensk*
- 30 Prefixes *pre* and *pri*
- 31 *y* and *i* after prefixes
- 32 Vowels after *c*
- 33 Vowels after sibilants
- 34 Separating soft and hard signs
- 35 Hyphens as part of written equivalents of complex words
- 36 Joint, separate or hyphenated spelling of adverbs
- 37 Compound adjectives
- 38 Particle *taki* ('still')
- 39 *zato* ('at least')
- 40 *ottogo* ('that is why')
- 41 *prichyom* and *pritom* ('moreover')
- 42 *takzhe* ('also')
- 43 *chtoby* ('to')
- 44 *pol-* ('half')
- 45 *ne* (negative particle) with verbs
- 46 *ne* with adjectives
- 47 *ne* with participles
- 48 *ne* with nouns

Complexity. As may be observed in the figure 1, the largest percentages of collected complex rules occur among punctuation and semantics.

D Details on LORuGEC format

The dataset consists of rules, their definitions, information on their complexity for the YandexGPT model, pairs of corresponding tokenized¹³ grammatical and ungrammatical sentences (see Table 8). There is some additional information, representing grammar sections which rules pertain to, sources of rules as well as indication of the subset for each sentence (validation or test, see more in the next section). There are few sentences in the dataset that do not contain any errors (see column *Correct source sentences* in Table 1), because it is also crucial to verify if models are prone to hypercorrection. These sentences are also marked with metadata. We also present our data in .M2, which is a conventional GEC format.

An example from LORuGEC in the first format type may be seen in the Table 8.

The same sentence, but expressed in the .M2-standard:

```
S Иванова , как художника , я совсем не знаю .
A 1 2|||None|||||REQUIRED|||-NONE-|||0
A 4 5|||None|||||REQUIRED|||-NONE-|||0
```

According to the .M2-standard, the source text is denoted with S, while the corresponding edits are prefixed with A. Each edit consists of the error span, error type, correction, if the edit is optional or required, additional remarks and annotator ID, yet we do not make use of error types. The given annotation demonstrates the requirement to delete two commas in the sentence.

E Model hyperparameters

E.1 Model prompt

Our final prompt for grammatical error correction of Russian texts is given in Figure 2.

E.2 Training hyperparameters

We train the model with Huggingface Transformers Trainer using the hyperparameters from Table 9 for all experiments. When two values are given, the first value is used for training from scratch, and the second – for finetuning from a checkpoint that was already trained on a larger general GEC corpus.

¹³We made use of NLTK Tokenizer: <https://www.nltk.org/api/nltk.tokenize.html>.

We also made the following model-specific changes:

1. Llama-8B-Instruct is tuned using `learning_rate = 3e-6` and YandexGPT5-Lite using `learning_rate = 1e-6`. For both these models we use `max_grad_norm = 0.3`.
2. LORA finetuning is performed with `learning_rate = 1e-4` and physical batch size 4.

F Retriever training

F.1 GECTOR pretraining

Since the morphological features of English and Russian differ significantly, we reimplement the GECTOR preprocessing by ourselves. The sets of G-labels correspond to combinations of morphological features, e.g., the label `NOUN,Nom+Plur` corresponds to putting the noun into Plural number and Nominative case, keeping other morphological features intact. When the corpus is converted into the pairs of word sequences and their edit labels, we implement training using standard HuggingFace Transformer instruments for sequence labeling. We omit the decoder as we do not need the exact surface transformations predicted by the GECTOR model, but only its labels and hidden states.

We train the GECTOR model on the concatenation of RULEC-GEC, ru-Lang8, GERA and 1 million sentences with synthetic errors. When generating synthetic data, we use the Russian subset of Oscar corpus¹⁴ as source and introduce artificial errors simulating the error distribution of three mentioned corpora. The model is initialized from ruRoberta-large¹⁵ model, the hyperparameters of training are given in Table 10.

F.2 Contrastive tuning

As mentioned in Subsection 4.3, we tune the retriever on the task of rule label prediction using contrastive learning. The tuning is performed on the validation set of LORuGEC. The training objective is a standard triplet loss

$$L(h, h^+, h^-) = \max\left(\frac{\rho(h, h^+) - \rho(h, h^-) + \alpha}{t}, 0\right),$$

where ρ is the distance function (e.g., cosine), α is the margin and t is the temperature. Here h^+

¹⁴<https://huggingface.co/datasets/oscar-corpus/OSCAR-2109>

¹⁵<https://huggingface.co/ai-forever/ruRoberta-large>

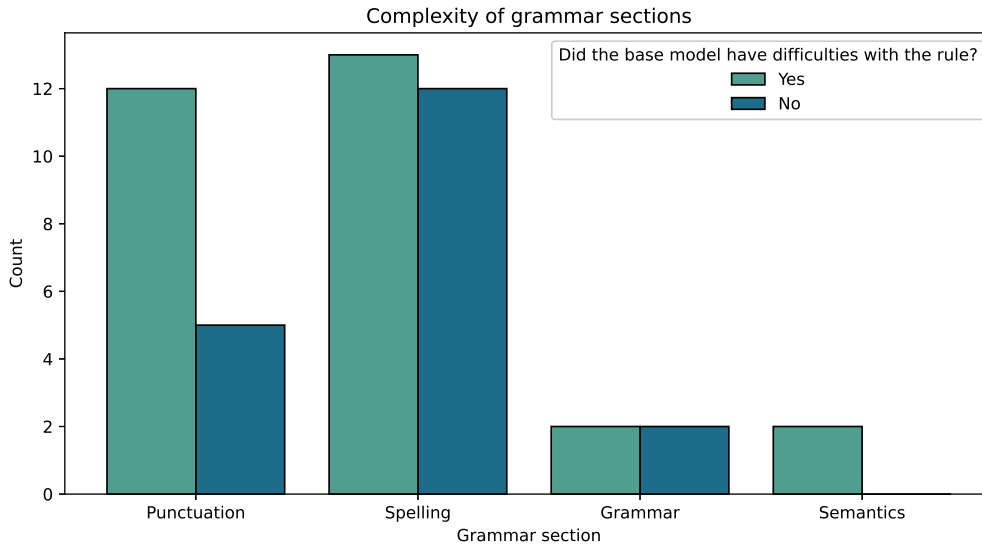


Figure 1: Complexity of different grammar sections is expressed by the number of complex rules for the YandexGPT3 Pro model. We considered the rule to be difficult if the model failed to correct some of its sentences (see 3.1).

The rule	Did the base model have difficulties with the rule?	Initial sentence	Correct sentence
Запятая перед союзом “как”: 2 случая (Commas before the conjunction <i>kak</i> ‘as’: second case)	Нет (No)	Иванова , как художника , я совсем не знаю . (I don’t know Ivanov at all , as an artist.)	Иванова как художника я совсем не знаю . (I don’t know Ivanov at all as an artist.)

Table 8: An example of a rule from the dataset with English translation. Additional metadata and other sentences for this rule are omitted for illustrative purposes.

Parameter	value
GPU	A100 80B
num GPUs	1
epochs	3/5
physical batch size	1
batch size	32
learning rate	1e-5/1e-6
max_grad_norm	1.0
optimizer	adafactor
scheduler	triangular
warmup	0.1
weight decay	0.01
precision	fp16
gradient checkpointing	yes

Table 9: Hyperparameters used for 7B/8B language models finetuning.

Parameter	Value
Epochs	3
Batch size	32
Learning rate	1e-5
Optimizer	AdamW
Scheduler	Triangular
Warmup	0.1

Table 10: Hyperparameters of GECTOR encoder training

is the closest example with the same class label and h^- is the closest example with incorrect label. We represent each sentence with up to 3 hidden states of the most probable error positions in it, provided their probability exceeds the threshold θ . When there is no such position, only the most probable position is extracted. If $\mathcal{H}(s)$ is the set of all hidden states corresponding to a sentence s ,

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 меняя слова и знаки препинания, которые и так правильные. *(Dear language model, after "The initial sentence" you'll be given a sentence in Russian which may contain spelling, punctuation, grammatical and speech errors. Print, please, only the correct version of this sentence without giving any comments and highlighting any symbols. Your task is to minimally edit the text, don't change the words and punctuation marks that are already correct.)*

Figure 2: Prompt for correction of Russian text. The English translation is given in brackets.

the distance between two sentences is the minimal distance between its state representations:

$$\rho(s, s') = \min_{h \in \mathcal{H}(s), h' \in \mathcal{H}(s')} \rho(h, h').$$

We collect training triples at the beginning of each epoch. For each sentence we search for its nearest neighbours using approximate nearest neighbour (ANN) search with cosine distance. We implement ANN search using Faiss. After processing all the batches we recalculate the hidden representations and update the vector storage. The hyperparameters of contrastive fine-tuning are given in Table 11.

Parameter	Value
Epochs	10
Batch size	8
Learning rate	1e-5
Optimizer	AdamW
Scheduler	Triangular
Warmup	0.1

Table 11: Hyperparameters of GECTOR encoder training

G Additional results

G.1 Additional results on LORuGEC

Here we evaluate two more models on LORuGEC, repeating the setup of Section 5. We select Llama3-8B-Instruct¹⁶(Meta, 2024) as a medium-size open-source model and GPT4o-2024-05-13(OpenAI, 2023) as a large open-source model. The results are provided in Table 12. The models follow the same pattern as the Qwen2.5-7B and YandexGPT models (see Table 4) with GECTOR+FT being the best few-shot selection method. That means that our approach works both for strong closed-source models and comparably weaker open-source models with limited knowledge of Russian. Interestingly, the GPT4o model almost reaches the level of YandexGPT5-Pro, providing additional evidence that huge language models trained on large amounts of different texts, e.g. educational ones, may not only memorize the rules encountered in these texts, but also apply them to similar language material.

G.2 Additional results on GERA

The comparison of different few-shot selection method on GERA is provided in Table 13.

G.3 Results for English

We evaluate our approach on English, using the development subset of W&I corpus(Bryant et al., 2019), also known as BEA-2019, as our evaluation corpus. We follow the setup of the previous subsection using Qwen2.5-7B Instruct as an open-source model and GPT4o-05-13(OpenAI, 2023) as the closed-source one. The main difference with LORuGEC experiments is the absence of analogous rule-type-annotated corpus for English. Therefore, we cannot readily adapt the contrastive tuning stage. We tried to replace the rule labels with the ERRANT edit types, however, most of the sentences contained several errors of different types. We attempted to train the encoder on the subset of single-error sentences but the approach was not successful.

The generative LLM is finetuned on the W&I corpus training set. For encoder training we utilize a larger cLang-8 corpus (Rothe et al., 2021), corpus parameters are given in Table 14. Note that we don't use BEA-2019 for GECTOR encoder training

¹⁶<https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>

Setup	Llama3-8B-Instruct			GPT4o-2024-05-13		
	P	R	F0.5	P	R	F0.5
zero-shot	24.0	30.3	25.1	65.6	68.6	66.2
1-shot, random	30.1	32.8	30.6	71.8	69.5	71.4
5-shot, random	32.1	30.2	31.7	75.3	70.3	74.3
1-shot, GECTOR	30.5	34.9	31.3	72.8	73.2	72.9
5-shot, GECTOR	37.6	37.7	37.6	76.2	74.8	75.9
1-shot, GECTOR+FT	32.7	36.7	33.4	74.8	75.9	75.0
5-shot, GECTOR+FT	42.7	42.9	42.7	79.6	77.9	79.2
ext. finetuning	39.5	14.7	29.6		NA	
ext.+LORuGEC finetuning	58.6	33.6	51.0		NA	
LORuGEC LORA finetuning	48.8	36.5	45.7		NA	

Table 12: Comparison of different LLMs on the LORuGEC test set in zero-shot, few-shot and finetuning modes. Ext. finetuning refers to training on the concatenation of other Russian GEC corpora. The best metric inside the same approach (e.g., 1-shot) is presented in italics and the best overall metric – in bold.

Setup	Qwen-2.5 7B Instruct			YandexGPT-5 Lite 8B Instruct		
	P	R	F0.5	P	R	F0.5
zero-shot	50.3	43.7	48.8	70.8	53.3	66.5
random, 1-shot	58.6	41.4	54.1	76.8	52.2	70.2
random, 5-shot	59.9	40.2	54.5	73.8	56.0	69.4
gector, 1-shot	58.9	44.0	55.1	77.7	54.8	71.7
gector, 5-shot	65.0	46.6	60.2	75.4	58.6	71.3
gector+FT, 1-shot	58.6	44.0	55.0	76.1	54.8	70.7
gector+FT, 5-shot	62.6	47.9	59.0	74.8	58.8	70.9
finetuning	75.8	45.9	67.1	78.0	59.0	73.3

Table 13: Comparison of different few-shot example selection methods on GERA. The best metric inside the same approach (e.g., 1-shot) is presented in italics and the best overall metric – in bold.

to simulate the case when large in-domain training corpus is not available.

Corpus	Size	Usage
BEA-2019 train	34308	Training
cLang-8	2372119	encoder training
BEA-2019 dev	4384	Testing

Table 14: GEC corpora used for experiments on English.

Results on BEA-2019 development set are available in Table 15. Here we use ERRANT-3.0 (Bryant et al., 2017) to obtain evaluation metrics. Comparing them to the results of the previous subsection, we observe the following:

1. Again, retriever-based selection of demonstration samples produces small but stable improvements. These improvements are stable across models and the number of few-shot examples.

2. However, the difference with baseline is smaller than for LORuGEC. In particular, the achieved enhancements are not sufficient to reach the level of finetuned model. We hypothesize that the reason for this is the larger size of training corpus in case of English that allows the finetuned model to achieve larger improvements over the zero-shot version.

H Implementation of our Approach

We present several responses of the YandexGPT-lite model to the sentences from LORuGEC, which were obtained using different kinds of 1-shot demonstration: random, GECTOR-based or finetuned GECTOR-based, see Figures 3 and 4.

Method	few-shot method	k	Qwen2.5-7B			GPT4o-05-13		
Zero-shot	–	0	36.2	43.4	37.5	34.2	52.6	36.8
few-shot	random	1	37.9	42.8	38.8	35.7	51.5	38.0
few-shot	random	5	38.4	43.6	39.4	37.2	49.0	39.1
few-shot	GECTOR	1	39.1	44.4	40.1	37.2	52.0	39.4
few-shot	GECTOR	5	40.0	46.0	41.1	39.4	51.5	41.4
LLM external	–	0	53.4	48.8	52.4	NA	NA	NA

Table 15: Comparison of different few-shot example selection methods on BEA development set.

Source sentence	Врачи буквально обомлели оттого , что увидели.		The doctors literally were stunned by what they saw.
Target sentence	Врачи буквально обомлели от того , что увидели.		(<i>om togo</i> ‘by what’ in this context must be written separately.)
Random 1-shot demonstration			
Исходное предложение:	Куда запропастилась моя подруженька ?		The initial sentence: Where is my girl-friend ?
Исправленное предложение:	Куда запропастилась моя подруженька ?		The correct sentence: Where is my girl-friend ?
			(The word <i>подруженька</i> ‘girl-friend’ contained a spelling error in the initial sentence.)
LLM Response	Врачи буквально обомлели оттого , что увидели.		(The model did not correct the sentence.)
GECTOR 1-shot demonstration			
Исходное предложение:	Родители были счастливы от того , что Владик передумал.		The initial sentence: Parents were happy be-cause Vladik changed his mind.
Исправленное предложение:	Родители были счастливы оттого , что Владик передумал.		The correct sentence: Parents were happy be-cause Vladik changed his mind.
			(The word <i>оттого</i> ‘because’ must be written in one word in this context.)
LLM Response	Врачи буквально обомлели оттого , что увидели.		(The model did not correct the sentence.)
GECTOR-FT 1-shot demonstration			
Исходное предложение:	Я расстроился оттого , что увидел.		The initial sentence: I was upset by what I saw.
Исправленное предложение:	Я расстроился от того , что увидел.		The correct sentence: I was upset by what I saw.
			(<i>om togo</i> ‘by what’ in this context must be written separately.)
LLM Response	Врачи буквально обомлели от того , что увидели .		(The model successfully corrected the sentence.)
Conclusion	Only the finetuned GECTOR was able to obtain the sentence with the same preposition and pronoun <i>om togo</i> ‘by what’ and the same context in which it must be written separately, not in one word, as opposed to the demonstration chosen by the basic GECTOR. Random selection had a spelling error in it which did not at all resemble the target error. Consequently, LLM was able to correct the sentence only with the GECTOR-FT demonstration.		

Figure 3: Implementation of our approach on the sentence from LORuGEC using YandexGPT5-Lite model. Incorrect parts are marked with red, corrected parts are marked with green for illustrative purposes. There were no highlights in experiments. In the second column we also present English translations of the sentence and demonstrations as well as comments to them in brackets. The same holds for Figure 4.

Source sentence	Кажется, это сон, и я сплю.	It seems, it's a dream, and I'm dreaming.
Target sentence	Кажется, это сон, и я сплю.	(Кажется'It seems' is a part of the sentence that is related to both clauses <i>это сон</i> 'it's a dream' and <i>я сплю</i> 'I'm dreaming' which are connected by the conjunction <i>и</i> 'and', that is why there must not be any commas between the clauses before the conjunction.)
Random 1-shot demonstration		
Исходное предложение: Вы можете подумать, что вас это некасается и даже рассмеяться.. Исправленное предложение: Вы можете подумать, что вас это не касается и даже рассмеяться..		The initial sentence: You may think, that it does not concern you and even laugh.. The correct sentence: You may think, that it does not concern you and even laugh.. (In Russian negative particle <i>не</i> must be written separately from the verb, so <i>не касается</i> 'does not concern' must not be written in one word.)
LLM Response	Кажется, это сон, и я сплю.	(The model did not correct the sentence.)
GECTOR 1-shot demonstration		
Исходное предложение: В это время раскрылась дверь поместья, и вышел начальник дозора. Исправленное предложение: В это время раскрылась дверь поместья, и вышел начальник дозора.		The initial sentence: At that moment, the door of the manor opened, and the head of the watch came out. The correct sentence: At that moment, the door of the manor opened, and the head of the watch came out. (<i>В это время</i> 'at that moment' denotes the time for both the opening of the door (<i>раскрылась дверь поместья</i>) and the arrival of the head of the watch (<i>вышел начальник дозора</i>), so there should not be any commas before the conjunction <i>и</i> 'and' which connects these two clauses.)
LLM Response	Кажется, это сон, и я сплю.	(The model successfully corrected the sentence.)
GECTOR-FT 1-shot demonstration		
Исходное предложение: Самгин понимал, что говорит плохо, и что слова его не доходят до неё. Исправленное предложение: Самгин понимал, что говорит плохо, и что слова его не доходят до неё.		The initial sentence: Samgin knew that he was speaking badly, and that his words were not reaching her. The correct sentence: Samgin knew that he was speaking badly, and that his words were not reaching her. (<i>Самгин понимал</i> 'Samgin knew' about both facts: that he was speaking badly (<i>что говорит плохо</i>) and that his words were not reaching her (<i>что слова его не доходят до неё</i>), so there must be no comma before the conjunction <i>и</i> 'and' that connects these clauses.)
LLM Response	Кажется, это сон, и я сплю.	(The model successfully corrected the sentence)
Conclusion	Both GECTOR-based models selected demonstrations that follow the punctuation pattern of the source sentence. These demonstrations allowed the LLM to effectively correct the sentence, unlike the randomly selected sentence which had to do with incorrect spelling.	

Figure 4: Implementation of our approach on another sentence from LORuGEC.

Explaining Holistic Essay Scores in Comparative Judgment Assessments by Predicting Scores on Rubrics

Michiel De Vrindt^{1ac} Anaïs Tack^{1ad} Renske Bouwer^{2b}

Wim Van den Noortgate^{1ac} Marije Lesterhuis^{3e}

¹ KU Leuven ^a imec research group itec ^b Institute for Language Sciences

^c Faculty of Psychology and Educational Sciences ² Utrecht University ³ UMC Utrecht

^d Faculty of Arts ^e Center for Research and Development of Health Professions Education

Abstract

Comparative judgment (CJ) is an assessment method in which multiple assessors determine the holistic quality of essays through pairwise comparisons. While CJ is recognized for generating reliable and valid scores, it falls short in providing transparency about the specific quality aspects these holistic scores represent. Our study addresses this limitation by predicting scores on a set of rubrics that measure text quality, thereby explaining the holistic scores derived from CJ. We developed feature-based machine learning models that leveraged complexity and genre features extracted from a collection of Dutch essays. We evaluated the predictability of rubric scores for text quality based on linguistic features. Subsequently, we evaluated the validity of the predicted rubric scores by examining their ability to explain the holistic scores derived from CJ. Our findings indicate that feature-based prediction models can predict relevant rubric scores moderately well. Furthermore, the predictions can be used to explain holistic scores from CJ, despite certain biases. This automated approach to explain holistic quality scores from CJ can enhance the transparency of CJ assessments and simplify the evaluation of their validity.

1 Introduction

Comparative judgment (CJ) is a widely used method for educational assessments, particularly for evaluating writing quality of essays (Baniya et al., 2019; Steedle and Ferrara, 2016; van Daal et al., 2016). In CJ, assessors repeatedly compare (different) pairs of essays and determine which one is superior in quality each time. Then, the Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 1959), relates the probability of one essay being preferred over another to the quality scores of the essays that are compared. Based on the judgments of assessors, the quality scores of essays are estimated.

CJ offers several advantages over traditional rubric-based assessments. Firstly, it allows assessors to use their professional expertise and intuition without strictly adhering to predetermined rubrics, making CJ a more natural assessment method (Bloxham, 2009; Laming, 2003). Assessors may have different conceptualizations of quality; some prioritize essay argumentation and organization, while others focus on language conventions (Lesterhuis et al., 2022). Even when assessors focus on different aspects, van Daal et al. (2016) found that their pairwise comparisons still reflected construct-relevant aspects of writing quality. Secondly, since CJ incorporates multiple judgments from various assessors, the resulting essay quality scores are generally reliable and valid, reflecting a consensus among the assessors (Lesterhuis et al., 2022; Verhavert et al., 2019; van Daal et al., 2016). Although CJ is a valid and reliable assessment method, the holistic scores it produces lack transparency regarding their specific meaning. Since judgments are made holistically, the assessors' decision-making process remains unclear. Assessors can provide feedback while making judgments, but this takes more time and may shift their focus from the overall quality of essays to specific analytic criteria (Verhavert et al., 2019). Finding a new way to explain holistic scores is therefore crucial for making CJ assessments more transparent and can also serve as a form of feedback.

In this study, we investigate the use of feature-based prediction models to explain holistic quality scores from CJ, with the goal of enhancing their transparency.

Our research addresses the following questions:

1. How reliably can scores on a set of rubrics measuring text quality be predicted based on linguistic features of essay texts?
2. To what extent do these predicted rubric scores accurately reflect the holistic quality

scores of essays obtained with CJ?

Our study comprised two phases. First, we conducted a machine learning experiment to assess how well rubric scores could be predicted from linguistic features of Dutch essays. Second, we performed a regression analysis to evaluate the validity of the predicted scores in explaining the holistic scores obtained with CJ.

2 Background

2.1 Comparative Judgment Assessments

CJ functions as an alternative assessment method to rubric scoring and has been shown to produce reliable and valid scores (Verhavert et al., 2019; Lesterhuis et al., 2022; van Daal et al., 2016; Heldsinger and Humphry, 2010). While primarily known for assessing essay quality, CJ has also been effectively used for various other types of assessments. These include evaluating conceptual understanding (Jones et al., 2019), mathematical problem-solving skills (Jones and Inglis, 2015), design portfolios (Newhouse, 2014), formative assessments (Potter et al., 2017; Bartholomew et al., 2019), and comparing assessment standards across examination boards (Bramley, 2007; D'Arcy, 1997).

Generally, CJ assessments are conducted by iterating through three key steps. In the first step, a pair of essays is chosen and assigned to one of several assessors. In the second step, the assessor compares the two essays and determines which demonstrates higher quality. This relative assessment approach is considered more intuitive than absolute assessments, such as rubric-based scoring. As Laming (2003) noted, all judgments inherently involve comparing one entity to another, and CJ explicitly makes use of this principle. In the third step, the BTL model is applied to link the outcomes of all pairwise comparisons to a quality scale (Bradley and Terry, 1952; Luce, 1959). The BTL model relates the probability of one essay being favored over another to the difference in their quality scores, expressed as logit values. Specifically, this probability is determined by the sigmoid function of the quality score difference: the greater the quality score of the first essay relative to the second, the higher the probability it will win the comparison. The quality scores in BTL model are continuously updated based on the judgments that assessors make. The assessment concludes once a sufficient number of judgments have been collected, typically requiring each essay to be compared 10

to 14 times to ensure reliable quality scores. Ultimately, the holistic scores derived from CJ are both reliable and valid, as they stem from numerous pairwise comparisons by multiple assessors (van Daal et al., 2016; Lesterhuis et al., 2022).

However, when the CJ assessment is completed, the resulting quality scores for essays lack clarity regarding what they represent. The issue stems from the scores being based on holistic pairwise comparisons by assessors (Steedle and Ferrara, 2016; Kelly et al., 2022), a method that, while reliable and valid, lacks the transparency offered by detailed rubric-based marking (Jonsson, 2014; Mortier et al., 2015). As a result of this ambiguity, the feedback function of the scores to students is hindered, and the validation of the assessors' judgments is complicated. Even though assessors can provide feedback comments when making judgments, doing so extensively would be time-consuming and reduce assessment efficiency. Furthermore, writing numerous comments to individual essays can lead assessors to adopt a more analytical approach (Verhavert et al., 2019), which conflicts with the holistic nature of CJ assessments (van Daal et al., 2016). Hence, there is a need to enhance the transparency of the holistic scores obtained with CJ without requiring more effort from assessors. To achieve this, we propose automatically predicting the scores on rubrics to explain the holistic scores derived from CJ. This prediction task is similar to that of automated essay scoring (AES).

2.2 Automated Essay Scoring

With recent advancements in NLP methods, AES for summative assessments and automatic writing evaluation (AWE) for formative assessments have received increasing attention. Initially, systems relied on analyzing hand-crafted linguistic features from essay texts to predict scores (Ke and Ng, 2019). However, following the ASAP Kaggle competition organized by the Hewlett Foundation (Hamner et al., 2012), deep learning models have gained prominence in this domain, often surpassing traditional feature-based prediction models in terms of agreement with human scoring (Dong et al., 2017; Taghipour and Ng, 2016; Wang et al., 2022). Despite these advances, practical AES and AWE systems, such as PEG (Dikli, 2006) and e-rater (Burstein et al., 2004), continue to rely heavily on hand-crafted linguistic features due to the need for transparency. Especially, text complexity features

such as syntactical complexity and lexical diversity have been shown to have a large predictive power for the writing quality of essays (McNamara et al., 2010). For instance, for English-written essays, the Coh-Matrix (Graesser et al., 2004) and SALAT toolsets (Crossley et al., 2023) are commonly used to extract complex linguistic features for AES (McNamara et al., 2015; Li and Liu, 2017; Latifi and Gierl, 2021; Kumar and Boulanger, 2020).

While feature-based AES models and AWE systems provide more transparency, the linguistic features themselves, such as complexity and cohesion features, can still be hard to interpret and may lack pedagogical clarity for students and teachers. As Deane (2013b) stated, using linguistic features as proxies for writing quality is neither transparent nor instructional for students. Additionally, Crossley (2020) noted that extensive knowledge is required in order to use linguistic features effectively.

As assessors mostly consider higher-order aspects of writing when making pairwise comparisons, such as structure and argumentation (Lesterhuis et al., 2022), linguistic features would not provide the desired transparency about the holistic quality scores. Therefore, in this study, we chose to explain the holistic scores based on more instructional rubrics that measure specific aspects of writing quality. We predicted scores across these rubrics based on linguistic features extracted from essays. The automated scoring task of predicting scores across multiple rubrics is also referred to as 'multi-trait' scoring within AES literature (He et al., 2022; Do et al., 2023; Mathias and Bhattacharyya, 2020).

3 Method

3.1 Data

We used data previously collected by Coertjens et al. (2017). The dataset, detailed in Table 1, included a total of 104 argumentative essays in Dutch written by students from secondary education. The students could choose to write an essay on one of the topics: (1) having children, (2) organ donation, and (3) stress experienced by students. Despite the differences in topics, the essays were quite similar in terms of the assessed competence: the ability to effectively integrate source material within argumentative writing. This allowed us to combine the essays from different assignments into one dataset for model training. We selected this data because it is the only CJ dataset where essays are labeled

with both holistic and rubric scores.¹

Assignment	Essays	Tokens	Tokens/Essay
	<i>N</i>		<i>M</i> ± <i>SD</i>
1. Children	34	11167	328 (± 92)
2. Organ	35	11358	293 (± 93)
3. Stress	35	11859	304 (± 97)

Table 1: Overview of the argumentative writing assignment gathered by Coertjens et al. (2017). Tokenization was performed using the Dutch n1_core_news_sm model from spaCy (Explosion, 2023).

3.1.1 Holistic Scores

Coertjens et al. (2017) used CJ to obtain holistic scores of essay quality. During the assessment, 40 assessors made pairwise comparisons and each essay was compared 25 times. The assessors were asked which essay in this pair is better in terms of argumentation. This assessment resulted in holistic scores with a reliability of 0.87, as measured by scale separation reliability (Verhavert et al., 2018).

3.1.2 Scores on Rubrics

Coertjens et al. (2017) asked 18 assessors to evaluate the same essays using a rubric set designed to measure 20 aspects of text quality. These were different assessors from those who scored the essays holistically with CJ. These aspects were grouped into four main components: structure (6 rubrics), content (7 rubrics), argumentation (4 rubrics), and language conventions (3 rubrics). The rubrics, originally developed and validated by Rijlaarsdam et al. (1994), were adapted by Coertjens et al. (2017) for this particular assignment on argumentative writing. According to Coertjens et al. (2017), the intraclass correlation coefficient was 0.85 after five different assessors assessed each essay. For an overview and description of all rubrics, refer to Appendix A.

3.2 Features

To extract linguistic features from the essays, we used T-Scan (Maat et al., 2014) because Dascalu et al. (2017) previously demonstrated that its features have strong predictive power for automated essay scoring. Using the T-Scan API (v0.10), we extracted 476 document-level features related to lexical complexity, sentence complexity, referential cohesion, lexical diversity, lexical semantics,

¹The data gathered by Lesterhuis et al. (2022), for example, includes a superset of the essays used by Coertjens et al. (2017), but it does not include any rubric scores.

and personal style. For details on the T-Scan configuration, see Appendix B.

Since T-Scan does not account for spelling and grammatical errors, we also used the LanguageTool package (v2.8.1) in Python to count the number of language mistakes in each essay. We normalized these counts by dividing them by the total number of tokens per essay (see Table 1).

3.3 Models

We trained regression models using the extracted features to predict scores of essay quality. This involved training multiple single-target regression models, with each model predicting either the holistic score or one of the rubric scores.

We experimented with five machine learning models for the regression tasks: **Lasso Regression**, **ElasticNet**, **Random Forest**, and **XGBoost** using scikit-learn 1.4.0 (Pedregosa et al., 2011), and **LightGBM** 4.6.0 (Ke et al., 2017) in Python 3.9.12. We applied min-max normalization to each input feature from T-Scan as well as the rubric scores. Before training the models, we excluded features from the training set that had a low Pearson correlation with the target. A correlation threshold of 0.12 was chosen based on Lovakov and Agadullina (2021).

For each individual model, we ran hyperparameter tuning on the training set using a randomized search strategy with up to 100 iterations. The optimal hyperparameters were selected based on the lowest mean absolute error (MAE) between the predicted and actual rubric scores across 20 folds.

3.4 Evaluation

To optimize the prediction performance and avoid overfitting on a small dataset, we performed leave-one-out cross-validation (LOOCV). This involved leaving out one essay for evaluation and training a model on all remaining essays, repeating the process for each essay in the dataset. The hyperparameter tuning with 20-fold cross-validation, as mentioned before, was conducted on the training data for each run of LOOCV. Refer to Appendix C for an overview of the selected hyperparameters.

3.4.1 Metrics

Using the optimal model and hyperparameters for each rubric, we evaluated the predictions of the rubric scores with various metrics on all left-out essays during LOOCV. We used the **squared Pearson correlation coefficient** (R^2) between predicted

and actual scores. R^2 is a measure of score reliability in classical test theory (Brennan, 2010) and is often used to measure the reliability of quality scores estimated from CJ relative to true scores (Verhavert et al., 2018).

Additionally, we used the **quadratic weighted kappa** (QWK) (Cohen, 1968) and the **mean absolute error** (MAE), two commonly used metrics in AES research (Ramesh and Sanampudi, 2022). QWK is a metric based on Cohen’s kappa that measures agreement between predicted and human-given scores, penalizing more divergent predictions. A score of 1 indicates perfect agreement, while -1 indicates perfect disagreement.

3.4.2 Predictive Power

To validate whether the predicted rubric scores accurately measure the assessed writing quality with CJ, we evaluated their predictive power for the holistic scores using linear regression models. Using the statsmodels package (0.13.2) (Seabold and Perktold, 2010), we constructed two regression models measuring the effects of rubric scores on holistic scores from CJ:

- **Regression Model 1** uses the rubric scores predicted by the model (see Section 3.3) as covariates and holistic quality scores from CJ as outcomes.
- **Regression Model 2** uses the rubric scores given by assessors as covariates and holistic quality scores from CJ as outcomes.

We compared their goodness-of-fit using the Akaike information criterion (AIC) and Bayesian information criterion (BIC), as well as the explained variance (R^2) of the holistic scores. In the context of statistical modeling, R^2 measures the proportion of variance in holistic scores that is explained by the (predicted) scores on rubrics. As it can be inflated by adding many covariates, we adjusted R^2 for the number of covariates.

We expected that Model 2, which uses human-assigned rubric scores, would fit the holistic scores better than Model 1, which uses predicted rubric scores. Therefore, we aimed to evaluate how closely the fit of Model 1 approximates that of Model 2.

To further investigate any potential biases in the effects (i.e., coefficients) of the predicted rubric scores on the holistic scores, we compared the coefficients of Model 1 with those of Model 2, along

with their confidence intervals. We calculated Student's t-tests to evaluate whether the coefficients are significantly different from zero. We omitted intercepts for both models to make the coefficients comparable, and inverted the normalization of all variables to their original scales.

4 Results

4.1 Predicting Scores on Rubrics

Table 2 shows the performance of the rubric scoring models evaluated using LOOCV. ElasticNet consistently demonstrated the best performance in predicting most rubric scores (9/20 rubrics), followed by XGBoost (5/20 rubrics), Lasso (3/20 rubrics), RandomForest (2/20 rubrics), and LightGBM (1/20 rubrics). Overall, model performance was moderately effective across all evaluation metrics, which is expected given the small sample size.

Performance varied notably across the different rubrics. Among all rubrics pertaining to essay structure, predictions showed the highest reliability (R^2) for Construction, Relationships, and Continuity, and the highest agreement (QWK) with human scores for Title compared to other rubrics. This suggests that the linguistic features employed in the models can capture markers of essay organization, despite the overall moderate prediction performance.

Among rubrics pertaining to essay content, scores on References and Citations were the most accurately predicted, but generally, the predictions were only moderate or poor. The scores for Introduction, Persuasion, Reader Focus, Reader Engagement, and Conclusion were comparatively worse. This suggests that the linguistic features employed in the models can capture some markers of essay content, albeit with limited accuracy.

Among rubrics pertaining to argumentation, Support and Relevance showed the best prediction performance, whereas Indication and Reference Cohesion Relationships were less accurately predicted. This shows the model's ability to capture, to a certain extent, the argumentative writing level related to how sources were integrated and used to support claims.

Generally, the rubrics pertaining to language were poorly predicted. Among these rubrics, the scores for Style were most accurately predicted, while predictions for Grammar and Spelling, and Punctuation were comparatively worse. Hence, the assessors' scoring of language conventions differed

from the model predictions.

4.2 Explaining Holistic Scores with Predictions

After evaluating the predicted scores on rubrics, we examined how well these scores can explain the holistic scores obtained through CJ. Table 3 shows the goodness-of-fit of the two models. As expected, Model 2 provided a better fit for the holistic scores than Model 1, as evidenced by smaller AIC and BIC values. Additionally, Model 2 explained 12% more of the variance in holistic scores compared to Model 1, as indicated by the higher R^2 . This difference was even more pronounced when considering the adjusted R^2 values. Although the predicted rubric scores explained the holistic scores reasonably well (Model 1), 40% of the variance in holistic scores still remained unaccounted for. In contrast, the rubric scores given by assessors (Model 2) had greater predictive power for the holistic scores. However, even in Model 2, 28% of the variance in holistic scores remained unexplained, indicating a difference in how assessors score essays with rubrics versus holistically using CJ.

Figure 1 illustrates the biases in the coefficients of the predicted rubrics on the holistic scores (Model 1) with respect to the coefficients of the assessor-assigned rubric scores (Model 2). Overall, the coefficients for the rubrics in both models were similar in magnitude and direction, which supports the validity of the predicted rubric scores. However, Model 1 exhibited some systematic biases, as it tends to overshoot the magnitude of the coefficients. Specifically, the coefficients for rubrics pertaining to essay structure showed upward biases, except for Subtopic. Conversely, the coefficients for rubrics related to content displayed downward biases, with the exception of Introduction and Citations.

The most significant coefficients in Model 2 were Relationships, References, Conclusion, and Grammar and Spelling. Except for Grammar and Spelling, these rubrics were all reasonably well approximated by Model 1, as their coefficients were similar and their confidence intervals overlapped. This shows that predicted rubric scores accurately explained the holistic scores based on the most important rubrics scored by assessors. However, of these rubrics, only the coefficients for Relationships, and Grammar and Spelling were significantly different from zero in Model 1. This can be attributed to the wider confidence intervals of Model

Aspect	Rubric	Best Model	R ²	QWK	MAE
Structure	Title	LightGBM	0.23	0.50	0.85
Structure	Construction	XGBoost	0.31	0.41	0.80
Structure	Layout	RandomForest	0.24	0.41	0.91
Structure	Subtopic	XGBoost	0.25	0.41	0.74
Structure	Relationships	ElasticNet	0.30	0.46	0.76
Structure	Continuity	RandomForest	0.34	0.45	0.48
Content	Introduction	ElasticNet	0.27	0.45	0.54
Content	Persuasion	ElasticNet	0.31	0.45	0.53
Content	References	Lasso	0.48	0.60	0.58
Content	Citations	XGBoost	0.53	0.64	0.54
Content	Reader Focus	ElasticNet	0.30	0.40	0.39
Content	Reader Engagement	ElasticNet	0.38	0.42	0.38
Content	Conclusion	XGBoost	0.28	0.46	0.66
Argumentation	Support	Lasso	0.51	0.59	0.35
Argumentation	Relevance	ElasticNet	0.46	0.54	0.36
Argumentation	Indication	ElasticNet	0.22	0.30	0.59
Argumentation	Reference Cohesion Relationships	XGBoost	0.28	0.37	0.47
Language	Grammar and Spelling	Lasso	0.25	0.43	0.48
Language	Punctuation	ElasticNet	0.27	0.35	0.49
Language	Style	ElasticNet	0.38	0.45	0.34

Table 2: Evaluation of LOOCV results for predicting scores on rubrics measuring aspects of text quality, with hyperparameter search conducted for each run.

Reg. Model	AIC	BIC	R ²	Adj. R ²
1	427.60	483.20	0.60	0.50
2	390.20	445.80	0.72	0.65

Table 3: Comparison of model fit for linear regression models with holistic scores from CJ as the outcome, where Regression Model 1 used predicted scores on rubrics as covariates and Regression Model 2 used scores on rubrics provided by assessors as covariates.

1 compared to Model 2.

When examining rubrics that were most accurately predicted (see Table 2), it is clear that their impact on holistic scores (Model 1) closely resembled that of scores given by assessors (Model 2). This similarity was evident for Relationships, References, and Citations, where Model 1’s coefficients aligned with those of Model 2, and their confidence intervals significantly overlapped. Although Support and Relevance were also predicted more accurately, their coefficients exhibited great uncertainty, as indicated by wider confidence intervals compared to Model 2, especially for Support. This indicates a potential lack of validity of the predicted rubrics related to argumentation.

Conversely, when the rubric was predicted more inaccurately, their coefficients showed more bias. This was observed for Layout and Reader Focus, as their rubric scores were poorly predicted and their coefficients overestimated. Similarly, the rubrics related to language were inaccurately predicted when compared to other rubrics, resulting in coefficients with large biases. More specifically, the importance of sound Grammar and Spelling was vastly overestimated using the predicted rubric scores, as these scores are much higher than when rubric scores were given by teachers. Conversely, the importance of correct Punctuation and Style was overly negative when using the predictions compared to the scores given by teachers. This shows that teachers score rubrics differently and that these scores contribute less to the holistic scores than when using predictions.

5 Discussion

The lack of transparency in holistic scores obtained through CJ limits the practical application of this assessment method (Steedle and Ferrara, 2016). Our analysis reveals that rubric scores can be predicted moderately well in terms of reliability and

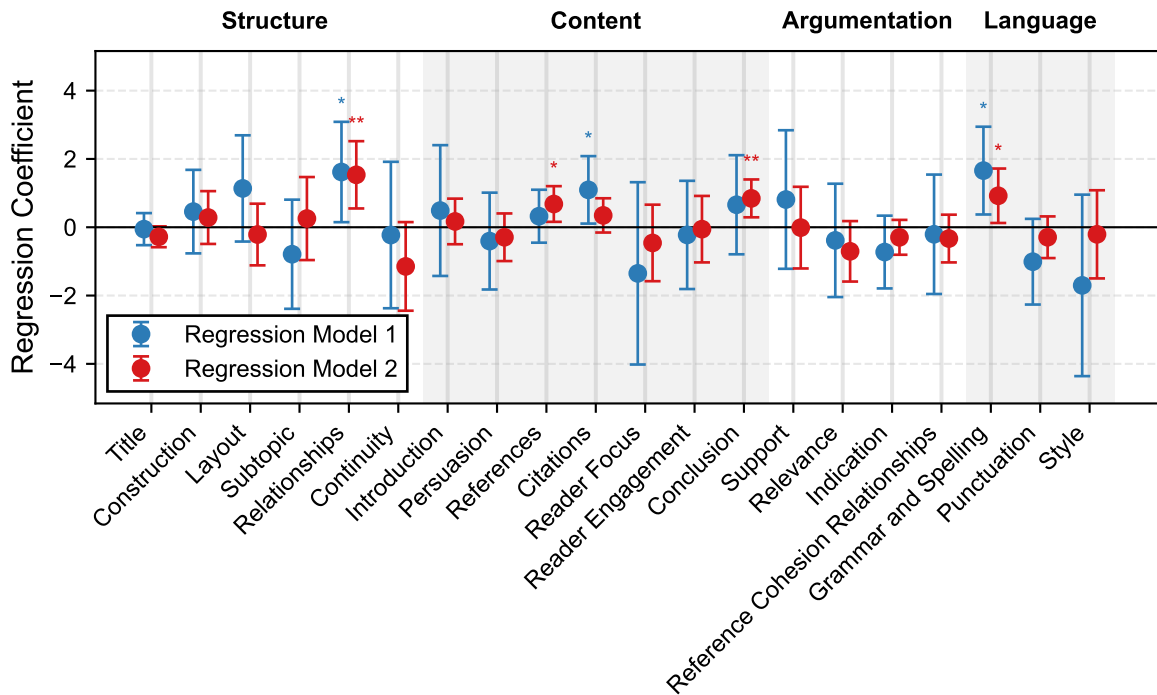


Figure 1: Comparison of regression coefficients of Model 1 and Model 2 for all rubric scores with 95% confidence intervals, where significance is denoted as * $p < 0.05$, ** at $p < 0.01$, and *** at $p < 0.001$.

agreement with assessor-assigned scores, with the best predictions for the rubrics relevant to the assignment on argumentative writing. However, not all rubrics can be predicted well, which can be due to the limited size of the training set.

Furthermore, the predicted rubric scores have explanatory power for holistic scores derived from CJ, thus showing potential for an automated approach to provide more transparency. However, it is unlikely that the rubric scores can fully explain the differences in holistic scores from CJ, as even the assessor-assigned scores do not fully explain them. This is not surprising, given that holistic scoring with CJ involves relative assessments while rubric-based scoring requires absolute assessments. While both assessment approaches are reliable on their own, they may yield slightly different results (Coertjens et al., 2017).

Generally, the relationship between predicted rubrics and holistic scores is similar to that of rubric scores given by assessors, which supports the validity of the predictions. Most importantly, we find that the validity of the predicted rubric scores depends on their predictability from linguistic features. The rubrics that demonstrate better predictability can also explain the holistic scores with minimal bias. This shows that predicted scores

on assignment-relevant rubrics can explain, in part, the holistic scores from CJ.

However, there are notable differences between human scoring and automated scoring (Ben-Simon and Bennett, 2007). Previously, Ramineni and Williamson (2018) found that the e-rater AES system often overvalues organization while undervaluing content. Our findings generally support this, as most structure-related rubrics exert an overly positive influence on holistic scores when predicted compared to when scored by assessors, whereas most content-related rubrics show an overly negative influence when predicted compared to when assessed by humans.

While certain argumentation-related rubrics can be predicted comparatively well, their influence on the holistic scores shows more uncertainty. As Attali (2007) stated, agreement between human and automated scores does not directly imply that the scores are valid. This discrepancy may be attributed to the inherent difficulty in measuring the quality of argumentation based on linguistic features (Deane, 2013a). Hence, more elaborate linguistic features based on argument mining may be needed for more valid predictions of argumentative-related rubrics.

Additionally, there is a difference between how

language-related rubrics are predicted and how they are assessed. These rubrics prove difficult to predict based on linguistic features, and their effect on holistic quality scores is biased, either undervalued or overvalued. Previously, [Ramineni and Williamson \(2018\)](#) noted that the e-rater AES system severely undervalues grammatical mistakes for essay scoring. Further analysis is needed to uncover the potential causes of this bias. However, for CJ assessments, language conventions are generally less important when making pairwise comparisons of essays ([Lesterhuis et al., 2022](#)).

6 Conclusion

To address the lack of transparency of holistic scores from CJ assessments, we used feature-based models to predict scores on a set of rubrics that explain the holistic scores. Based on linguistic features extracted with T-Scan, rubric scores of Dutch essays were predicted with moderate success. However, we found that the most relevant rubrics were predicted more reliably compared to other rubrics. Furthermore, we noted that these predicted scores on rubrics can explain holistic scores from CJ in a manner comparable to the assessor-assigned rubric scores.

While the automated predictions of rubrics offer more transparency regarding the meaning of holistic scores, they do differ from human assessor scores in certain respects. For instance, structure-related rubrics were slightly overvalued, content-related rubrics were slightly undervalued, and the effect of argumentation-related rubrics showed more uncertainty. Additionally, predictions for language convention rubrics diverged notably from assessor-given scores.

Despite some discrepancies in how predicted rubric scores explain holistic scores compared to rubrics scored by assessors, they generally aligned well for the most important rubrics and demonstrate predictive power. This suggests that predicting scores on rubrics can help explain the holistic scores obtained with CJ. However, their acceptance and effectiveness as feedback for students require future research.

Limitations

Even though the scores given by assessors are reliable and valid, the size of the available dataset used for training is rather limited, which could explain the moderate prediction performance. We expect

that increasing the dataset would improve prediction performance and, therefore, produce scores on rubrics that better explain the holistic scores from CJ. With a larger training set, it would be possible to determine the best-performing hyperparameters and models for each rubric for all essays, rather than per fold as was done in this study. This approach would enhance the generalizability of the models' performance.

Future research could, for example, leverage the larger ASAP dataset, which contains English essays scored on rubrics such as ideas, organization, style, and conventions, for different writing genres ([Hammer et al., 2012](#)). However, the granularity of features is higher in this dataset, which would provide less specific explanations than in the current study.

In case only a small set of rubric-scored texts is available, it may be more suitable to extract rubric scores using language models, which can capture complex textual features. Large Language Models (LLMs) have been applied for this purpose through fine-tuning ([Do et al., 2024](#)) or zero-shot prompting ([Lee et al., 2024](#)). However, relying on LLMs would make it less transparent how the predicted scores are derived compared to using hand-crafted linguistic features, as in this study.

To better understand the validity of predicted scores on rubrics in relation to how they are predicted, future research could examine the most important features for making these predictions. This is important as feature-based approaches for AES do not capture meaning directly. Previously, it has been noted that essay length is highly influential for AES models, and it is advised that its effect be studied by controlling for it ([Chodorow and Burstein, 2004](#)). Text length is especially influential for structure, content, and argumentation, and less so for language ([Enright and Quinlan, 2010](#); [Barkaoui and Woodworth, 2023](#)). While essay length is a valid factor that human assessors also consider, its disproportionate influence can be problematic. Moreover, analyzing the importance of linguistic features in the model's predictions can help clarify why language-related rubrics were predicted with low reliability and validity. This can be achieved by evaluating whether features extracted with LanguageTool significantly contributed to the prediction of language-related rubric scores.

Additionally, interpreting the coefficients of a linear regression model as effects of rubric scores on holistic scores requires caution. Rubrics measuring

aspects of text quality tend to be highly correlated, which raises the potential for multicollinearity. To gain a more accurate understanding of how rubric scores influence the holistic scores, we recommend employing regularization techniques or incorporating interaction effects into the model. These approaches can help mitigate the challenges posed by correlated predictors and provide clearer insights into the effects of rubric scores on holistic scores from CJ. Furthermore, the validity of the predicted scores on rubrics for explaining holistic scores is contingent on the assessment context. In second language (L2) writing, for instance, criteria such as language accuracy may be weighted more heavily than they were in the L1 context of this study. Therefore, future research is essential to validate these findings across different assessment contexts.

Acknowledgments

This research was supported by a Baekeland mandate (HBC.2022.0164) granted by the Flanders Innovation & Entrepreneurship (VLAIO) to the first author, in cooperation with the company Comproved (D-Pac BV).

References

- Yigal Attali. 2007. [Construct validity of e-rater® in scoring toefl® essays](#). *ETS Research Report Series*, 2007(1):i–22.
- Sweta Baniya, Nathan Mentzer, Scott R Bartholomew, Amelia Chesley, Cameron Moon, and Derek Sherman. 2019. [Using adaptive comparative judgment in writing assessment](#). *The Journal of Technology Studies*, 45(1):24–35.
- Khaled Barkaoui and Johanathan Woodworth. 2023. [An exploratory study of the construct measured by automated writing scores across task types and test occasions](#). *Studies in Language Assessment*, 12(1):26.
- Scott R Bartholomew, Greg J Strimel, and Emily Yoshikawa. 2019. [Using adaptive comparative judgment for student formative feedback and learning during a middle school design project](#). *International Journal of Technology and Design Education*, 29:363–385.
- Anat Ben-Simon and Randy Elliot Bennett. 2007. Toward more substantively meaningful automated essay scoring. *The Journal of Technology, Learning and Assessment*, 6(1).
- Sue Bloxham. 2009. [Marking and moderation in the UK: false assumptions and wasted resources](#). *Assessment & Evaluation in Higher Education*, 34(2):209–220.
- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Tom Bramley. 2007. Paired comparison methods. In Peter Newton, J Baird, H Goldstein, H Patrick, and P Tymms, editors, *Techniques for monitoring the comparability of examination standards*, pages 246–300. Qualifications and Curriculum Authority London, London, United Kingdom.
- Robert L Brennan. 2010. [Generalizability theory and classical test theory](#). *Applied Measurement in Education*, 24(1):1–21.
- Jill Burstein, Martin Chodorow, and Claudia Leacock. 2004. Automated essay evaluation: The criterion online writing service. *AI Magazine*, 25(3):27–27.
- Martin Chodorow and Jill Burstein. 2004. Beyond essay length: evaluating e-rater®’s performance on toefl® essays. *ETS Research Report Series*, 2004(1):i–38.
- Liesje Coertjens, Marije Lesterhuis, San Verhavert, Roos Van Gasse, and Sven De Maeyer. 2017. Teksten beoordelen met criterialijsten of via paarsgewijze vergelijking: een afweging van betrouwbaarheid en tijdsinvestering. *Pedagogische studiën*, 94(4):283–303.
- Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213.
- Scott A Crossley. 2020. Linguistic features in writing quality and development: An overview. *Journal of Writing Research*, 11(3):415–443.
- Scott A. Crossley, Kristopher Kyle, and Danielle S. McNamara. 2023. [Suite of automatic linguistic analysis tools \(salat\)](#). <https://www.linguisticanalysistools.org/>. Accessed: 2025-04-20.
- Mihai Dascalu, Wim Westera, Stefan Ruseti, Stefan Trausan-Matu, and Hub Kurvers. 2017. Readerbench learns dutch: Building a comprehensive automated essay scoring system for dutch language. In *Artificial Intelligence in Education*, pages 52–63, Cham. Springer International Publishing.
- Paul Deane. 2013a. Covering the construct: An approach to automated essay scoring motivated by a socio-cognitive framework for defining literacy skills. In M. D. Shermis and J. Burstein, editors, *Handbook of Automated Essay Evaluation*, pages 298–312. Routledge.
- Paul Deane. 2013b. [On the relation between automated essay scoring and modern views of the writing construct](#). *Assessing Writing*, 18(1):7–24. Automated Assessment of Writing.
- Semire Dikli. 2006. An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment*, 5(1).

- 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.
- Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2024. Autoregressive score generation for multi-trait essay scoring. *arXiv preprint arXiv:2403.08332*.
- 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.
- J D’Arcy. 1997. Comparability studies between modular and non-modular syllabuses in gce advanced level biology, english literature and mathematics in the 1996 summer examinations. In *Standing Committee on Research on behalf of the Joint Forum for the GCSE and GCE*.
- Mary K Enright and Thomas Quinlan. 2010. Complementing human judgment of essays written by english language learners with e-rater® scoring. *Language Testing*, 27(3):317–334.
- Explosion. 2023. Available trained pipelines for dutch: nl_core_news_sm. https://spacy.io/models/nl#nl_core_news_sm [Accessed on March 11, 2025].
- Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. 2004. Coh-matrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202.
- 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.
- Yaqiong He, Feng Jiang, Xiaomin Chu, and Peifeng Li. 2022. Automated Chinese essay scoring from multiple traits. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3007–3016, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Sandra Heldsinger and Stephen Humphry. 2010. Using the method of pairwise comparison to obtain reliable teacher assessments. *The Australian Educational Researcher*, 37(2):1–19.
- Ian Jones, Marie Bisson, Camilla Gilmore, and Matthew Inglis. 2019. Measuring conceptual understanding in randomised controlled trials: Can comparative judgement help? *British Educational Research Journal*, 45(3):662–680.
- Ian Jones and Matthew Inglis. 2015. The problem of assessing problem solving: Can comparative judgement help? *Educational Studies in Mathematics*, 89(3):337–355.
- Anders Jonsson. 2014. Rubrics as a way of providing transparency in assessment. *Assessment & Evaluation in Higher Education*, 39(7):840–852.
- 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. <https://github.com/microsoft/LightGBM>. Accessed: 2025-04-22.
- Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.
- Kate Kelly, Mary Richardson, and Talia Isaacs. 2022. Critiquing the rationales for using comparative judgement: a call for clarity. *Assessment in Education: Principles, Policy & Practice*, 29:1–15.
- Vivekanandan Kumar and David Boulanger. 2020. Explainable automated essay scoring: Deep learning really has pedagogical value. *Frontiers in Education*, 5:572367.
- Donald Laming. 2003. *Human judgment: The eye of the beholder*. Cengage Learning, London, United Kingdom.
- Syed Latifi and Mark Gierl. 2021. Automated scoring of junior and senior high essays using coh-matrix features: Implications for large-scale language testing. *Language Testing*, 38(1):62–85.
- Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024. Unleashing large language models’ proficiency in zero-shot essay scoring. *arXiv preprint arXiv:2404.04941*.
- Marije Lesterhuis, Renske Bouwer, Tine van Daal, Vincent Donche, and Sven De Maeyer. 2022. Validity of comparative judgment scores: How assessors evaluate aspects of text quality when comparing argumentative texts. *Frontiers in Education*, 7:122–131.
- Xia Li and Jianda Liu. 2017. Automatic essay scoring based on coh-matrix feature selection for chinese english learners. In *Emerging Technologies for Education*, pages 382–393, Cham. Springer International Publishing.
- Andrey Lovakov and Elena R. Agadullina. 2021. Empirically derived guidelines for effect size interpretation in social psychology. *European Journal of Social Psychology*, 51(3):485–504.
- R. Duncan Luce. 1959. On the possible psychophysical laws. *Psychological Review*, 66(2):81–95.

- H.L.W. Maat, Rogier Kraf, Antal Van den Bosch, Nick Dekker, Maarten Van Gompel, Suzanne Kleijn, and Ko Sloot. 2014. T-scan: A new tool for analyzing dutch text. *Computational Linguistics in the Netherlands*, 4:53–74.
- Sandeep Mathias and Pushpak Bhattacharyya. 2020. Can neural networks automatically score essay traits? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91, Seattle, WA, USA (Online). Association for Computational Linguistics.
- Danielle S McNamara, Scott A Crossley, and Philip M McCarthy. 2010. Linguistic features of writing quality. *Written Communication*, 27(1):57–86.
- Danielle S. McNamara, Scott A. Crossley, Rod D. Roscoe, Laura K. Allen, and Jianmin Dai. 2015. A hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23:35–59.
- Anneleen V. Mortier, Marije Lesterhuis, Peter Vlerick, and Sven De Maeyer. 2015. Comparative judgment within online assessment: Exploring students feedback reactions. In *Computer Assisted Assessment. Research into E-Assessment*, pages 69–79, Cham. Springer International Publishing.
- C Paul Newhouse. 2014. Using digital representations of practical production work for summative assessment. *Assessment in Education: Principles, Policy & Practice*, 21(2):205–220.
- 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.
- Tiffany Potter, Letitia Englund, James Charbonneau, Mark MacLean, Jonathan Newell, and Ido Roll. 2017. Compare: A new online tool using adaptive comparative judgement to support learning with peer feedback. *Teaching & Learning Inquiry*, 5:89.
- Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.
- Chaitanya Ramineni and David Williamson. 2018. Understanding mean score differences between the e-rater® automated scoring engine and humans for demographically based groups in the gre® general test. *ETS Research Report Series*, 2018(1):1–31.
- Gert Rijlaarsdam, D. Weijen, and Huub Bergh. 1994. Relations between writing processes and text quality: When and how? *Cognition and Instruction*, 12:103–123.
- Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.
- Jeffrey T. Steedle and Steve Ferrara. 2016. Evaluating comparative judgment as an approach to essay scoring. *Applied Measurement in Education*, 29(3):211–223.
- Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.
- Tine van Daal, Marije Lesterhuis, Liesje Coertjens, Vincent Donche, and Sven De Maeyer. 2016. Validity of comparative judgement to assess academic writing: examining implications of its holistic character and building on a shared consensus. *Assessment in Education Principles Policy and Practice*, 26:59–74.
- San Verhavert, Renske Bouwer, Vincent Donche, and Sven De Maeyer. 2019. A meta-analysis on the reliability of comparative judgement. *Assessment in Education: Principles, Policy & Practice*, 26(5):541–562.
- San Verhavert, Sven De Maeyer, Vincent Donche, and Liesje Coertjens. 2018. Scale separation reliability: what does it mean in the context of comparative judgment? *Applied Psychological Measurement*, 42(6):428–445.
- 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.

A Rubric Description

Main Component	Rubric	Description
Structure	Title	The text has a title that clearly matches the content of the text.
Structure	Construction	The text contains a clear division into: introduction, argumentation, and conclusion.
Structure	Layout	The text is well-organized. There is a clear division into paragraphs. Paragraphs are separated by: blank lines, indentation, or starting on a new line.
Structure	Subtopic	Each paragraph has its own single (sub)topic.
Structure	Relationships between Paragraphs	There is a clear 'train of thought' between paragraphs: based on the text, coherence relationships between paragraphs can be clearly (easily) identified.
Structure	Continuity	Information that belongs together is also grouped together in the text.
Content	Introduction	In the introduction, the proposition/statement is presented, and optionally, the writer's opinion on the proposition is also made clear.
Content	Persuasion	It is clear what the writer wants to convince the reader of: a choice for or against the presented proposition.
Content	References	The text contains at least two (parts of) references, which are meaningfully incorporated into the text. For example, they support the argumentation or are used as an example in the introduction.
Content	Citations (quoting from references)	The quotes from the references are correctly marked in the text. Direct quotes (between quotation marks) and paraphrases both have a source citation.
Content	Reader Focus	The text is easily understandable for a reader unfamiliar with the assignment. For example, there is no reference to the writing task assignment or the writer's environment.
Content	Reader Engagement	The reader is clearly engaged with the text through examples referring to daily life or common experiences.
Content	Conclusion	The text contains a clear conclusion that aligns with the rest of the text and from which the writer's opinion is evident. It is clear that this concludes the text.
Argumentation	Support	The argumentation consists of multiple arguments that support the writer's opinion.
Argumentation	Relevance	The argumentation does not contain too much superfluous information, i.e., information that does not contribute to supporting the writer's opinion.
Argumentation	Indication of Argumentation	The arguments are clearly recognizable as arguments; e.g., through the use of constructions like "therefore I believe (do not believe) that...", "I find/think...", "I (do not) agree with this", etc.
Argumentation	Referential and Coherence Relations	The referential and coherence relations are clear when implicit, or explicitly marked. Examples of markers are: therefore, thereby, thus, because, since, first, second, third, then, etc.
Language	Grammar and Spelling	The text contains no grammatical and/or spelling errors.
Language	Punctuation	Punctuation marks are applied correctly.

Language	Style	The tone and word choice are appropriate for the purpose and audience of the text.
----------	-------	--

Table 4: List of rubrics used to assess Dutch essays on argumentative writing. Each rubric was assigned a score between 1 and 5. For the original Dutch version, see [Coertjens et al. \(2017\)](#).

B T-Scan Configuration

Parameter	Value
Overlap Size	50
Frequency Clipping	99.0
MTLD factor size	0.72
Use Alpino parser?	yes
Store Alpino output?	yes
Use Wopr?	yes
One sentence per line?	no
Prevalence data	Belgium
Word Frequency List	subtlex_words.freq
Lemma Frequency List	subtlex_lemma.freq
Top Frequency List	subtlex_words20000.freq
Compound split method	compound-splitter-nl

Table 5: Configuration of T-Scan ([Maat et al., 2014](#)) used to extract linguistic features from Dutch essays.

C Hyperparameters

Rubric	Model	Optimal Hyperparameters
Title	LightGBM	boosting_type = gbdt, num_leaves=31, max_depth=-1, learning_rate=0.1, n_estimators=100, min_child_weight=0.001, min_child_samples=20
Construction	XGBoost	colsample_bytree = 0.6, gamma = 0.1, learning_rate = 0.05, max_depth = 10, min_child_weight = 7, n_estimators = 800, reg_alpha = 0.5, reg_lambda = 0.5, subsample = 0.4
Layout	RandomForest	max_depth = None, min_samples_split = 2, n_estimators = 100
Subtopic	XGBoost	colsample_bytree = 1.0, gamma = 0, learning_rate = 0.01, max_depth = 20, min_child_weight = 7, n_estimators = 300, reg_alpha = 0.0, reg_lambda = 0.0, subsample = 0.6
Relationships	ElasticNet	alpha = 0.01, l1_ratio = 0.3
Continuity	RandomForest	max_depth = None, min_samples_split = 2, n_estimators = 100
Introduction	ElasticNet	alpha = 0.01, l1_ratio = 0.3
Persuasion	ElasticNet	alpha = 0.01, l1_ratio = 0.1
References	Lasso	alpha = 0.001
Citations	XGBoost	colsample_bytree = 0.6, gamma = 0.1, learning_rate = 0.05, max_depth = 10, min_child_weight = 7, n_estimators = 800, reg_alpha = 0.5, reg_lambda = 0.5, subsample = 0.4
Reader Focus	ElasticNet	alpha = 0.01, l1_ratio = 0.1
Reader Engagement	ElasticNet	alpha = 0.01, l1_ratio = 0.1
Conclusive	XGBoost	colsample_bytree = 1.0, gamma = 0, learning_rate = 0.01, max_depth = 20, min_child_weight = 7, n_estimators = 300, reg_alpha = 0.0, reg_lambda = 0.0, subsample = 0.6
Support	Lasso	alpha = 0.001
Relevance	ElasticNet	alpha = 0.01, l1_ratio = 0.1
Indication	ElasticNet	alpha = 0.01, l1_ratio = 0.1
Reference Cohesion Relationships	XGBoost	colsample_bytree = 1.0, gamma = 0, learning_rate = 0.01, max_depth = 20, min_child_weight = 7, n_estimators = 300, reg_alpha = 0.0, reg_lambda = 0.0, subsample = 0.6
Grammar and Spelling	Lasso	alpha = 0.001
Punctuation	ElasticNet	alpha = 0.01, l1_ratio = 0.1
Style	ElasticNet	alpha = 0.01, l1_ratio = 0.1

Table 6: The optimal hyperparameters that were selected for the best-performing model during LOOCV. For each run of LOOCV, the optimal hyperparameters were selected based on the lowest average MAE, using 20-fold cross-validation with 100 randomized iterations. For brevity, we only report the most frequently selected optimal hyperparameters for the best models.

Enhancing Arabic Automated Essay Scoring with Synthetic Data and Error Injection

Chatrine Qwaider,¹ Bashar Alhafni,^{1,2} Kirill Chirkunov,¹
Nizar Habash,^{1,2} Ted Briscoe¹

¹MBZUAI, ²New York University Abu Dhabi
{chatrine.qwaider, kirill.chirkunov, ted.briscoe}@mbzuai.ac.ae
{alhafni, nizar.habash}@nyu.edu

Abstract

Automated Essay Scoring (AES) plays a crucial role in assessing language learners' writing quality, reducing grading workload, and providing real-time feedback. The lack of annotated essay datasets inhibits the development of Arabic AES systems. This paper leverages Large Language Models (LLMs) and Transformer models to generate synthetic Arabic essays for AES. We prompt an LLM to generate essays across the Common European Framework of Reference (CEFR) proficiency levels and introduce and compare two approaches to error injection. We create a dataset of 3,040 annotated essays with errors injected using our two methods. Additionally, we develop a BERT-based Arabic AES system calibrated to CEFR levels. Our experimental results demonstrate the effectiveness of our synthetic dataset in improving Arabic AES performance. We make our code and data publicly available.¹

1 Introduction

Automated Essay Scoring (AES) is a technology that automates the evaluation and scoring of essays to assess language learners' writing quality while eliminating the need for human intervention (Shermis and Burstein, 2003). AES has gained great interest due to its significant benefits in the field of education (Lagakis and Demetriadis, 2021; Susanti et al., 2023). AES systems help teachers evaluate many essays with consistent scoring and reduced workload. On the other hand, AES helps students improve their writing quality through rapid real-time scoring and feedback (Hahn et al., 2021).

Unlike for English, it is difficult to develop robust and scalable AES systems for Modern Standard Arabic (MSA), primarily due to the lack of essay datasets necessary for building effective Arabic AES (Lim et al., 2021; Elhaddadi et al., 2024). This

paper presents a framework to tackle the issue of data scarcity and quality by utilizing Transformers and Large Language Models (LLMs) to generate and build a synthetic dataset.

Our approach begins with prompting GPT-4o to generate a variety of Arabic essays covering multiple topics and different writing proficiency levels as defined by the Common European Framework of Reference (CEFR) (Council of Europe, 2001). Subsequently, we use a controlled error injection model to introduce errors into the correct Arabic essays, ensuring that erroneous essays reflect the type of errors that are commonly made by learners of Arabic in real-world scenarios. Our error injection approach consists of two steps: (i) *Error Type Prediction*, where a fine-tuned CAMELBERT MSA model (Inoue et al., 2021) classifies the most likely error type for each word, and (ii) *Error Realization*, where we apply a bigram MLE model to determine the most probable transformation for each predicted error type. Our framework enables the generation of realistic human-like essays, enhancing data augmentation for Arabic AES systems.

Our main contributions are as follows:

- Proposing a framework based on LLMs and Transformers for augmenting Arabic essays that accurately reflect human writing patterns.
- Creating a synthetic Arabic AES dataset with 3,040 essays annotated with CEFR proficiency levels.
- Developing an Arabic AES system using a BERT-based model, enabling accurate and scalable evaluation of Arabic essays based on CEFR standards.

The rest of the paper is organised as follows: §2 reviews related work on AES, §3 describes the dataset, and §4 outlines our data augmentation approach. §5 details the error injection methods, followed by an evaluation in §6. We discuss our results in §7 and §8 presents the conclusion and future work.

¹<https://github.com/mbzuai-nlp/arabic-aes-bea25>

2 Related Work

AES has been investigated extensively, particularly in English (Lim et al., 2021; Ramesh and Sanampudi, 2022), where multiple tools have been introduced such as IntelliMetric (Elliott et al., 2003), e-rater (Attali and Burstein, 2006), Grammarly,² Write and Improve³ (Yannakoudakis et al., 2018), and others. The development of English AES systems has been enabled by large scale annotated datasets such as the First Cambridge English (FCE) dataset⁴ (Yannakoudakis et al., 2011), Automated Student Assessment Prize (ASAP) dataset,⁵ the TOEFL11 corpus (Blanchard, 2013), and the ICLE (International Corpus of Learner English) (Granger, 2003). These datasets contain thousands of student essays with proficiency level grades, often along multiple dimensions.

In contrast, Arabic AES research has received less attention. Some studies have applied feature engineering and machine learning to develop models (Alghamdi et al., 2014; Al-Shalabi, 2016; Alobed et al., 2021; Gaheen et al., 2021), but they partially address key challenges, especially the scarcity of large, publicly available annotated datasets for improving Arabic writing quality.

Ghazawi and Simpson (2024) introduced AR-AES, a benchmark of 2,046 undergraduate essays from three university faculties, annotated by two educators per faculty using rubrics to assess academic performance. In contrast, our work focuses on writing proficiency, using the CEFR standard.

Bashendy et al. (2024) presented QAES, the first publicly available trait specific annotations for Arabic AES. QAES extends the Qatari Corpus of Argumentative Writing (QCAW) (Ahmed et al., 2024), which consists of 195 Arabic argumentative essays. They implemented multi-layered annotation of traits such as coherence, organization, grammar, and others. Despite its comprehensive annotation, it is small in size and limited to two prompts. While QAES multi-traits scores are publicly available, the QCAW holistic score is not.

Habash and Palfreyman (2022) presented the Zayed University Arabic-English Bilingual Undergraduate Corpus (ZAEBUC). This corpus comprises non-parallel essays in Arabic and English related to three prompts collected from first-year uni-

versity students with differing writing proficiency. ZAEBUC includes 216 annotated Arabic essays featuring manual annotations for syntactic and morphological characteristics and a CEFR-based proficiency assessment. Again, ZAEBUC is small in size and limited to three prompts.

Researchers have explored data augmentation methods like sampling, noise injection, and paraphrasing to address data scarcity and quality (Li et al., 2022). The recent development of LLMs has paved the way for researchers to explore promising new data synthesis solutions (Wang et al., 2024a; Long et al., 2024). Transformers and LLMs can closely mirror real-world distributions while introducing valuable variations across multiple tasks and domains (Wang et al., 2024b).

GPT models have shown strong capabilities in generating synthetic essays for English AES (Ramesh and Sanampudi, 2022). LLMs and Transformers have also generalized well in Arabic NLP tasks, including Question Answering (Samuel et al., 2024), Code Switching (Alharbi et al., 2024), NER (Sabty et al., 2021), Grammatical Error Correction (Alhafni and Habash, 2025; Solyman et al., 2023), and Sentiment Analysis (Refai et al., 2023). However, to the best of our knowledge no research has utilized such models to generate Arabic essays across CEFR writing proficiency levels.

3 Data: The ZAEBUC Corpus

For all our experiments, we use the ZAEBUC corpus (Habash and Palfreyman, 2022). ZAEBUC comprises essays written by native Arabic speakers, which were manually corrected and annotated for writing proficiency using the CEFR (Council of Europe, 2001) rubrics and scale. Each essay was annotated by three CEFR-proficient bilingual speakers. Habash and Palfreyman (2022), assigned a holistic CEFR level to each essay by converting the three CEFR ratings into numerical scores (ranging from 1 to 6) and then taking the rounded average. The essays in the corpus were limited to three prompt choices on *Social Media*, *Tolerance*, and *Development*; see Table 1. We use the splits created by Alhafni et al. (2023). Table 2 shows the CEFR level distribution of the ZAEBUC corpus based on holistic CEFR scores. The ZAEBUC corpus is limited in size and skewed toward B1–B2 levels, with no A1 or C2 essays. This common imbalance in Arabic learner data motivated our synthetic approach to create a more balanced CEFR distribution.

²<https://app.grammarly.com/>

³<https://writeandimprove.com/>

⁴<https://ilexir.co.uk/datasets/index.html>

⁵<https://www.kaggle.com/c/asap-aes>

وسائل التواصل الاجتماعي وتأثيرها على الفرد والمجتمع. How do social media affect individuals and society?
كيف نعزز ثقافة التسامح في المجتمع؟ How can the UAE promote a culture of tolerance in society?
التطور الحضاري الذي تشهده دولة الإمارات العربية المتحدة What do you think are the most important developments in the UAE at the moment?

Table 1: The prompts given to the essay writers in the ZAEBUC corpus (Habash and Palfreyman, 2022).

CEFR Level	Count	Percentage
A1	0	0%
A2	7	3%
B1	110	51%
B2	80	37%
C1	11	5%
C2	0	0%
Unassessable	6	3%
Total	214	100%

Table 2: ZAEBUC corpus CEFR level distributions.

4 Synthetic Data Augmentation

We propose a synthetic data augmentation approach leveraging the ZAEBUC dataset to generate synthetic essays that align with CEFR rubrics and have features similar to human text. The pipeline utilizes three phases: Building Essay Prompts, Feature Profiling, and finally Data Augmentation.

4.1 Building Essay Prompts

We began by compiling a diverse set of essay prompts across various categories and CEFR levels. While not directly drawn from established frameworks, our prompts were inspired by themes common in language assessments, including placement tests and academic writing. We aimed to cover familiar and level-appropriate topics, such as social issues, education, and personal experiences, while ensuring balance across the CEFR bands. We considered three proficiency levels: Beginner (A1–A2), Intermediate (B1–B2), and Advanced (C1–C2). General themes, such as hobbies, suited all levels, while more complex topics, including politics, Technology, and Education, were reserved for advanced learners.

Topic	B	I	A
Culture and Traditions	1	3	2
Daily Life	2	2	2
Education	3	6	8
Environment	2	2	3
Future	1	2	2
History and Culture	2	2	2
Hobbies	3	2	2
Imaginary	5	2	2
Life/Time Management	4	4	2
Personal Experiences	7	2	2
Relations	4	2	2
School Life	4	2	2
Sport and Health	2	3	1
Technology and Media	2	8	6
Travel and Experience	1	2	1
Politics and Government	2	2	7
Social Issues	2	7	6
Total	47	53	52

Table 3: Count of Arabic text prompts by level and topic. B: Beginner level (A1, A2), I: Intermediate level (B1, B2), A: Advanced level (C1, C2).

Using LLMs like GPT-4o⁶, Gemini⁷, and Copilot⁸, we generated 100 prompts, followed by a manual review to remove redundancies and ensure both relevance for Arabic essay writing and balanced proficiency coverage. The final collection consists of 152 balanced and diverse prompts. Table 3 presents the selected categories and the distribution of the prompts across levels, while Table 4 provides example prompts for the Hobbies category.

4.2 Feature Profiling

We construct linguistic profiles for each CEFR level using the ZAEBUC corpus. Each profile contains various levels of linguistic information. Representing different lexical and syntactic features, we use the number of words/sentences (N_w, N_s), the number of tokens/vocabulary (N_v), words/sentences lengths (L_w, L_s), and sentence complexity measured by syntactic tree depth (D_s).

We define the lexical diversity (Type-Token Ratio, TTR) as:

$$\text{TTR} = \frac{\text{Unique Tokens}}{\text{Total Tokens}} \quad (1)$$

⁶<https://openai.com/index/hello-gpt-4o/>

⁷<https://gemini.google.com/app>

⁸<https://copilot.microsoft.com/>

Level	Arabic Prompt	English Prompt
Beginner	<ul style="list-style-type: none"> • ما هي هوايتك المفضلة؟ • تحدث عن نشاط تحبه في عطلة نهاية الأسبوع. • ما هي الرياضة التي تحب ممارستها؟ لماذا؟ 	<ul style="list-style-type: none"> • What is your favorite hobby? • Talk about a weekend activity you love. • What is your favorite sport? Why?
Intermediate	<ul style="list-style-type: none"> • ما هي هوايتك المفضلة وكيف بدأت في ممارستها؟ لماذا تحبها؟ • هل ترغب في تعلم هواية جديدة؟ ما هي ولماذا تهمك؟ 	<ul style="list-style-type: none"> • What is your favorite hobby and how did you start practicing it? Why do you enjoy it? • Do you wish to learn a new hobby? What is it and why does it interest you?
Advanced	<ul style="list-style-type: none"> • ناقش كيف تؤثر الهواية على صحتك النفسية والجسدية. • كيف يمكن أن تكون الهوايات وسيلة للتعبير عن الذات؟ • هل هناك هواية جديدة ترغب في تجربتها؟ • ناقش الأسباب التي تجعلك مهتمًا بها وكيف تعتقد أنها ستفيدك. 	<ul style="list-style-type: none"> • Discuss how hobbies impact your mental and physical health. • How can hobbies serve as a means of self-expression? • Is there a new hobby you wish to try? Discuss the reasons you are interested in it and how you believe it will benefit you.

Table 4: Examples of prompts related to the topic of Hobbies and classified into one of three different levels.

Similarly, we calculate the sentence complexity by:

$$C_s = \frac{\sum_{i=1}^N D_i}{N} \quad (2)$$

where D_i is the syntactic depth of sentence i and N is the total number of sentences.

For morphological features, we use the ZAE-BUC morphological annotations: the most frequent POS tags, such as nouns, verbs, adjectives, etc.

We aggregate all extracted features across the essays to get a quantitative representation at different writing CEFR levels, which serves as a reference for later stages.

4.3 Zero-shot Data Augmentation

Effective LLM prompt engineering has become increasingly important, as the model’s output varies based on the prompt, provided instructions, and prompt language. Previous studies in Arabic NLP have shown that using English as the instruction language for input prompts can improve output quality (Kmainasi et al., 2024; Koto et al., 2024).

In our approach, we experiment with various prompts for zero-shot data augmentation to identify those that produce human-like text while adhering to guideline instructions. We use GPT-4o as our generation model due to its affordability and larger token capacity for both input and output. The GPT prompts include (a) the target CEFR level, (b) CEFR guidelines and instructions, (c) the linguistic profile for the targeted CEFR level to control the prompt output, and (d) the topic prompt or question from the previously mentioned topic prompts dataset. For these missing levels (A1 and C2), instead of injecting a pre-defined profile, GPT-4o was directly prompted to act as an assistant and gener-

ate data based on the general standards and rubrics of the CEFR.

To check the quality of the generated essays and whether they follow the prompt instructions, we build a linguistic feature profile (vector) for each augmented essay. We then assess the alignment between the generated essays and the reference CEFR-level profiles by computing their feature vectors’ cosine similarity as in equation 3. Specifically, given two real-valued feature vectors P_i (the CEFR reference profile) and Q_i (the generated essay), the cosine similarity is calculated as:

$$\cos(\theta) = \frac{\sum_i P_i Q_i}{\sqrt{\sum_i P_i^2} \cdot \sqrt{\sum_i Q_i^2}} \quad (3)$$

This metric ensures that the synthetic data closely aligns with real human essay patterns. Based on the computed similarity score, we assign a predicted CEFR level to each essay.

Later, we calculate the alignment between the predicted CEFR level and the target level specified in the GPT-4o prompt (ground truth) using the following agreement formula:

$$\text{Agreement} = \frac{\sum_{i=1}^n (\hat{y}_i = y_i)}{n} \quad (4)$$

where \hat{y}_i is the predicted level and y_i is the ground truth. This process evaluates how well GPT-4o succeeded in aligning the generated content with the intended proficiency level, serving as a measure of agreement rather than a prediction from an external model.

We conducted multiple rounds of prompt engineering refinements to improve the quality of the generated Arabic essays and ensure alignment with CEFR levels.

CEFR Level	Count	Percentage
A1	470	15.5%
A2	470	15.5%
B1	530	17.4%
B2	530	17.4%
C1	520	17.1%
C2	520	17.1%
Total	3,040	100%

Table 5: The generated corpus CEFR level distributions.

First, we found that straightforward prompts without explicit controlled linguistic instructions and explanations resulted in incoherent essays, including irrelevant topics and English text, achieving only 20.5% matching agreement with linguistic feature profiles. In a subsequent round, we introduced detailed definitions of linguistic features and restricted outputs to Arabic-only text, which improved agreement to 26%. However, the model still occasionally produced incomplete essays and injected text from the prompt into the essay.

The most effective prompt structure format is illustrated in Figure 1. We separated system-level control instructions from user-defined parameters, thereby providing clearer guidance for structured and proficiency-aligned text generation. This refinement increased agreement to 27.5%, demonstrating that precisely controlled instructions enhance LLM performance in structured writing tasks.

Ultimately, we generated 3,040 Arabic essays covering all CEFR levels and various topics, where each prompt was used to create ten essays. This effort was intentionally designed to address the imbalanced CEFR distribution in the original ZAEBUC corpus, where B-level essays were overrepresented. By constructing a more balanced synthetic dataset, we aimed to enhance model performance across the full proficiency spectrum. The structured and controlled prompt design also improved alignment with learner writing styles while providing a consistent framework for generating realistic Arabic essays. Table 5 presents the distribution of generated essays across different CEFR levels. The full dataset statistics are provided in Appendix A.1.

5 Error Injection

Human-generated text naturally contains some grammatical errors and linguistic infelicities. In

```
{
  "role": "system",
  "content": "You are a helpful assistant that generates essays in Arabic. Try to make them different, focus on other aspects and ideas, DO NOT generate anything from the prompt about the level or the features. Generate the Essays only in Arabic and make sure you generate completed sentences."
},
{
  "role": "user",
  "content": "f'Generate an Arabic essay talking about prompt: {prompt_text} for CEFR level {cefr_level} based on the CEFR Guidelines: {guidelines}. You have to follow the linguistics features profile as {linguistics_profile}. Here is the feature explanation: {features_guidelines}"
}
```

Figure 1: GPT-4o prompts messages that have been used to generate Arabic essays

order to create human-like essays, we need to add similar kinds of errors to the synthetic essays that reflect the level of writing attainment. In this phase, we prompt GPT-4o to inject errors into the previously generated essays while maintaining their aligned CEFR levels by utilizing error profiling.

5.1 Error Profiling

Error Distribution Profiles To model the distribution of errors to inject into the synthetic essays, we again leverage the ZAEBUC corpus, which contains the erroneous essays aligned with the manually corrected ones. We followed the same methodology we used to construct the linguistic feature profiles for each CEFR level to develop error distribution profiles aligned with CEFR levels. The error profile captures and reflects the authentic distribution patterns observed in human writing at different CEFR levels.

Developing an Error Instruction Repository To prompt GPT-4o to generate essays containing errors we applied the Grammatical Error Detection (GED) model proposed by (Alhafni et al., 2023) to the ZAEBUC corpus to annotate errors using 13 error tags and to obtain error distributions for each CEFR level. We created the repository using the error tags, where we also added a formal definition of what those tags describe in terms of linguistic errors. In addition, we expanded the error taxonomy by splitting it into finer-grained classes. Each error instruction is followed by an example showing the correct word and the erroneous version. The explanation was based on the *extended ALC taxonomy* (Alfaifi et al., 2013), which was refined later and introduced as ARETA (Belkebir and Habash, 2021). Appendix C presents examples of the error types. Figure 2 shows some examples from our error instruction repository.

5.2 GPT-Based Error Injection

We prompted GPT-4o to inject errors into the synthetically generated essays based on the error distri-


```

"REPLACE_0":
"Orthographical Error for example : Use incorrect Ya and Alif-Maqsurah forms at the end of words, replacing 'ي' with 'ي' or 'ي' with 'ي', such as using 'علي' instead of 'علي'."
"Orthographical Error for example : Swap the order of two adjacent characters in words to create orthographic errors, e.g., 'تقريبنا' instead of 'تقريبنا'."
"Orthographical Error for example : Introduce a common error by lengthening a vowel sound in words, for instance, changing 'نغم' to 'نغمي'."
"Orthographical Error for example : Confuse 'س' with 'س' at the end of words, creating common spelling errors, like 'مشاركة' becoming 'مشاركه'."
"Orthographical Error for example : Omit or misuse Alif Fariqa, such as writing 'ويكتف' instead of 'ويكتف'."
. . . . .

```

Figure 2: An example of orthographical error instructions from the developed errors instructions repository

tribution profiles while maintaining the CEFR level. The model processed one essay at a time in a zero-shot setting, except that we included the definition and explanation of the error tags. For example, **M** indicates a morphological error, while **Merge** targets two mistakenly split tokens that need to be merged, and so on.

After conducting multiple experiments, we observed the following issues: (i) The model struggled to follow the predefined error distribution perhaps due to the complexity of the prompts. (ii) The model was confused by certain error tags, particularly **Split** and **Merge**. These errors were mainly ignored in the injected text. (iii) We calculated the cosine similarity between the main error profile and the injected essays’ error distributions as shown in Equation 3. When we injected all errors at once, the similarity agreements did not exceed 20%; however, when we reduced the number of error tags per essay the agreements significantly improved, reaching 86%.

Therefore we implemented a method where each error type was injected separately. This required multiple iterations over the same essay, corresponding to the number of error tags shown in the error distribution profile for each CEFR level. Figure 3 shows an example of a GPT-4o prompt for error injection. Some error types, especially orthographic errors, are more frequent among Arabic writers than others. The prompt was intentionally designed through prompt engineering. The ‘helpful assistant’ component establishes a cooperative persona for the LLM, while the subsequent instruction to ‘inject erroneous tokens’ explicitly guide GPT-4o towards the specific task of error introduction. This approach ensures that GPT-4o is not making random edits but is rather following predefined instructions to create targeted errors, aligning with the overall goal of generating realistic synthetic data.

```

{"role": "system", "content": "You are a helpful assistant that injected erroneous tokens in Arabic essay based on given error instruction"},
{"role": "user", "content": f"Rewrite the following Arabic essay with the {cefr} CEFR level, following the specified error Instruction without removing, changing or fixing any existing mistakes.\n\n" f"Essay in Arabic:\n{essay}\n\n" f"Error Instruction: Please Inject exactly {num_words_to_inject} error/s of this type: {random_error_prompt}\n\n" "Only apply the specified errors directly in the text without any introductory or additional comments."}

```

Figure 3: Sample GPT-4o error injection prompt

```

For each error tag:
  • Randomly select error prompts from Error instruction repository based on the average error count:
    • Average count <5: Use 1 Error instruction.
    • Average count between (5,10): Use 2 Errors instructions
    • Average count >10: Use 3 Errors instructions.
  • Inject errors into the essay according to the determined distribution.

```

Figure 4: Error injection based on average error count

To reflect this, we randomly select weighted error instructions based on the average frequency of each error type. Figure 4 shows the pseudocode for the selection process. The full pseudocode is in Appendix B.

5.3 Controlled Error Injection

We introduce a controlled method for injecting errors into clean text, ensuring that the resulting erroneous sentences follow the empirical error distributions observed at each CEFR level. More formally, given an input sentence (X) and its CEFR level (L), we introduce errors in two steps: Error Type Prediction and Error Realization.

Error Type Prediction We estimate the probability of an error type occurring at a given word, i.e., $P(\text{error_type}|\text{word})$. To do so, we leverage ARETA in a reverse annotation process where we process correct–erroneous sentence pairs, tagging each correct word with its corresponding error type. Using this annotated data, we train a token-level BERT classifier to predict the most likely error type for each word in a given correct sentence. We fine-tune CAMELBER T MSA (Inoue et al., 2021) to build our classifier.

Error Realization To determine how a word should be corrupted, we first align correct–erroneous sentence pairs using the algorithm proposed by Alhafni et al. (2023). For each aligned pair, we extract edit transformations that capture the operations required to convert a correct word into its erroneous

counterpart. Using this data, we estimate $P(\text{transformation}|\text{error_type})$ with a bigram Maximum Likelihood Estimation (MLE) lookup model: $\text{count}(\text{transformation}, \text{error_type}) / \text{count}(\text{error_type})$. During inference, we apply the BERT classifier to predict error types for each word in a sentence. We then filter these predictions, retaining only error types relevant to the sentence’s CEFR level. Finally, the MLE model selects the most probable corruption for a given error type. A complete example of a B1-level essay generated by the proposed model is in Figure 5.

6 Experimental Setup

This study focuses on introducing a data augmentation framework and synthetic Arabic essay corpus, rather than proposing a new AES model. We use a BERT-based model trained on the original ZAEBUC dataset as the reference baseline, evaluating how different augmentation strategies (e.g., GPT-4o generation, BERT-based error injection) improve performance relative to this setup.

6.1 Data and Metrics

We use the ZAEBUC dataset for all the experiments, following the splits created by [Alhafni et al. \(2023\)](#): 70% Train, 15% Dev, and 15% Test.

Our primary evaluation metric is Quadratic Weighted Kappa (QWK) ([Cohen, 1968](#)), the most widely used metric in AES research ([Ke and Ng, 2019](#)). We also report accuracy, macro precision (P), recall (R), and F_1 scores. Model predictions are evaluated in two settings: average-reference and multi-reference. The average-reference setting uses the rounded average of the three scores as the gold label, while the multi-reference considers each of the three human-assigned labels as a valid reference during evaluation, following a more tolerant evaluation strategy (§3).

6.2 Model

We treat the task of AES as a text classification problem. We fine-tune CAMeLBERT MSA ([Inoue et al., 2021](#)) on the training split of ZAEBUC. The models were trained by using the average CEFR gold labels. During training, we ignore the essays that are labeled as Unassessable, but we penalize the models for missing them in the evaluation. We fine-tune the models for 5 epochs, with a maximum sequence length of 512, a learning rate of $5e-5$, and a batch size of 32.

6.3 Results

Our results are presented in Table 6. Our baseline system, only trained on the ZAEBUC training set, indicates room for improvement, with the F_1 at 24.50% and QWK at 22.44%. We then switched between different datasets to measure the impact of data augmentation on the model.

Impact of Synthetic Data We tested data augmentation by adding 3,040 corrected GPT-4o-generated essays, which lowered QWK but increased F_1 . Notably, the multi-reference setting saw significant gains, with QWK at 96.00% and F_1 at 92.32%. This pattern stems from the flexibility of multi-reference evaluation, which treats all three human-assigned CEFR labels as valid references. This accommodates natural scoring variations and increases the chance that model predictions, especially on synthetic data, align with at least one reference label, boosting QWK and F_1 scores for both GPT-generated and error-injected essays.

Comparison of Error Injection Methods As the initial synthetic essays were error-free, we further refined the model by adding essays with human-like errors. We compared two methods from §5: (1) GPT-based error injection (with and without instruction examples) and (2) the controlled BERT-based method.

The results demonstrate that the controlled error model improves performance in all metrics, particularly in the average reference setting, which achieved 27.87% and 38.02% for QWK and F_1 , respectively. This result aligns with expectations, as the BERT-injected errors closely follow CEFR-based error distributions, producing errors that realistically reflect learner writing and better match the average of human ratings.

GPT-based error injection performed best in the multi-reference setting, with one-shot examples reaching 96.47% QWK and zero-shot boosting F_1 to 95.12%. While less aligned with CEFR profiles, GPT errors benefit from fluency and variability, increasing the chance of matching at least one human reference in this flexible evaluation.

7 Discussion

This study demonstrated the effectiveness of synthetic data and controlled error injection in enhancing Arabic AES, providing key insights into metric interpretation, data expansion, and methodological choices.

Level	B1	Topic	Hobbies
Prompt	هل ترغب في تعلم هواية جديدة؟ ما هي ولماذا تهتمك؟		
	Do you wish to learn a new hobby? What is it and why does it interest you?		
Correct Essay			
<p>تعلم هواية جديدة يمكن أن يكون تجربة مثيرة ومفيدة في حياتنا. في الوقت الحالي، أفكر في تعلم الرسم كمهارة جديدة. الرسم ليس فقط وسيلة للتعبير عن النفس، ولكنه أيضًا يساعد في تحسين التركيز والصبر. عندما أرسم، أجد نفسي أركز على التفاصيل الصغيرة، مما يساعدني على تحسين مهارات الملاحظة لدي.</p> <p>الرسم يعطيني الفرصة للتعبير عن مشاعري وأفكاري بطريقة بصرية. أحيانًا، تكون الكلمات غير كافية للتعبير عما نشعر به، وهنا يأتي دور الفن. بالإضافة إلى ذلك، تعلم الرسم يمكن أن يكون وسيلة رائعة للاسترخاء بعد يوم طويل من العمل أو الدراسة. الجلوس أمام لوحة بيضاء وتحويلها إلى قطعة فنية يمكن أن يكون تجربة مهدنة ومرحة.</p> <p>أيضًا، الرسم يمكن أن يفتح لي أبوابًا جديدة للتفاعل مع الآخرين. يمكنني الانضمام إلى ورش عمل أو مجموعات فنية، حيث يمكنني مقابلة أشخاص يشاركونني نفس الاهتمام. هذا يمكن أن يساعدني في بناء شبكة اجتماعية جديدة وتعلم تقنيات جديدة من الآخرين.</p> <p>بشكل عام، تعلم الرسم كهواية جديدة يهمني لأنه يجمع بين التعبير الفني، تحسين الذات، والتفاعل الاجتماعي. إنه تحد جديد أود أن أواجهه، وأنا متحمس لاكتشاف ما يمكنني تحقيقه من خلال هذه الهواية.</p>			
Erroneous Essay			
<p>ت علم هواية جديدة يمكن أن يكون تجربة مثيرة ومفيدة في حياتنا. في الوقت الحالي، أفكر في تعلم الرسم كمهارة جديدة. الرسم ليس بس وسيلة للتعبير عن النفس، ولكنه أيضًا يساعد في تحسين التركيز والصبر. عندما أرس، أجد نفسي أركز على التفاصيل الصغيره، مما يساعدني على تحسين مهارات الملاحظة لدي.</p> <p>الرسم يعطيني الفرصة للتعبير عن مشاعري وأفكاري بطريقة بصرية. أحيانا، تكون الكلمات غير كافية للتعبير عما نشعر به، وهنا يأتي دور الفن. بالإضافة إلى ذلك، تعلم الرسم يمكن أن تكون وسيلة رائعة للإسترخاء بعد يوم طويل من العمل أو الدراسة. جلوس أمام لوحة بيضاء وتحويلها الى قطعة فنية يمكن أن تكون تجربة مهدنة ومرحة.</p> <p>أيضا، الرسم يمكن أن يفتح لي ابوابا جديد لتفاعل مع الآخرين. يمكنني الانضمام الى ورش عمل أو مجموعات فنية، حيث يمكنني مقابلة أشخاص يشاركونني نفس الاهتمام. هذا يمكن أن يساعدني في بناء شبكة اجتماعية جديدة وتعلم تقنيات جديد من الآخرين.</p> <p>بشكل عام، تعلم الرسم كهواية جديدة يهمني لأنه يجمع بين التعبير الفني، تحسين الذات، والتفاعل الاجتماعي. إنه تحد جديد أود أن أواجهه، وأنا متحم لاكتشاف ما يمكنني تحقيقه من خلال هذه الهواية.</p>			

Figure 5: An Example of a B1 Arabic Essay generated by GPT-4o using the Hobbies prompt and the same essay after injecting errors by the controlled BERT-based model.

First, we emphasize that QWK offers a more robust metric than accuracy for evaluating AES systems, particularly under imbalanced class distributions. Unlike accuracy, which is biased by majority classes, QWK penalizes errors by their ordinal distance from the correct label. As Table 6 shows, even modest QWK improvements indicate meaningful advancements in differentiating CEFR levels, a distinction especially relevant given the skewed ZAEBUC dataset.

The significant gains observed in the multi-reference setting with generated GPT-4o essays stem from its flexibility. This evaluation approach treats all three human-assigned CEFR labels as valid references, accommodating natural scoring variations and increasing the chance that model

predictions align with at least one reference label.

Our analysis revealed that while GPT-4o is powerful for generating diverse content, it struggles to precisely follow the nuanced distribution and specific linguistic features, including error patterns, observed in the manually annotated ZAEBUC dataset. In the GPT-based error injection approach, error type selection is guided by average error counts from the ZAEBUC corpus, but error realization depends on GPT-4o's interpretation of the prompt, making it less predictable. This inherent challenge in mimicking human-like linguistic and error distributions through zero-shot generation directly contributed to the observed lower agreement rate.

In contrast, the controlled method employs a BERT-based classifier for error prediction and ap-

Train Data	Average Reference					Multi-Reference				
	QWK	Acc	F ₁	P	R	QWK	Acc	F ₁	P	R
ZAEBUC (baseline)	22.44	57.58	24.50	23.33	26.76	61.06	84.85	43.70	42.50	45.31
ZAEBUC + GPT essays	14.92	60.61	26.43	25.55	27.45	96.00	96.97	92.32	98.04	88.89
ZAEBUC + BERT errors	27.87	57.58	38.02	35.86	44.93	82.70	87.88	71.66	70.83	74.38
ZAEBUC + GPT errors_1	17.14	57.58	25.64	25.18	26.27	96.47	96.97	94.16	97.92	91.67
ZAEBUC + GPT errors_0	20.84	57.58	32.76	31.53	46.08	93.79	93.94	95.12	96.49	94.44

Table 6: Performance comparison of different training datasets. GPT essays are the original correct essays generated from GPT-4o, BERT errors are the erroneous essays using the controlled injection BERT model, GPT errors_1 are the erroneous essays using GPT-4o with one-shot error example, while GPT errors_0 with Zero-shot settings.

plies transformations using bigram-MLE. This systematic approach resulted in a more robust replication of empirically observed error patterns, leading to its superior performance in the average-reference setting. This is expected, as BERT-injected errors more closely resemble learner writing and align more closely with average human ratings.

Overall, our findings highlight a trade-off between error alignment and fluency in data augmentation. Controlled error injection excels in the average-reference setting due to its closer alignment with learner errors, while GPT-based augmentation benefits from multi-reference flexibility but less reliably replicates authentic errors. The controlled BERT-based method thus serves as a key component of our pipeline, effectively addressing the limitations of direct GPT error injection.

Qualitative Analysis The qualitative analysis of the generation process revealed various biases in the GPT-4o outputs, including cultural, gender, and ideological biases. For instance, the essays frequently referenced traditional Arabic themes, reinforced stereotypical gender roles, and reflected culturally narrow assumptions. A clear example of religious bias is that الجمعة ‘Friday’ was selected as *the favorite day* in all 20 generated essays. Additionally, there was a noticeable tendency to use masculine forms throughout the texts. Such biases may unintentionally disadvantage students whose writing reflects different experiences, perspectives, or identities. Examples of these biases, along with their frequencies, are provided in Appendix A.2. We also observed a lack of diversity among the ten essays generated per prompt, with GPT-4o often repeating similar lexical and structural patterns.

8 Conclusions and Future Work

This paper presents a hybrid framework for Arabic AES, using LLMs and transformers to tackle

data scarcity by generating synthetic essays that partly replicate Arabic learner writing. Building on the ZAEBUC corpus, we developed CEFR-aligned linguistic and error profiles and used GPT-4o to produce 3,040 essays across 152 prompts. However, GPT-4o’s performance relies heavily on prompt engineering, achieving only 27.5% alignment with our reference profiles.

To introduce errors, we compare our two methods: (1) GPT-4o prompted multi-step error injection, and (2) our controlled method fine-tuning the CAMELBERT MSA model to inject errors proportionally to their profiled occurrence.

Evaluated with a fine-tuned BERT classifier, our hybrid framework, combining GPT-generated data with controlled error injection, outperformed the baseline (QWK: 27.87%, F₁: 38.02%), offering more reliable and interpretable results. These findings demonstrate the effectiveness of controlled error injection in capturing learner error distributions across CEFR levels.

For future work, we will prioritize integrating a human evaluation into our framework. Human annotators will assess the fluency and naturalness of synthetic essays, as well as the realism of injected errors, ensuring that they reflect typical learner patterns at specific CEFR levels.

To improve generalizability, we also plan to expand the diversity of prompts beyond predefined topics and incorporate a wider set of writing traits, including coherence, logical flow, and topic relevance, beyond syntactic and lexical features.

We also intend to enhance CEFR-level modelling by incorporating more manually annotated essays. This will help capture nuanced linguistic variations across levels and increase the robustness of our dataset. Lastly, we aim to deploy the AES system as an interactive tool to provide users with instant feedback on errors and proficiency levels.

Limitations

Despite the effectiveness of our hybrid Arabic AES framework, we note several limitations related to the quality of generated Arabic essays, error injection accuracy, and the generalization of the AES model. The lack of A1 and C2 essays in ZAEBUC means that there is no gold reference data for these levels, which may impact both linguistic and error profiles, affecting the accuracy of GPT-generated essays. Furthermore, different biases are present in both the ZAEBUC dataset and GPT-4o outputs as discussed in (§7)

In addition, due to the lack of comprehensive gold data, GPT struggles to fully replicate real learner writing styles, achieving only 27.5% agreement with linguistic feature profiles.

Another limitation is the model's ability to generalize across various domains and question types. The AES system may struggle with broader writing tasks and alternative prompts since the dataset and augmentation methods focus on predefined prompts. Relying solely on CEFR as a holistic scoring method limits interpretability. Enhancing the dataset with multi-trait annotations, such as coherence, argumentation, and organization, could improve scoring accuracy and feedback quality. Moreover, better-controlled GPT prompting could refine the quality and diversity of generated essays, reducing biases and improving alignment with real learner writing patterns.

Due to resource constraints, human evaluation was not feasible in this study; however, we plan to engage CEFR-trained annotators in the future.

Ethical Considerations

While Arabic AES systems provide significant support in assessing Arabic learners' writing proficiency, it is essential to highlight the ethical implications of their use. Automatic assessment and scoring may lead to misjudgments that could distress learners and students, especially if their work is incorrectly evaluated. AES tools should serve as an educational assistive technology, complementing the teacher's judgment, not replacing it in educational settings.

References

Abdelhamid M Ahmed, Xiao Zhang, Lameya M Rezk, and Wajdi Zaghouni. 2024. Building an annotated I1 arabic/I2 english bilingual writer corpus: The qatari

corpus of argumentative writing (qcaw). *Corpus-Based Studies across Humanities*, 1(1):183–215.

Emad Fawzi Al-Shalabi. 2016. An automated system for essay scoring of online exams in arabic based on stemming techniques and levenshtein edit operations. *arXiv preprint arXiv:1611.02815*.

Abdullah Alfaifi, Eric Atwell, and Ghazi Abuhakema. 2013. Error annotation of the arabic learner corpus: A new error tagset. In *Language Processing and Knowledge in the Web: 25th International Conference, GSCL 2013, Darmstadt, Germany, September 25-27, 2013. Proceedings*, pages 14–22. Springer.

Mansour Alghamdi, Mohamed Alkanhal, Mohamed Al-Badrashiny, Abdulaziz Al-Qabbany, Ali Areshey, and Abdulaziz Alharbi. 2014. A hybrid automatic scoring system for arabic essays. *Ai Communications*, 27(2):103–111.

Bashar Alhafni and Nizar Habash. 2025. [Enhancing text editing for grammatical error correction: Arabic as a case study](#). *Preprint*, arXiv:2503.00985.

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.

Sadeen Alharbi, Reem BinMuqbil, Ahmed Ali, Raghad AlOraini, Saiful Bari, Areeb Alowisheq, and Yaser Alonaizan. 2024. Leveraging llm for augmenting textual data in code-switching asr: Arabic as an example. *Proc. SynData4GenAI*.

Mohammad Alobed, Abdallah MM Altrad, and Zainab Binti Abu Bakar. 2021. An adaptive automated arabic essay scoring model using the semantic of arabic wordnet. In *2021 2nd International Conference on Smart Computing and Electronic Enterprise (IC-SCEE)*, pages 45–54. IEEE.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Eman Zahran, Hamdo Elhuseyin, Tamer Elsayed, Walid Massoud, and Houda Bouamor. 2024. Qaes: First publicly-available trait-specific annotations for automated scoring of arabic essays. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 337–351.

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.

D Blanchard. 2013. Toefl11: A corpus of non-native english. *Educational Testing Service*.

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

- C. o. E. Council of Europe. 2001. Common european framework of reference for languages: learning, teaching, assessment.
- Chima Elhaddadi, Imad Zeroual, and Anoual El Kah. 2024. Automatic arabic essays scoring: A scoping review. In *International Conference on Arabic Language Processing*, pages 38–48. Springer.
- S Elliott, MD Shermis, and J Burstein. 2003. Overview of intellimetric. *Automated Essay Scoring: A Cross-Disciplinary Perspective*, pages 67–70.
- Marwa M Gaheen, Rania M ElEraky, and Ahmed A Ewees. 2021. Automated students arabic essay scoring using trained neural network by e-jaya optimization to support personalized system of instruction. *Education and Information Technologies*, 26:1165–1181.
- Rayed Ghazawi and Edwin Simpson. 2024. Automated essay scoring in arabic: a dataset and analysis of a bert-based system. *arXiv preprint arXiv:2407.11212*.
- Sylviane Granger. 2003. The international corpus of learner english: a new resource for foreign language learning and teaching and second language acquisition research. *Tesol Quarterly*, 37(3):538–546.
- 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.
- Marcelo Guerra Hahn, Silvia Margarita Baldiris Navarro, Luis De La Fuente Valentín, and Daniel Burgos. 2021. A systematic review of the effects of automatic scoring and automatic feedback in educational settings. *IEEE Access*, 9:108190–108198.
- 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, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.
- Zixuan Ke and Vincent Ng. 2019. **Automated essay scoring: A survey of the state of the art**. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.
- Mohamed Bayan Kmainasi, Rakif Khan, Ali Ezzat Shahroor, Boushra Bendou, Maram Hasanain, and Firoj Alam. 2024. Native vs non-native language prompting: A comparative analysis. In *International Conference on Web Information Systems Engineering*, pages 406–420. Springer.
- Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Sadallah, Aisha Alraeesi, Khalid Al-mubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. **ArabicMMLU: Assessing massive multitask language understanding in Arabic**. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5622–5640, Bangkok, Thailand. Association for Computational Linguistics.
- Paraskevas Lagakis and Stavros Demetriadis. 2021. Automated essay scoring: A review of the field. In *2021 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6. IEEE.
- Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *Ai Open*, 3:71–90.
- Chun Then Lim, Chih How Bong, Wee Sian Wong, and Nung Kion Lee. 2021. A comprehensive review of automated essay scoring (aes) research and development. *Pertanika Journal of Science & Technology*, 29(3):1875–1899.
- Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.
- Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.
- Dania Refai, Saleh Abu-Soud, and Mohammad J Abdel-Rahman. 2023. Data augmentation using transformers and similarity measures for improving arabic text classification. *IEEE Access*, 11:132516–132531.
- Caroline Sabty, Islam Omar, Fady Wasfalla, Mohamed Islam, and Slim Abdennadher. 2021. Data augmentation techniques on arabic data for named entity recognition. *Procedia Computer Science*, 189:292–299.
- Vinay Samuel, Houda Aynaou, Arijit Chowdhury, Karthik Venkat Ramanan, and Aman Chadha. 2024. **Can LLMs augment low-resource reading comprehension datasets? opportunities and challenges**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 307–317, Bangkok, Thailand. Association for Computational Linguistics.
- Mark D Shermis and Jill C Burstein. 2003. *Automated essay scoring: A cross-disciplinary perspective*. Routledge.
- Aiman Solyman, Marco Zappatore, Wang Zhenyu, Zeinab Mahmoud, Ali Alfatemi, Ashraf Osman Ibrahim, and Lubna Abdelkareim Gabralla. 2023. Optimizing the impact of data augmentation for low-resource grammatical error correction. *Journal of King Saud University-Computer and Information Sciences*, 35(6):101572.
- Meilia Nur Indah Susanti, Arief Ramadhan, and Harco Leslie Hendric Spit Warnars. 2023. Automatic essay exam scoring system: A systematic literature review. *Procedia Computer Science*, 216:531–538.
- Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, et al. 2024a. A survey on data synthesis and augmentation for large language models. *arXiv preprint arXiv:2410.12896*.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, et al. 2024b. Autosurvey: Large language models can automatically write surveys. *arXiv preprint arXiv:2406.10252*.

Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for esl learners. *Applied Measurement in Education*, 31(3):251–267.

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.

A GPT-4o Generated Essays

A.1 Statistics

CEFR	#Essays	#Words	#Sentences	#Tokens	Avg_W_L	Avg_S_L	Unique_Tokens	Unique_Words
A1	470	39367	6721	47612	4.66	6.08	4518	4512
A2	470	54980	5565	63769	4.82	10.46	7147	7143
B1	530	100185	7276	113672	4.97	14.62	10742	10734
B2	530	127029	7804	142995	5.03	17.32	12522	12509
C1	520	136849	7630	152962	5.16	19.05	12832	12823
C2	520	146253	7946	163461	5.16	19.57	13686	13676
Total	3040	604663	42942	684471	5.04	14.94	27286	27272

Table 7: Summary statistics of the generated Arabic synthetic essay corpus across CEFR levels. #Essays denotes the number of essays; #Words refers to the total word count; #Sentences indicates the total number of sentences; #Tokens represents the total number of tokens (vocabulary items); Avg_W_L corresponds to the average word length in characters; Avg_S_L refers to the average sentence length in words; lexical diversity is captured through the counts of unique tokens and unique words.

A.2 Bias in the Generated Essays

Arabic Prompt	English Prompt	Arabic Response	English Response	Occurrences /20	Bias
ما هو يومك المفضل؟	What is your favorite day?	الجمعة	Friday	20	Cultural/Religious Bias
ما هو طعامك المفضل؟	What is your favorite food?	البيتزا	Pizza	16	Globalization Bias
ما هي هوايتك المفضلة؟	What is your favorite hobby?	القراءة	Reading	10	Socioeconomic/Class Bias
رحلة ذهبت إليها	A trip you went on	ذهبت إلى البحر	I went to the beach	18	Geographical/Cultural Bias
ماذا تفعل في عطلة نهاية الأسبوع؟	What do you do on the weekend?	نذهب إلى الحديقة	We go to the park	20	Geographical/Cultural Bias
ماهي المادة الدراسية المفضلة؟	What is your favorite school subject?	الرياضيات	Mathematics	17	Educational System Bias
ما هي رياضتك المفضلة؟	What is your favorite sport?	كرة القدم	Football	20	Cultural Bias
شخص تعتبره مثلك الأعلى	A person you consider your role model	والدي	My father	17	Gender Bias
من هو أفضل صديق لك؟	Who is your best friend?	أحمد	Ahmed	20	Gender/Name Bias
المهنة المستقبلية	Your future profession	طبيب	Doctor	13	Stereotype Bias
معلم شهير في العالم الأدبي	A famous teacher in the world of literature	الأهرامات	The Pyramids	17	Cultural Bias
أفضل فصول السنة	Your favorite season of the year	الصيف	Summer	20	Climate Bias
لغة تود تعلمها	A language you would like to learn	الأسبانية	Spanish	16	Language Bias
بلد ترغب في السفر إليه	A country you would like to visit	مصر	Egypt	10	National Identity Bias
قوة خارقة تمنها	A superpower you wish to have	الطيران	Flying	19	Media Bias

Table 8: Examples of response biases in GPT-4o generated essays.

B GPT-4o Error Injection Algorithm

Algorithm: Inject_Errors_and_Verify

Inject Errors into Essays

Input: Augmented_Essays (3040 essays from GPT-4o), Error_instructions, CEFR_Error_Profiles

Output: Errerouns_Essays

FOR each Essay in Generated_Essays:

Determine target CEFR level's error distribution

Target_CEFR = Get_CEFR_Level(Essay)

Retrieve the Error Profile

Error_Profile = Get_Error_Profile(Target_CEFR)

Inject errors based on error distribution

FOR each Error_Type in Error_Profile:

Select error instruction prompts based on average error count

Avg_Error_Count = Get_Avg_Error_Count(Error_Type)

IF Avg_Error_Count < 5:

Num_Prompts = 1

ELSE IF $5 \leq \text{Avg_Error_Count} \leq 10$:

Num_Prompts = 2

ELSE:

Num_Prompts = 3

Selected_Prompts = Select_Random_Prompts(Error_Prompts.json, Error_Type,
Num_Prompts)

Inject errors according to determined distribution

Essay = Inject_Errors(Essay, Selected_Prompts)

Verify Error Injection

FOR Errerouns_Essays:

Apply Grammar Error Detection (GED) to identify errors

Detected_Errors = Apply_GED(Injected_Essay)

Recalculate the error distribution for injected essays

Injected_Error_Profile = Calculate_Error_Distribution(Detected_Errors)

Compare the injected error distribution with the target CEFR profile

Similarity_Score = Cosine_Similarity(Injected_Error_Profile, Target_CEFR_Error_Profile)

End Algorithm

C Error Types Taxonomy

	11-Classes	42-Classes	Error Description	Correct Word	Erroneous Word
Morphology (M)	M	MI	Inflection	عارف	معروف
		MT	Tense	ذهب	يذهب
Orthography (O)	O	OA	Alef-Maqsura	القاضي	القاضي
		OA+OH	Alef-Maqsura + Hamza	أضحي	اضحاً
		OA+OR	Alef-Maqsura + Wrong Character	كشيء	كشيء
		OC	Chatacter Order	المدرسة	المردسة
		OD	Extra Character	هذا	هاذا
		OD+OG	Extra Character + Lengthening Short Vowels	تطورو	تطور
		OD+OH	Extra Character + Hamza	لأنهم	الأنهم
		OD+OM	Extra Character + Missing Character	الاجتماعي	الاجتماعي
		OD+OR	Extra Character + Wrong Character	الصور	السور
		OH	Hamza	العب	إعب
		OH+OM	Hamza + Missing Character	الأشياء	الاشياء
		OH+OT	Hamza + Ta-Marbuta	إمارة	اماره
		OM	Missing Character	المدرسة	المدسة
		OM+OR	Missing Character + Wrong Character	المجتمع	الحطمع
		OR	Wrong Character	المدرسة	المدرصة
		OR+OT	Wrong Character + Ta-Marbuta	مكتظة	مكتضه
		OT	Ta-Marbuta	غرفة	غرفه
		OW	Alef-Fariqa	كتبوا	كتبو
Semantics (S)	S	SF	Conjunction	فسبجان	سبجان
		SW	Word Selection	على	من
Punctuation (P)	P	P	Punctuation	السوق،	السوق.
Syntax (X)	X	XC	Case	رائعا	رائع
		XC+XG	Case + Gender	مجتهدا	مجتهدة
		XC+XN	Case + Number	نواح	نواحي
		XF	Definiteness	المفيد	مفيد
		XG	Gender	كان	كانت
		XM	Missing Word	على	NULL
		XN	Number	كتابين	كتب
		XT	Unnecessary Word	NULL	على
Combination (Comb.)	M+O	MI+OH	Inflection + Hamza	أشخاص	اشخاصك
	O+X	OH+XC	Hamza + Case	أضرارا	اضرار
SPLIT	SPLIT	SPLIT	Split	دولة الإمارات	دولة الإمارات
MERGE	MERGE	MERGE	Merge	بالعلم	ب العلم
DELETE	DELETE	DELETE	Delete	NULL	داخل

Table 9: Illustrative examples of error types categorized according to the ARETA error taxonomy (Belkebir and Habash, 2021). The table presents hierarchical mappings from coarse-grained (11-Class) to fine-grained (42-Class) error categories, alongside representative corrections.

Direct Repair Optimization: Training Small Language Models For Educational Program Repair Improves Feedback

Charles Koutcheme, Nicola Dainese and Arto Hellas

Aalto University, Espoo, Finland

first.last@aalto.fi

Abstract

Locally deployed Small Language Models (SLMs) offer a promising solution for providing timely and effective programming feedback to students learning to code. However, SLMs often produce misleading or hallucinated feedback, limiting their reliability in educational settings. Current approaches for improving SLM feedback rely on existing human annotations or LLM-generated feedback. This paper addresses a fundamental challenge: Can we improve SLMs' feedback capabilities without relying on human or LLM-generated annotations? We demonstrate that training SLMs on the proxy task of program repair is sufficient to enhance their ability to generate high-quality feedback. To this end, we introduce Direct Repair Optimization (DRO), a self-supervised online reinforcement learning strategy that trains language models to reason about how to efficiently fix students' programs. Our experiments, using DRO to fine-tune LLaMA-3.1-3B and Qwen-2.5-3B on a large-scale dataset of Python submissions from real students, show substantial improvements on downstream feedback tasks. We release our code to support further research in educational feedback and highlight promising directions for future work.

Code:  github.com/KoutchemeCharles/r1pf

1 Introduction

Learning to program remains challenging for many students, despite advances in teaching methodologies (Luxton-Reilly et al., 2018; Vihavainen et al., 2014). A key part of addressing these challenges is providing timely and accurate feedback (Jeuring et al., 2022), which is crucial for learning (Hattie and Timperley, 2007). Large Language Models (LLMs) such as GPT-4 have shown exceptional success in that task (Lohr et al., 2025), leading to their growing adoption in classrooms (Ahmed et al., 2025; Wang et al., 2024; Vadaparty et al., 2024; Liu et al., 2024a; Liffiton et al., 2024).

However, reliance on vendor-hosted LLMs raises substantial privacy concerns and potential ethical issues related to institutional control (Das et al., 2025). The privacy issues, jointly with scalability issues and associated costs, are driving a growing shift towards leveraging smaller open-source models (SLMs), which can be deployed and run locally within educational institutions or on students' computers and browsers (Yu et al., 2025b; Liu et al., 2024b).

However, smaller language models tend to produce misleading or hallucinated feedback, potentially confusing learners and negatively impacting learning (Koutcheme et al., 2025). Current methods for enhancing SLMs typically rely on supervised learning (Kotalwar et al., 2024) or reinforcement learning from either human annotations (Woodrow et al., 2025) or synthetic data generated by larger models (Ashok Kumar and Lan, 2024). These strategies come with limitations: human annotations are difficult to scale and replicate, while synthetic data inherits biases and capabilities from general-purpose LLMs.

Addressing these limitations raises a broader challenge for the field of programming education: *can we improve small language models' abilities to generate meaningful programming feedback without relying on external LLMs or human annotations?* Exploring this question opens a promising research direction focused on assessing how effectively small models can perform when trained exclusively on educational data. Understanding this can yield insights into the inherent capabilities of these models, free from external biases (DeepSeek-AI, 2025). This question matters not just technically, but also pedagogically; training models exclusively grounded in student work and learning contexts could facilitate more adaptable classroom deployment, enhance institutional control over model behaviour, and mitigate privacy concerns (Das et al., 2025).

Recent reinforcement learning techniques such as Group Relative Preference Optimization (GRPO) (Shao et al., 2024) have shown great promise in improving language models’ reasoning capabilities, allowing relatively small models to reach the performance of significantly larger ones with minimal LLM supervision (DeepSeek-AI, 2025). In parallel, prior work reveals a strong correlation between language models’ abilities to generate programming feedback and their capacity for fixing students’ programs (Koutcheme et al., 2024a): models proficient in program repair are independently good at generating feedback.

Building on this insight, we hypothesise that improving a small language model’s repair capabilities through reasoning could be sufficient to enhance the model’s feedback-generation abilities. We argue that the reasoning needed for generating a repair involves a thinking process useful to provide students with feedback, much like how teaching assistants reason about students’ mistakes before giving advice (Koutcheme, 2022).

Our paper thus aims to answer the following research question:

(RQ) How effective is training small language models to reason about educational program repair for improving their ability to generate high-quality programming feedback, and how does this technique compare to training models with LLM supervision?

To address this question, we introduce Direct Repair Optimization (DRO). This reinforcement learning training pipeline, based on Group Relative Preference Optimization, leverages historical datasets of student submissions to fine-tune small language models for program repair, using unit test results and syntactic/semantic distance measures as rewards to guide learning.

We apply DRO to fine-tune LLaMA-3.1-3B (Dubey et al., 2024) and Qwen-2.5-3B (Hui et al., 2024) on a large-scale dataset of student code from introductory programming courses (de Freitas et al., 2023). We evaluate the resulting model on multiple feedback tasks — including bug explanations, patch descriptions, and next-step hints — and show consistent improvements over all feedback criteria. Our contributions are as follows:

- We introduce a new LLM-free training pipeline for feedback generation based on program repair and reinforcement learning.

- We show that reasoning about program repair transfers to various forms of feedback, including explanations, fixes, and hints.
- We further demonstrate that reasoning about program repair improves the performance of LLM-distilled models.
- We release our code and data processing pipeline to support future research in aligning language models for educational feedback¹.

2 Related work

2.1 Programming Feedback

Program repair. Program repair has long been a cornerstone of AI-driven programming education, serving as a foundation for generating actionable feedback, such as next-step hints, through Intelligent Tutoring Systems (Rivers and Koedinger, 2017). Before the rise of instruction-tuned and chat language models, much of the work in this domain focused on leveraging closed pre-trained models, such as OpenAI Codex, to generate repairs through zero- or few-shot prompting. These approaches often relied on historical student submissions, automated unit tests, and other contextual information to guide the repair generation process (Zhang et al., 2022; Joshi et al., 2023).

In parallel, open-source language models were also explored for program repair tasks. For example, prior efforts have fine-tuned such models using datasets derived from student submissions (Koutcheme et al., 2023b) and automated repair tools (Koutcheme, 2023). These works demonstrated the viability of repair-focused training but did not directly explore its implications for improving natural language feedback.

Using program repair for improving feedback.

Even with the advent of chat models, several works proposed leveraging the quality of repairs generated alongside feedback as a validation mechanism to ensure only relevant suggestions reach learners (Phung et al., 2024; Sahai et al., 2023). In parallel, other studies propose generating a high-quality candidate repair program as a reasoning step to generate higher-quality feedback. (Phung et al., 2023; Sahai et al., 2023). This strategy has been extended with success to distil LLM-generated repair-induced feedback to small language models via Supervised Fine-tuning (Kotalwar et al., 2024).

¹  github.com/KoutchemeCharles/rlpf

Reinforcement learning from human and AI feedback. Supervised Fine-Tuning (SFT) methods are limited in their ability to align language models with nuanced human objectives (Ouyang et al., 2022). Several works have explored reinforcement learning techniques, such as Direct Preference Optimization (DPO) (Rafailov et al., 2024), to train small language models to generate higher-quality feedback. For example, this has been done by leveraging teaching assistants’ (TAs) edits of their responses to student forum questions (Hicke et al., 2023), collecting live TA preferences over several model-generated feedback (Woodrow et al., 2025), or combining existing high-quality human annotations with AI-generated alternatives (Ashok Kumar and Lan, 2024).

However, these methods typically rely on human supervision or the existence of labeled human feedback. This reliance poses challenges in contexts where such annotations are scarce or unavailable. While full Reinforcement Learning with AI Feedback (RLAIF) (Lee et al., 2024) approaches have been theorized to work well in the programming domain (Scarlatos et al., 2024), in this paper, we explore whether we can bootstrap feedback capabilities in small language models without requiring any human or LLM involvement.

2.2 Improving SLM Reasoning Without LLMs

Recent work has proposed leveraging automatically evaluable tasks to define preference pairs for DPO optimization (Pang et al., 2024), replacing the need for human or LLM judgments. In domains like programming and math, where correctness can be verified programmatically, this strategy has shown promising results. Combined with large-scale sampling and chain-of-thought prompting (Wei et al., 2022), such methods have yielded substantial improvements with minimal supervision (Pang et al., 2024).

Most recently, a new line of alignment techniques (Shao et al., 2024; Liu et al., 2025; Yu et al., 2025a) takes the idea back to a fully online optimization paradigm, showing major improvements. Inspired by the success of small models such as DeepScaleR (Luo et al., 2025), we explore whether these reinforcement learning techniques can be adapted to improve small language models in programming education.

3 Methodology

In this work, we hypothesise that training small language models for program repair will improve their feedback ability. To validate this hypothesis, we propose Direct Repair Optimization.

3.1 Background

Before presenting our approach, we first describe the setup and assumptions underlying our work.

3.1.1 Environment

We assume a typical educational programming setting, where student submissions are regularly collected and evaluated using automated assessment tools, such as unit tests, to assign scores and provide feedback (Paiva et al., 2022). Leveraging this infrastructure, we make two key assumptions: first, we assume access to a training dataset $\mathcal{D} = \{(d^i, s^i, c^i)\}_{i=1}^N$, comprising N tuples, where each tuple consists of a problem description d^i , a corresponding student program s^i , and a correctness label c^i , with $c^i = 0$ indicating an incorrect program and $c^i = 1$ indicating a correct one; second, we assume the availability of a grading function $u(s)$ that assigns a normalized score $c^i \in [0, 1]$ to each program s^i , reflecting its functional correctness based on unit test results.

3.1.2 Definition: high-quality repair

To support the rest of this article, we formalize the concept of a high-quality repair. A repair is typically considered high-quality if it meets two key criteria: functional correctness and closeness to the student’s original incorrect program (Koutcheme et al., 2024c; Phung et al., 2023; Joshi et al., 2023; Zhang et al., 2022). Functional correctness ensures that the repair successfully resolves the intended issues, while closeness ensures the repair preserves the student’s original approach, making the solution more interpretable and educationally meaningful (Price et al., 2017). Given a candidate repair \mathcal{R} (generated by an LM) for an incorrect program s_i , we assess its quality using automated evaluation methods. Functional correctness is measured through unit test results provided by the grading function u . For closeness, we use ROUGE (Lin, 2004), as it has been shown to be an effective and efficient measure for selecting high-quality repairs (Koutcheme et al., 2023a).

3.2 Direct Repair Optimization

In this section, we introduce Direct Repair Optimization (DRO), our approach to improve language models’ ability to generate educationally meaningful program repairs. DRO is an online reinforcement learning training method based on variants of Group Relative Preference Optimization (Shao et al., 2024). Figure 1 shows an overview of the method. At each iteration, given an incorrect program s^i , we (1) generate several completions, (2) compute individual reward scores, based on such generations and (3) update the model parameters based both on such rewards and a divergence score. Below we detail each step.

3.2.1 Sampling answers

Given an incorrect program, we sample G generations from our language model $\mathcal{G}_1^i, \mathcal{G}_2^i, \dots, \mathcal{G}_g^i \sim \pi_\theta(s^i, d^i)$. Our prompt asks our model to fix the student’s program but to reflect thoroughly before providing an answer (see Figure 2, Appendix C). Each generation contains a thought \mathcal{T}_g^i and a final repair \mathcal{R}_g^i : $\mathcal{G}_g^i = (\mathcal{T}_g^i, \mathcal{R}_g^i)$. Following (Shao et al., 2024), our prompt imposes the language model to structure its response using a set of predefined tags: the thought pattern should be generated within `<think> . . . </think>` and the repair within `<answer> . . . </answer>` tags.

3.2.2 Computing rewards

For each generation \mathcal{G}_g^i , we compute a reward r_g^i , reflecting the two criteria that define a high-quality repair (see Section 3.1.2): functional correctness and closeness (or “proximity”) to the student’s original incorrect program. The total reward r_g^i is thus the sum of two separate components: $r_g^i = f_g^i + p_g^i$.

The functional reward is an outcome-based reward (Luo et al., 2025) computed by extracting \mathcal{R}_g^i and passing it through the available grading function (i.e., unit tests):

$$f_g^i = \begin{cases} +1.0 & \text{if } u(\mathcal{R}_g^i) = 1 \\ +0.5 & \text{if } u(\mathcal{R}_g^i) - u(s^i) > 0 \\ -1.0 & \text{if } \mathcal{R}_g^i \text{ not compiling} \\ 0 & \text{otherwise} \end{cases}$$

We reward fully correct repairs (+1.0), give partial credit (+0.5) if the repair improves upon the student’s original program (i.e., it passes more tests than s^i), and penalize repairs that fail to compile or

generations that do not follow the expected format (−1.0). Our reward encourages the model to make meaningful progress toward correctness, even when it cannot fully solve the task. We believe that partially correct repairs that are better than the student’s original work could also benefit feedback generation.

The closeness reward evaluates how well the generated repair aligns with the student’s original code:

$$p_g^i = \begin{cases} \text{ROUGE}(s^i, \mathcal{R}_g^i) & \text{if } u(\mathcal{R}_g^i) = 1 \\ 0 & \text{otherwise} \end{cases}$$

By integrating this reward, we encourage the model not only to solve the programming task but to do so by building on the student’s own approach, implicitly forcing reasoning about what the student is currently doing and trying to achieve. This makes the repair (and the resulting feedback) more pedagogically aligned. We provide this reward only when the repair is fully correct. Since repairing a student program inherently requires changes, correctness and closeness can become competing objectives. Rewarding both simultaneously for partial outputs would risk destabilising training.

3.2.3 Updating the model using the rewards

We update the model using the computed rewards with the dr.GRPO loss function (Liu et al., 2025), a recent reinforcement learning loss function designed for training stability and efficiency:

$$\mathcal{J}_{\text{dr.GRPO}} = -\frac{1}{LG} \sum_{g=1}^G \sum_{t=1}^{|o_g|} l_{m,t} \quad (1)$$

where

$$l_{g,t} = \frac{\pi_\theta(o_{g,t} \mid d^i, s^i, o_{g,<t})}{\pi_{\theta_{\text{old}}}(o_{g,t} \mid d^i, s^i, o_{g,<t})} \hat{A}_g$$

and

$$\hat{A}_g = (r_g^i - \bar{r})$$

Here, LG is the maximum allowed completion length, $\pi_{\theta_{\text{old}}}$ is the model before the current update, and \hat{A}_g is the advantage, computed as the reward deviation from the batch mean \bar{r} . In practice, we use a clipped surrogate version of this objective that accounts for multiple updates per generation. For clarity and space, we provide the full objective and implementation details in section A.1 (Appendix A, where we also show how this formulation differs from the original GRPO loss introduced in the DeepSeek paper (DeepSeek-AI, 2025) and how it is better adapted to our task.

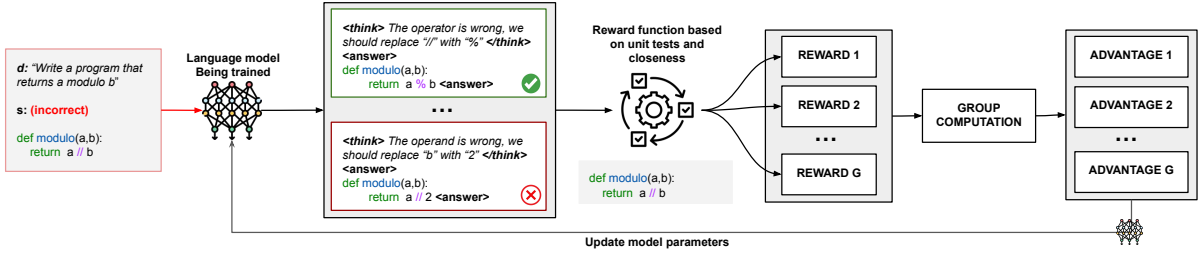


Figure 1: Overview of Direct Repair Optimization.

4 Experiments

In this section, we present the experiments conducted to answer our research question.

4.1 Falconcode: A Real-life Dataset

To answer our research question, we train language models using FalconCode (de Freitas et al., 2023), a publicly available dataset containing real-life CS1 students’ solutions to many Python programming exercises. This dataset distinguishes itself through the presence of free-form assignments, enabling a broader evaluation of feedback abilities.

We follow the preprocessing approach of (Koutchme et al., 2024a) by selecting the incorrect programs from all students’ last submitted solutions (as these specific solutions often reflect students who need the most help) for each assignment that can be automatically evaluated with unit tests. Following those steps results in training, validation, and testing splits with 690, 826, and 711 incorrect programs from 44, 62 and 62 assignments.

4.2 Feedback Tasks

We train our model to generate better program repairs and aim to evaluate whether this improvement transfers to feedback generation. Specifically, we assess our models performance on three feedback types widely studied in prior work (Koutchme et al., 2025; Kotalwar et al., 2024; Phung et al., 2024; Hellas et al., 2023): explanations (\mathcal{E}), code patches (\mathcal{P}), and hints (\mathcal{H}).

Explanations identify and describe *all* issues in a student’s program, while code patches outline the necessary corrections. These two types of diagnostic feedback help students understand their mistakes after submitting an incorrect solution. Hints, in contrast, are more Socratic, guiding students toward resolving one of the issues without giving away the answer, and are most valuable while students are still actively working and may be stuck.

Quality attributes. We evaluate the generated feedback using quality criteria established in prior work. Explanations and code patches are assessed based on accuracy ($\mathcal{E}_A, \mathcal{P}_A$) and selectiveness ($\mathcal{E}_S, \mathcal{P}_S$) (Koutchme et al., 2025). Hints, are evaluated along three dimensions: correctness (\mathcal{H}_C), informativeness (\mathcal{H}_I), and concealment (\mathcal{H}_{Com}) (Phung et al., 2024). Table 3 (Appendix A) provides detailed definitions for each attribute. We later detail our evaluation strategy.

Generation strategy. When generating feedback, we always prompt our models to generate the three types of feedback sequentially: first the explanations, then the code patches, and finally a *single hint* for the first identified issue $\mathcal{F} = (\mathcal{E}, \mathcal{P}, \mathcal{H})$. This ordering draws from prior work, treating the explanation as a form of chain-of-thought reasoning (Wei et al., 2022) that supports the generation of more accurate patches (Koutchme et al., 2025) and a more helpful hint (Phung et al., 2023).

4.3 Models

We train Llama-3.2-3B (Dubey et al., 2024) and Qwen-2.5-3B (Hui et al., 2024), two language models with strong performance on programming tasks. Models in the 3B parameters range strike a practical balance: they are small enough for deployment on edge devices (Kotalwar et al., 2024) yet large enough to rapidly benefit from reinforcement learning optimization (Sui et al., 2025).

Parameter efficient finetuning. We train both models using QLoRa (Dettmers et al., 2023), a Parameter-Efficient Fine-Tuning technique (PEFT) (Houlsby et al., 2019) that quantises a language model to 4-bit and adds on top a set of trainable parameters called adapters, while the base model remains frozen. Using PEFT techniques has two benefits: quantized models allow easy edge device deployment, while adapters allow easy recovery of the base model functionalities.

4.4 Baseline and Oracle

Our objective is to assess (1) whether reasoning for repair improves programming feedback and (2) how close our DRO-trained SLMs’ can approach the performance of models having access to LLM supervision. To help answer each sub-question, we consider two LLM-distillation training references:

- **Reason-Repair (RR):** For each incorrect program s^i , we prompt an LLM to generate a corrected repair by reasoning, without producing feedback (see Figure 3, Appendix C). We then fine-tune our small models on this thought and repair $(\mathcal{T}_{LLM}^i, \mathcal{R}_{LLM}^i)$ sequence. These models allow us to evaluate whether learning to repair through thoughts using supervised fine-tuning improves feedback performance.
- **Repair-First Feedback (RFF):** We adopt the repair-before-feedback strategy from (Sahai et al., 2023; Phung et al., 2023), prompting the LLM (see Figure 4) to first generate a repair \mathcal{R}_{LLM}^i for the incorrect program as an intermediate reasoning step to produce the full feedback \mathcal{F}_{LLM}^i . While this two-step prompting strategy introduces an *implicit* form of reasoning similar to chain of thought, we note that it differs from the more *explicit* form of reasoning used in models such as DeepSeek (Luo et al., 2025). To our knowledge, such explicit reasoning has not been explored in prior work for providing feedback. Our small models are fine-tuned on these repair-feedback sequences $(\mathcal{R}_{LLM}^i, \mathcal{F}_{LLM}^i)$. Since this approach relies on direct access to feedback during training, we consider RFF-trained models as oracles, serving as a strong upper bound.

We use OpenAI GPT-4o-mini (OpenAI, 2024) as the LLM due to its strong feedback performance (Koutchme et al., 2024b) and cost efficiency.

4.5 Prompting Strategy

In our main experiments, we prompt the DRO and Reason-Repair models to generate feedback \mathcal{F} immediately (see Figure 3), without any intermediate repair step (Koutchme et al., 2025). This prompting strategy aims to study whether repair fine-tuning transfers directly into feedback improvements. In contrast, following (Sahai et al., 2023), we prompt the Repair-First Feedback models to first generate a repair for the incorrect programs before generating the final feedback \mathcal{F}^i .

We present in Appendix B the full results of our experiments, prompting all models with both strategies. The second approach evaluates whether and to what extent generating repairs before feedback effectively enhances small language models’ performance.

4.6 LLMs-as-feedback-judges

We leverage LLMs-as-judges (Zheng et al., 2023; Thakur et al., 2024) to evaluate our models. Following the jury-based approach proposed by (Verga et al., 2024), we use a panel of two language models: GPT-4o-mini and Gemini-2.0-flash (Google DeepMind, 2024). While GPT-4o-mini and Gemini-2.0-flash are lighter versions of their full-size counterparts, they remain strong judges for programming feedback. For instance, GPT-4o-mini has been shown to perform on par with GPT-4o for evaluating feedback quality (Koutchme et al., 2025). Moreover, (Verga et al., 2024) demonstrate that ensembles of smaller LLMs of different families can outperform even single large models, particularly by mitigating individual model biases.

For each feedback \mathcal{F} generated on the test set, we prompt both judges (see Figure 6, Appendix C) to provide binary decisions across all quality criteria (Table 3). We adopt a strict unanimity policy: a criterion is marked correct only if both judges agree. While this method does not provide absolute performance guarantees (see Section 6), it offers a consistent, scalable, and reliable strategy for comparing the relative effectiveness of different training approaches.

4.7 Experiment Details

We describe our experiment settings, including specific hyperparameters used for training and inference, in section A.3 (Appendix A).

5 Results

In this section, we present the results of our experiments. A more in-depth analysis, including results for the *repair-first* strategy, is provided in Appendix B. For DRO, we show results for training for one epoch (DRO-1) and two epochs (DRO-2).

5.1 Direct Repair Optimization Results

Table 1 shows the results of our experiments answering the question: *How effective is training for repair in improving feedback abilities?* We can make the following observations.

Table 1: **Feedback performance results.** We contrast DRO model performance at n training epoch (**DRO- n**) against LLM-distilled training variants RR (Reason-Repair) for two (BASE) language models, Llama-3.1-3B and Qwen-2.5-3B. We bold (resp. underline) the best (resp. second best) results.

Method	\mathcal{E}_A	\mathcal{E}_S	\mathcal{P}_A	\mathcal{P}_S	\mathcal{H}_C	\mathcal{H}_I	\mathcal{H}_{Con}
Llama-3.1-3B							
BASE	37.4	55.4	34.5	25.2	67.5	63.4	67.8
DRO-1	40.5	64.4	36.6	30.4	73.6	69.2	72.0
DRO-2	43.5	64.7	<u>40.5</u>	<u>32.2</u>	72.6	<u>67.9</u>	<u>71.9</u>
RR	42.9	<u>62.9</u>	42.9	41.8	52.7	49.4	56.3
Qwen-2.5-3B							
BASE	49.6	69.2	39.1	32.2	85.1	80.0	78.2
DRO-1	53.2	66.0	47.4	34.7	87.3	84.5	80.7
DRO-2	59.8	74.1	51.6	38.7	90.7	<u>88.3</u>	88.3
RR	<u>59.1</u>	70.0	<u>51.1</u>	38.7	91.4	90.0	<u>85.9</u>

Training for repair improves feedback abilities.

DRO improves feedback quality across all criteria for both base models. We also observe gains when training a model via supervised fine-tuning on thoughts and repairs generated by a language model (*RR*), although hint performance declines for Llama.

We interpret this success as a form of transfer learning (Raffel et al., 2020), where training on one task improves performance on another related task. Butler et al. (Butler and Winne, 1995) characterize feedback as a two-step process: first, noticing mistakes, and second, communicating them to the learner. We view program repair as a “super-task” of the noticing stage: to fix a student’s code, the model must identify what is wrong (analogous to generating an explanation) and then determine how to correct it (analogous to generating a patch). Manual inspection also suggests that most, if not all, reasoning traces produced during repair explicitly highlight both the underlying issues and their corresponding fixes.

Unlike full feedback, however, program repair does not involve pedagogical communication. In other words, it does not improve the second stage: the model’s ability to convey information effectively to a learner. Still, we believe the use of low rank adapters allows the model to preserve its original pedagogical capabilities while refining its analytical skills through targeted repair training. Lastly, since a hint is a form of non-revealing explanation, stronger repair capabilities indirectly enhance Socratic feedback.

DRO improvements scale rapidly with base model performance.

We observe that the speed and magnitude of the improvement at each training epoch n depend on the base model. Qwen, which is a stronger base model than Llama, is showing more substantial gains after just one epoch of training, and even faster gains after a second training epoch. These results align with findings from prior work (DeepSeek-AI, 2025; Luo et al., 2025), suggesting that performance improvements when training reasoning models scale faster as the quality (i.e. initial performance) of the base model increases.

DRO is competitive with LLM-distillation.

After two epochs of training (DRO-2), our models reach performance comparable to LLM-distilled variants. With Llama-3.1-3B, DRO-2 matches *RR* for generating explanations. While it underperforms on patches, it significantly outperforms *RR* on hint generation, where *RR* even falls behind the base model. With Qwen-2.5-3B, DRO-2 surpasses *RR* for generating both explanations and patches, and performs on par for writing hints.

5.2 Refining Distilled Models with Direct Repair Optimization

Table 2 shows the results of an experiment combining model distillation and reinforcement learning. Prior work highlights how SLM reasoning abilities can be bootstrapped by distilling chain-of-thoughts from an LLM, before being further enhanced through RL training (Sui et al., 2025). Following such an approach, we further fine-tuned Reason-Repair models using DRO for one epoch.

As we can observe, applying Direct Repair Optimization on top of the *RR* models consistently enhances feedback performance across almost all types of feedback for both models, allowing the *RR*-trained models to reach overall stronger performance (with the exception of a drop in Hint Concealment \mathcal{H}_{Con}). Interestingly, we observe a stronger boost in diagnostic feedback performance for our Llama *RR* model than for our Qwen model. Looking more closely, both models reach similar overall performance after RL training, suggesting a performance trade-off might be happening between diagnostic and Socratic feedback abilities. We hypothesize that this plateau may stem from the limitations of LoRA adapters, which restrict how much new “knowledge” the model can acquire (Dettmers et al., 2023).

Table 2: **Feedback performance results.** We further fine-tune the Reasoning-Repair (RR) models with DRO and show performance benefits of the resulting **COMB** models. We bold (resp. underline) the best (resp. second best) results.

Method	\mathcal{E}_A	\mathcal{E}_S	\mathcal{P}_A	\mathcal{P}_S	\mathcal{H}_C	\mathcal{H}_I	\mathcal{H}_{Com}
Model: Llama-3.2-3B							
RR	42.9	62.9	42.9	<u>41.8</u>	52.7	49.4	56.3
DRO-2	<u>43.5</u>	64.7	40.5	32.2	<u>72.6</u>	<u>67.9</u>	<u>71.9</u>
RFF	<u>43.5</u>	<u>70.9</u>	36.3	<u>40.4</u>	94.2	92.3	89.3
COMB	59.4	72.7	56.3	48.1	53.9	48.9	51.9
Model: Qwen-2.5-3B							
RR	59.1	70.0	51.1	38.7	91.4	90.0	85.9
DRO-2	59.8	<u>74.1</u>	51.6	38.5	90.7	88.3	<u>88.3</u>
RFF	72.4	82.4	62.7	50.1	94.4	91.7	89.5
COMB	<u>60.3</u>	72.6	<u>54.7</u>	<u>40.4</u>	<u>91.6</u>	91.7	85.0

5.3 Comparison Against Oracles

Table 2 also contrasts the performance of models trained with Direct Repair Optimization against the Repair-Feedback-First (RFF) models. We consider RFF models as oracles as those were trained with privileged supervision in the form of both LLM-generated feedback and repairs used as implicit reasoning steps.

Despite not using any feedback supervision, our DRO-2 models perform competitively in several areas, particularly with Llama-3B, where they closely approach RFF performance on diagnostic feedback tasks. Moreover, our combined approach (COMB), which applies DRO fine-tuning on top of LLM-distilled models, surpasses RFF on several criteria for Llama, including both explanation and patch quality. However, for Qwen, both DRO and COMB consistently fall short of RFF performance across most evaluation metrics.

These findings suggest that while DRO can serve as a valuable complement to LLM supervision, and even outperform direct finetuning in some settings, it is not yet a reliable substitute for training models directly on feedback generated by LLMs. Further work is needed to understand when and how DRO can consistently match or exceed the performance of supervised approaches.

6 Discussion and Conclusion

In this paper, we explored whether training language models to reason about students’ programs could improve feedback and provided insights into how this strategy compares to LLM supervision.

Summarizing answers to our research question.

Our findings show that (1) reasoning to repair programs improves a model’s ability to generate feedback, (2) DRO can further improve models fine-tuned on LLM-generated repairs, and (3) such refined models can, in some instances, match the performance of models trained directly on LLM-generated feedback.

Implications for programming education.

Programming education is increasingly turning to open-source language models, particularly smaller ones, to support teaching at scale (Liu et al., 2024b). We are anticipating a shift from using proprietary LLMs (e.g., GPT-4o) to open-source alternatives (e.g., LLaMA-3.3-70B) for distillation (Kotalwar et al., 2024), as the latter can be hosted locally and offer more control (Denny et al., 2024). Our work takes a step further by exploring whether we can eliminate reliance on LLMs and train SLMs directly on educational data, avoiding both the costs of third-party APIs and the computational demands of hosting large open-source models.

Although our work focuses on programming data, we believe DRO could be adapted to provide feedback in all educational domains where the correctness of a student’s work can be automatically evaluated. Methods such as Direct Repair Optimization can leverage much of the readily available data in educational platforms (i.e., student submissions and unit tests) without requiring extensive curation. Recent work has shown that combining such reinforcement learning methods with large-scale data can allow relatively small models to reach the performance of state-of-the-art proprietary models (Luo et al., 2025).

Extensions to other training approaches.

Direct Repair Optimization can easily be combined with human and LLM-supervised training strategies. Models trained with DRO can be bootstrapped from a handful of high-quality LLM-generated examples (Hicke et al., 2023; Ashok Kumar and Lan, 2024; Muennighoff et al., 2025), and further refined using Reinforcement Learning from Human Feedback (RLHF) (Woodrow et al., 2025) to align with specific instructional goals. Such hybrid pipelines offer a practical path forward, starting from refined educational data, scaling up performance through large-scale reinforcement learning, and applying targeted human supervision as a final step to meet specific classroom needs.

Alleviating privacy concerns. Although third-party LLM hosting services and the use of proprietary APIs are becoming more affordable, institutional policies on sending student data to third-party services can restrict their use. Our experiments show that institutions with access to modest computational power (such as a single consumer-grade GPU)² can obtain powerful programming teaching assistant models tailored to their classes.

Such models can also be directly deployed on students' laptops (Liu et al., 2024b; Kotalwar et al., 2024; Ruan et al., 2024), enabling personalized, timely, and offline support.

Future work. Our future work will explore how Direct Repair Optimization performs compared to proprietary and open-source LLMs when trained on large-scale private educational programming data as well as public programming data from HuggingFace³. To this end, we plan to conduct human expert evaluations and perform A/B studies to evaluate how real students respond to such feedback (SLM vs LLMs). We will also investigate how human data and preferences can be integrated into the training pipeline to better align small models with specific institutional goals.

Looking ahead, we aim to move beyond training individual models on private institutional data and tackle the broader challenge of building foundation models for programming education (Bommasani et al., 2022). We believe such models could be pre-trained from publicly open-source large-scale ethical data, and further refined with federated learning across multiple institutions.

Limitations

Our study is not without limitations. First, we conducted all experiments on a single dataset of Python programming submissions collected from one institution and did not explore whether our results hold in other contexts. Second, and perhaps more importantly, our evaluation lacks human annotations, expert assessment, or qualitative analysis. While prior work suggests LLMs can be used to assess programming feedback (Seo et al., 2025; Koutchme et al., 2024b, 2025), such works also highlight that their judgments are not always perfect. Although we partly mitigate this by combining multiple LLMs-as-judges, our results must still be interpreted with caution.

²We trained our models on a single 32GB VRAM GPU.

³<https://huggingface.co/datasets>

We do not claim that DRO-trained models produce feedback that meets any absolute standard of quality (e.g., “nearly perfect feedback”). Rather, our findings establish DRO’s relative performance: it improves feedback quality over a base untrained model and can match the performance of models trained via LLM distillation. Whether such feedback is ultimately pedagogically effective for students remains an open question until validated through human studies.

Additionally, our experiments were limited to two small models with around 3B parameters. While prior work suggests that performance improves with base model size (Sui et al., 2025), it remains to be seen whether the same trends hold when applying Direct Repair Optimization for improving other language models’ programming feedback. Moreover, our experiments also did not include new state-of-the-art reasoning large language models such as OpenAI o3. Such models, which were effectively trained for reasoning, would probably act as better candidates for LLM-distillation and combined LLM-distillation and RL training.

Ethics Statement

This work has been conducted in accordance with national and institutional ethical guidelines. We recognize the growing importance of ethical considerations in AI research, particularly with respect to data use, model deployment, and societal impact.

The dataset used in this study is publicly available to the research community. Our primary goal is to advance the development and evaluation of open-source language models for feedback generation in programming education. By prioritizing open-source models, we aim to promote transparency, accessibility, and accountability, while mitigating privacy concerns associated with proprietary LLMs.

We also acknowledge broader ethical dimensions of our work. These include questions of fairness and equity in access to high-quality feedback, the risk that language models may favor certain interaction styles or learner backgrounds, and the potential for such technologies to either reduce or exacerbate global disparities in education. As the use of LLMs in learning environments grows, we believe it is essential to continuously assess and address these challenges in collaboration with educators, institutions, and affected communities.

References

- Umair Z. Ahmed, Shubham Sahai, Ben Leong, and Amey Karkare. 2025. [Feasibility study of augmenting teaching assistants with ai for cs1 programming feedback](#). In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1, SIGCSETS 2025*, page 11–17, New York, NY, USA. Association for Computing Machinery.
- 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.
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, et al. 2022. [On the opportunities and risks of foundation models](#).
- Deborah L. Butler and Philip H. Winne. 1995. [Feedback and self-regulated learning: A theoretical synthesis](#). *Review of Educational Research*, 65(3):245–281.
- Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. 2025. [Security and privacy challenges of large language models: A survey](#). *ACM Comput. Surv.*, 57(6).
- Adrian de Freitas, Joel Coffman, Michelle de Freitas, Justin Wilson, and Troy Weingart. 2023. [Falconcode: A multiyear dataset of python code samples from an introductory computer science course](#). In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, SIGCSE 2023*, page 938–944, New York, NY, USA. Association for Computing Machinery.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#).
- Paul Denny, James Prather, Brett A. Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N. Reeves, Eddie Antonio Santos, and Sami Sarsa. 2024. [Computing education in the era of generative ai](#). *Commun. ACM*, 67(2):56–67.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, page 441, Red Hook, NY, USA. Curran Associates Inc.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and Aiesha Letman et al. 2024. [The llama 3 herd of models](#).
- Google DeepMind. 2024. Gemini 2.0: Our largest and most capable AI model. <https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/>. Accessed: 2025-04-24.
- John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research*, 77(1):81–112.
- Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutcheme, Lilja Kujanpää, and Juha Sorva. 2023. [Exploring the responses of large language models to beginner programmers' help requests](#). In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1, ICER '23*, page 93–105, New York, NY, USA. Association for Computing Machinery.
- Yann Hicke, Anmol Agarwal, Qianou Ma, and Paul Denny. 2023. [Ai-ta: Towards an intelligent question-answer teaching assistant using open-source llms](#).
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.
- Johan Jeuring, Hieke Keuning, Samiha Marwan, Dennis Bouvier, Cruz Izu, Natalie Kiesler, Teemu Lehtinen, Dominic Lohr, Andrew Peterson, and Sami Sarsa. 2022. [Towards giving timely formative feedback and hints to novice programmers](#). In *Proceedings of the 2022 Working Group Reports on Innovation and Technology in Computer Science Education, ITiCSE-WGR '22*, page 95–115, New York, NY, USA. Association for Computing Machinery.
- Harshit Joshi, José Pablo Cambronero Sánchez, Sumit Gulwani, Vu Le, Gust Verbruggen, and Ivan Radicek. 2023. [Repair is nearly generation: Multilingual program repair with llms](#). In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 5131–5140. AAAI Press.
- Nachiket Kotalwar, Alkis Gotovos, and Adish Singla. 2024. [Hints-in-browser: Benchmarking language models for programming feedback generation](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Charles Koutcheme. 2022. [Towards open natural language feedback generation for novice programmers using large language models](#). In *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research, Koli Calling '22*, New

- York, NY, USA. Association for Computing Machinery.
- Charles Koutcheme. 2023. Training Language Models for Programming Feedback Using Automated Repair Tools. In *Artificial Intelligence in Education*, pages 830–835, Cham. Springer Nature Switzerland.
- Charles Koutcheme, Nicola Dainese, and Arto Hellas. 2024a. 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, 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*, SIGCSETS 2025, page 624–630, New York, NY, USA. Association for Computing Machinery.
- Charles Koutcheme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, and Paul Denny. 2024b. Open source language models can provide feedback: Evaluating llms’ ability to help students using gpt-4-as-a-judge. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education, Volume 1*, ITICSE ’24.
- Charles Koutcheme, Nicola Dainese, Sami Sarsa, Juho Leinonen, Arto Hellas, and Paul Denny. 2024c. Benchmarking educational program repair.
- Charles Koutcheme, Sami Sarsa, Juho Leinonen, Lassi Haaranen, and Arto Hellas. 2023a. Evaluating distance measures for program repair. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, ICER ’23, page 495–507, New York, NY, USA. Association for Computing Machinery.
- Charles Koutcheme, Sami Sarsa, Juho Leinonen, Arto Hellas, and Paul Denny. 2023b. Automated Program Repair Using Generative Models for Code Infilling. In *Artificial Intelligence in Education*, pages 798–803, Cham. Springer Nature Switzerland.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Mark Liffiton, Brad E Sheese, Jaromir Savelka, and Paul Denny. 2024. Codehelp: Using large language models with guardrails for scalable support in programming classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research, Koli Calling ’23*, New York, NY, USA. Association for Computing Machinery.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J. Malan. 2024a. 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. 2*, SIGCSE 2024, page 1927, New York, NY, USA. Association for Computing Machinery.
- Suqing Liu, Zezhu Yu, Feiran Huang, Yousef Bulbulia, Andreas Bergen, and Michael Liut. 2024b. Can small language models with retrieval-augmented generation replace large language models when learning computer science? In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, ITICSE 2024, page 388–393, New York, NY, USA. Association for Computing Machinery.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding rl-zero-like training: A critical perspective.
- Dominic Lohr, Hieke Keuning, and Natalie Kiesler. 2025. You’re (not) my type—can llms generate feedback of specific types for introductory programming tasks? *Journal of Computer Assisted Learning*, 41(1):2025.
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.
- Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. Introductory programming: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITICSE 2018 Companion, page 55–106, New York, NY, USA. Association for Computing Machinery.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling.
- OpenAI. 2024. Gpt-4o system card.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al.

2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- José Carlos Paiva, José Paulo Leal, and Álvaro Figueira. 2022. Automated assessment in computer science education: A state-of-the-art review. *ACM Trans. Comput. Educ.*, 22(3).
- Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Tung Phung, JosÃ© 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. In *Proceedings of the 16th International Conference on Educational Data Mining*, pages 370–377, Bengaluru, India. International Educational Data Mining Society.
- 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*, page 12–23, New York, NY, USA. Association for Computing Machinery.
- Thomas W. Price, Rui Zhi, and Tiffany Barnes. 2017. Evaluation of a data-driven feedback algorithm for open-ended programming. In *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017, Wuhan, Hubei, China, June 25-28, 2017*. International Educational Data Mining Society (IEDMS).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).
- Kelly Rivers and Kenneth R Koedinger. 2017. Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education*, 27:37–64.
- Charlie F Ruan, Yucheng Qin, Xun Zhou, Ruihang Lai, Hongyi Jin, Yixin Dong, Bohan Hou, Meng-Shiun Yu, Yiyan Zhai, Sudeep Agarwal, et al. 2024. Weblm: A high-performance in-browser llm inference engine. *arXiv preprint arXiv:2412.15803*.
- Shubham Sahai, Umair Z. Ahmed, and Ben Leong. 2023. Improving the coverage of gpt for automated feedback on high school programming assignments.
- Alexander Scarlato, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. *Improving the Validity of Automatically Generated Feedback via Reinforcement Learning*, page 280–294. Springer Nature Switzerland.
- Hyein Seo, Taewook Hwang, Jeesu Jung, Hyeonseok Kang, Hyuk Namgoong, Yohan Lee, and Sangkeun Jung. 2025. Large language models as evaluators in education: Verification of feedback consistency and accuracy. *Applied Sciences*, 15:671.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.
- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models.
- Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2024. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges.
- Annapurna Vadaparty, Daniel Zingaro, David H. Smith IV, Mounika Padala, Christine Alvarado, Jamie Gorson Benario, and Leo Porter. 2024. Cs1-llm: Integrating llms into cs1 instruction. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1, ITiCSE 2024*, page 297–303, New York, NY, USA. Association for Computing Machinery.
- Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models.
- Arto Vihavainen, Jonne Airaksinen, and Christopher Watson. 2014. A systematic review of approaches for teaching introductory programming and their influence on success. In *Proceedings of the Tenth Annual Conference on International Computing Education Research, ICER '14*, page 19–26, New York, NY, USA. Association for Computing Machinery.
- Sierra Wang, John Mitchell, and Chris Piech. 2024. A large scale rct on effective error messages in cs1. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1, SIGCSE 2024*, page 1395–1401, New York, NY, USA. Association for Computing Machinery.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Juliette Woodrow, Sanmi Koyejo, and Chris Piech. 2025. Improving generative ai student feedback: Direct preference optimization with teachers in the loop. https://juliettewoodrow.github.io/paper-hosting/dpo_feedback.pdf. Accessed: 2025-04-12.

Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, et al. 2025a. [Dapo: An open-source llm reinforcement learning system at scale](#).

Ze Zhu Yu, Suqing Liu, Paul Denny, Andreas Bergen, and Michael Liut. 2025b. [Integrating small language models with retrieval-augmented generation in computing education: Key takeaways, setup, and practical insights](#). In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1, SIGCSETS 2025*, page 1302–1308, New York, NY, USA. Association for Computing Machinery.

Jialu Zhang, José Cambronero, Sumit Gulwani, Vu Le, Ruzica Piskac, Gustavo Soares, and Gust Verbruggen. 2022. [Repairing bugs in python assignments using language models](#).

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. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.

A Methodological and Experimental Details

A.1 Training loss

The original GRPO loss function is given by:

$$\mathcal{J}_{\text{GRPO}} = -\frac{1}{\sum_{g=1}^G |o_g|} \sum_{g=1}^G \sum_{t=1}^{|o_g|} l_{g,t} - \beta \text{KL}(\pi_{\theta}, \pi_{\text{ref}})$$

where

$$l_{g,t} = \frac{\pi_{\theta}(o_{g,t} \mid d^i, s^i, o_{g,<t})}{\pi_{\theta_{\text{old}}}(o_{g,t} \mid d^i, s^i, o_{g,<t})} \hat{A}_{g,t}$$

and \hat{A}^i is a value called the advantage. Intuitively, the advantage tells each generation how much better it is than the $g - 1$ other generations.

Compared to popular offline preference methods such as DPO (Rafailov et al., 2024), which use explicit preference pairs, the advantage function helps "ranking" which of the multiple generations, without relying on pairwise comparisons.

$$\hat{A}_g = \frac{(r_g^i - \bar{r})}{\text{std}(r)}$$

$\text{KL}(\pi_{\theta}, \pi_{\text{ref}})$ is a value called the KL divergence. This value essentially tells how much the model responses are diverging from the model prior to the start of training. We omit the definition of this term for simplicity and refer the reader to the DeepSeek paper (DeepSeek-AI, 2025). In essence, $\mathcal{J}_{\text{GRPO}}$ is the weighted average of the advantage of all completions and a β scaled approximation of the KL divergence.

The following works have found a few issues with the original formulation.

Removing the KL term. (Yu et al., 2025a) finds that the $\text{KL}(\pi_{\theta}, \pi_{\text{ref}})$ term can slow down training, as in practice we want to allow the trained model to diverge from the original policy.

Length response bias. (Liu et al., 2025) show that the term $-\frac{1}{\sum_{g=1}^G |o_g|}$ introduces a *response length bias* favourizing longer generations. To address this issue, the authors propose dividing by a constant length, being the maximum allowed size of each generation (LG).

Program difficulty bias. Moreover, they also show that the standard deviation in the advantage computation $\hat{A}_g = \frac{(r_g^i - \bar{r})}{\text{std}(r)}$ introduces a *problem difficulty bias*, where overly hard or overly easy questions are weighted more heavily in the loss. In our situation, the "problem" is the student program to solve, and this bias would lead to scenarios where student programs which are too easy to fix or student programs which are too hard to solve would be given more attention. Removing the standard deviation addresses this issue. Taking these two changes into account yields the proposed loss function (see equation 1).

Taking into account multiple updates. Because sampling generations is computationally and time-intensive, in practice, we use a version of this loss function which takes into account multiple updates per generation, as proposed by (Yu et al., 2025a):

$$\mathcal{J}_{\text{dr.GRPO}} = -\frac{1}{\sum_{g=1}^G |o_g|} \sum_{g=1}^G \sum_{t=1}^{|o_g|} \left[\min \left(l_{g,t}, \hat{C}_g \hat{A}_g \right) \right]$$

where

$$\hat{C}_g = \text{clip}(l_{g,t}, 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}})$$

constrains the subsequent updates to stay within a reasonable range of the original policy.

A.2 Feedback Quality Attributes.

Table 3 shows the definitions of the feedback quality criteria used in our work. These definitions are taken from prior work in programming feedback.

A.3 Experimental Details

We outline training and inference-specific details.

A.3.1 Training

We train our models using the HuggingFace TRL library. Unless explicitly outlined below, all hyperparameters were left at default values. We train all models with QLoRa (Dettmers et al., 2023) using an alpha $\alpha = 128$ and a rank $r = 128$. All models are trained on a single NVIDIA V100 GPU using our institution’s research cluster. Training for one epoch on such compute takes approximately 8 hours. Training on an A100 takes less than 5 hours.

dr.GRPO specific hyperparameters. Table 4 shows the hyperparameters used to train our DRO model. These parameters follow prior work (DeepSeek-AI, 2025; Luo et al., 2025; Yu et al., 2025a). We train all models for two epochs on the training set of FalconCode. For each incorrect program, we generate four ($G = 4$) candidate reasoning and repairs $\mathcal{G}^i = (\mathcal{T}_i^i, \mathcal{R}_i^i)$. We highlight that we designed our method to run on an entry-level GPU with 32GB of RAM. Prior work (DeepSeek-AI, 2025) suggests that a higher number can substantially improve results, however, more generations require more GPU RAM.

Supervised Fine-tuning. Table 5 shows the hyperparameters used to train our distilled models via supervised fine-tuning on the training set of FalconCode. We train all models for three epochs.

A.3.2 Inference

For generating feedback at inference time, for all models, we generate both repair and feedback using greedy decoding. For judging, we query proprietary models GPT-4o-mini and Gemini-2.0-flash using the OpenAI Python API, also using greedy decoding.

B Results Details

Table 6 shows the full results of all our experiments, including the repair-first prompting strategy. We additionally report the performance of our models on additional clarity criteria.

B.1 Prompting for Repair

Prompting for repair often decreases base model performance. Prompting the *BASE* models to generate a repair before producing feedback decreases diagnostic feedback performance. This observation aligns with findings from (Koutcheme et al., 2024a), who showed that a model’s ability to provide diagnostic feedback scales independently from its ability to perform program repair. In our case, using SLMs, a poor-quality greedy repair may degrade feedback quality more than providing no repair. This does not contradict (Sahai et al., 2023), as the authors use the Repair-First strategy with LLMs, which are strong at both repair and feedback. (Phung et al., 2023, 2024) extend the single-repair strategy with multiple repairs, but whether an SLM benefits from this is unknown. While it might alleviate this issue, it remains computationally expensive when running models locally.

Prompting to repair before feedback also decreases Socratic feedback performance for Qwen but increases it for Llama.

Prompting for repair brings benefits in diagnostic feedback performance for strong base models. Prompting to repair before feedback decreases the performance of the Llama DRO-trained models for generating diagnostic feedback but increases diagnostic feedback performance for Qwen models. We hypothesize that this effect is due to the base model overfitting on low-quality, greedy-generated repairs. We observe the same phenomenon for the RFF models, which were trained to repair before feedback: a decrease for Llama, but an increase for Qwen. For RR and their extended version with DRO (COMB), the effect is unclear.

B.2 Generations Clarity

We also studied how language models perform in terms of the clarity (Cle) of the generations. However, we mostly observe that there does not seem to be a clear correlation with base model performance, training method, or prompting strategy.

Name	Notation	Definition	Used in
Accuracy	$\mathcal{E}_A, \mathcal{P}_A$	All issues in the student’s code (or all required fixes) are correctly identified.	(Koutcheme et al., 2024b, 2025)
Selectiveness	$\mathcal{E}_S, \mathcal{P}_S$	No non-existent or irrelevant issues are mentioned; no unnecessary changes are proposed.	(Koutcheme et al., 2024b, 2025)
Clarity	$\mathcal{E}_C, \mathcal{P}_C$	The explanation or patch is easy to understand, well-formatted, and concise.	(Koutcheme et al., 2025)
Correctness	\mathcal{H}_C	The hint provides correct information that would help fix the student’s code.	(Phung et al., 2024; Kotalwar et al., 2024)
Informativeness	\mathcal{H}_I	The hint contains useful information that helps the student understand or resolve the issue.	(Phung et al., 2024; Kotalwar et al., 2024)
Concealment	\mathcal{H}_{Con}	The hint avoids revealing the full solution and encourages reasoning.	(Phung et al., 2024; Kotalwar et al., 2024)
Clarity	\mathcal{H}_{Cle}	The hint is clearly written, easy to read, and free of unnecessary complexity.	(Phung et al., 2024; Kotalwar et al., 2024)

Table 3: Feedback quality attributes used in this study, taken from prior work.

Table 4: GRPO training hyperparameters.

Hyperparameter	Value
Learning rate	1e-6
Epochs	2
Warmup ratio	0.1
Max gradient norm	0.2
Scheduler type	constant_with_warmup
Optimizer	paged_adamw_8bit
Gradient checkpointing	True
Batch size	2
Max prompt length	512
Max completion length (LG)	1512
GRPO-specific	
Num generations	4
Num iterations	2
Epsilon	0.2
Epsilon high	0.28
Top-p	0.95
Temperature	0.7
Model settings	
Precision	fp16
LoRA config	
LoRA rank (r)	128
LoRA alpha	128

Table 5: Supervised Fine-Tuning hyperparameters.

Hyperparameter	Value
Learning rate	1e-4
Epochs	3
Warmup ratio	0.1
Scheduler type	cosine
Batch size	8
Model settings	
Precision	fp16
LoRA config	
LoRA rank (r)	128
LoRA alpha	128

C Prompts

This section shows all the prompts used in our study.

Table 6: **Feedback performance results.** We contrast DRO model performance at n training epoch (**DRO- n**) against LLM-distilled training variants RR (Reason-Repair) and RFF (Repair First then Feedback), as well as the RR further fine-tuned with DRO(COMB), for two (BASE) language models, Llama-3.1-3B and Qwen-2.5-3B, for two prompting strategies: Direct Feedback and Repair First.

Method	Llama-3.1-3B										Qwen-2.5-3B									
	\mathcal{E}_A	\mathcal{E}_S	\mathcal{E}_{Cle}	\mathcal{P}_A	\mathcal{P}_S	\mathcal{P}_{Cle}	\mathcal{H}_C	\mathcal{H}_I	\mathcal{H}_{Com}	\mathcal{H}_{Cle}	\mathcal{E}_A	\mathcal{E}_S	\mathcal{E}_{Cle}	\mathcal{P}_A	\mathcal{P}_S	\mathcal{P}_{Cle}	\mathcal{H}_C	\mathcal{H}_I	\mathcal{H}_{Com}	\mathcal{H}_{Cle}
Prompting Strategy: Direct Feedback																				
BASE	37.4	55.4	61.9	34.5	25.2	74.7	67.5	63.4	67.8	72.7	49.6	69.2	60.2	39.1	32.2	68.4	85.1	80.0	78.2	80.7
DRO-1	40.5	64.4	62.7	36.6	30.4	75.5	73.6	69.2	72.0	77.6	53.2	66.0	65.3	47.4	34.7	71.4	87.3	84.5	80.7	83.1
DRO-2	43.5	64.7	58.2	40.5	32.2	72.7	72.6	67.9	71.9	75.8	59.8	74.1	59.5	51.6	38.7	68.6	90.7	88.3	88.3	80.3
RR	42.9	62.9	65.7	42.9	41.8	72.4	52.7	49.4	56.3	59.6	59.1	70.0	59.5	51.1	38.7	75.8	91.4	90.0	85.9	86.6
RFF	54.9	76.9	56.8	44.6	40.8	66.1	90.3	86.1	87.2	84.8	52.9	65.8	66.2	44.9	30.8	68.6	91.6	89.3	88.5	81.0
COMB	59.4	72.7	64.4	56.3	48.1	70.5	53.9	48.9	51.9	54.9	60.3	72.6	62.6	54.7	40.4	74.7	91.6	91.7	85.0	86.5
Prompting Strategy: Repair First																				
BASE	29.0	56.0	58.5	18.7	19.5	51.5	83.0	79.0	82.3	77.8	34.7	57.7	65.8	26.6	24.5	61.6	79.5	74.1	73.1	81.9
DRO-1	38.0	71.7	60.1	26.9	31.9	49.8	86.6	82.3	85.9	78.6	61.2	76.5	60.5	51.1	41.6	61.7	90.0	84.1	84.4	78.3
DRO-2	33.5	69.6	66.0	24.8	32.9	51.5	88.2	83.8	89.2	82.8	63.9	82.4	63.7	49.6	45.9	57.7	89.7	83.0	86.8	79.0
RR	47.7	76.1	60.3	39.1	41.6	56.4	93.1	88.7	90.3	82.0	46.7	66.2	58.8	40.6	35.6	66.0	88.5	82.3	80.7	82.1
RFF	43.5	70.9	67.4	36.3	40.4	56.0	94.2	92.3	89.3	88.0	72.4	82.4	68.2	62.7	50.1	63.4	94.4	91.7	89.5	79.7
COMB	50.9	79.3	57.9	44.3	46.3	50.6	94.0	91.7	90.2	82.8	56.8	73.4	58.5	47.7	40.1	64.0	89.0	87.2	81.9	84.2

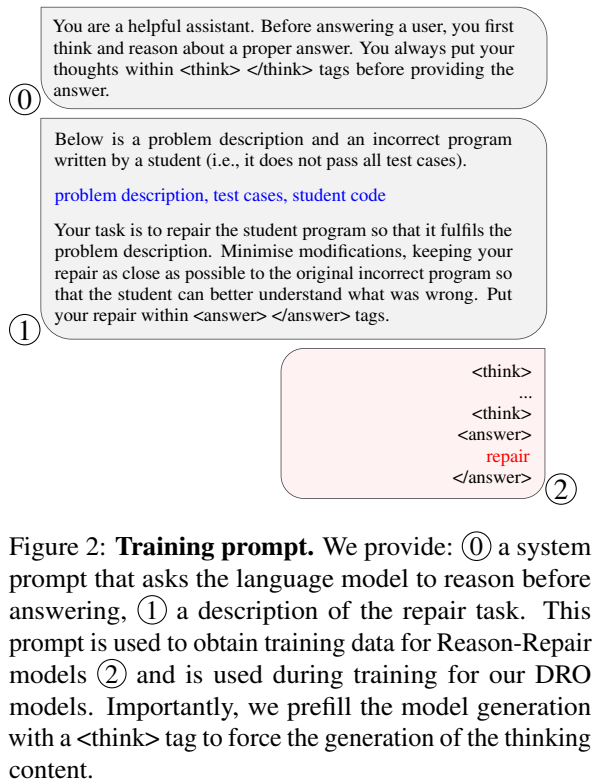


Figure 2: **Training prompt.** We provide: ① a system prompt that asks the language model to reason before answering, ② a description of the repair task. This prompt is used to obtain training data for Reason-Repair models ③ and is used during training for our DRO models. Importantly, we prefill the model generation with a <think> tag to force the generation of the thinking content.

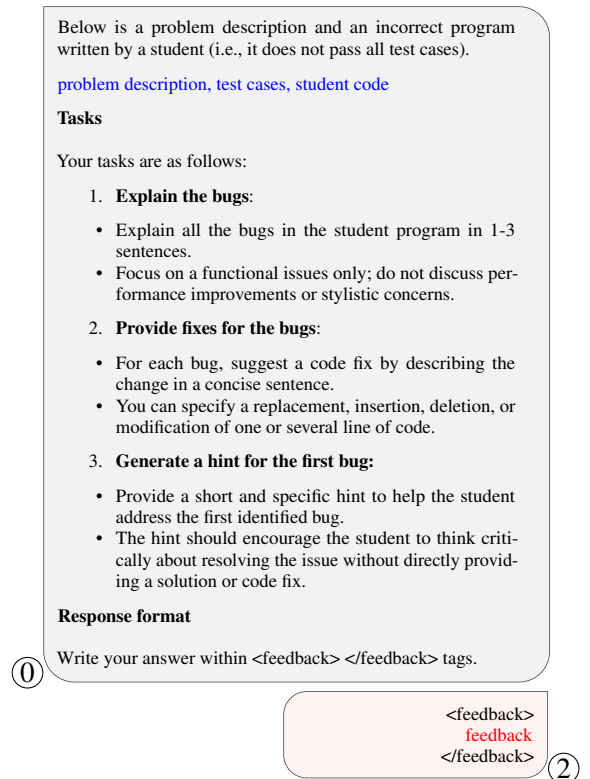


Figure 3: **Generation prompt: Providing direct feedback.** We provide: ① a description of the repair task, and ask the language models to generate feedback ②.

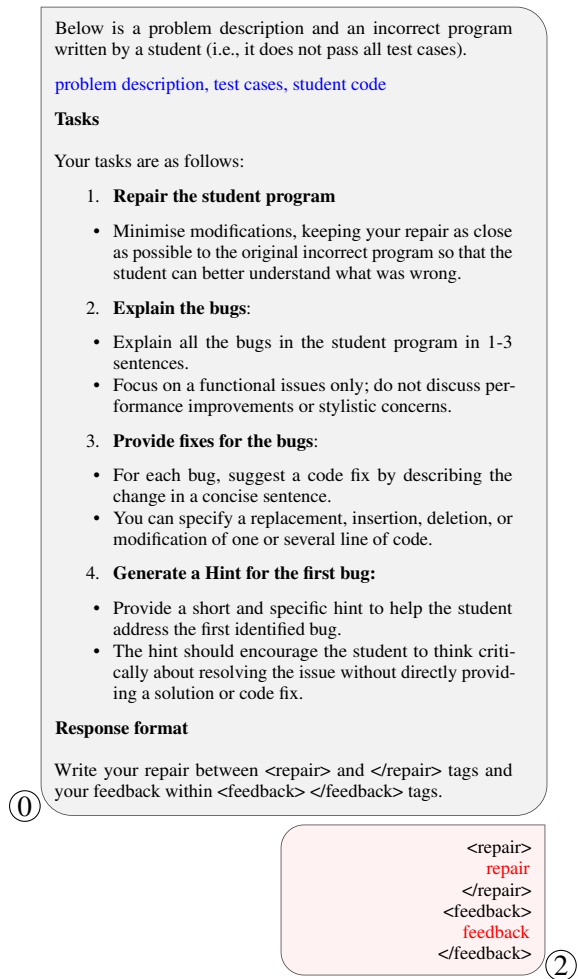


Figure 4: **Generation and training prompt for supervised finetuning.** We provide: ① a description of the repair task, and ② ask the LLM models to generate feedback ③. The full completion is then learned by the Repair-First-Feedback models using supervised finetuning.

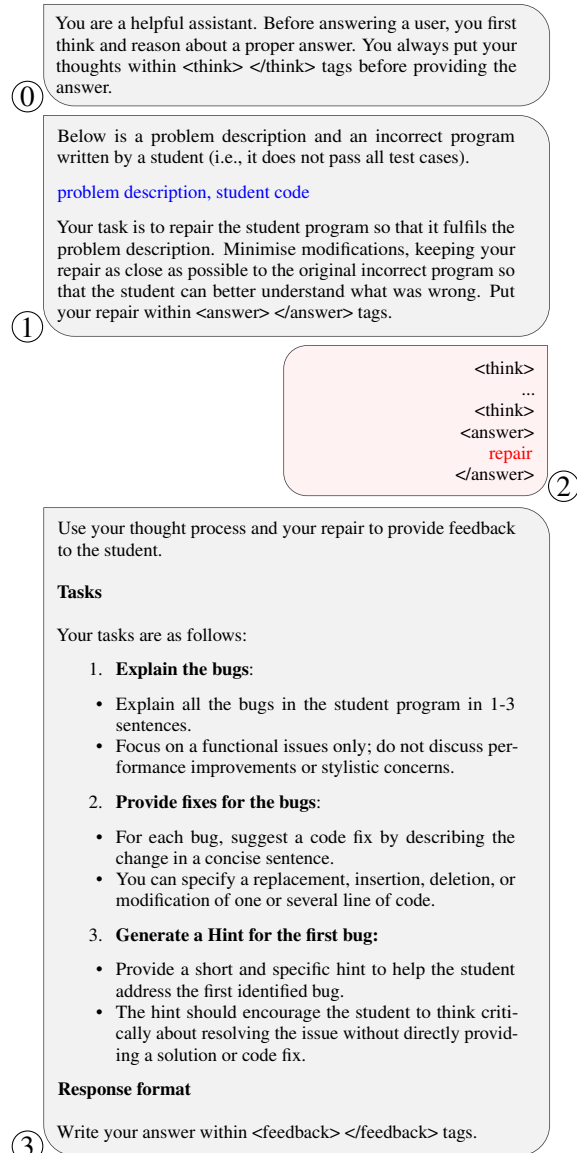


Figure 5: **Inference prompt: Repair before feedback.** We provide: ① a system prompt that asks the language model to reason before answering (only provided for DRO and Reason-Repair models), ② a description of the repair task. We obtain the model generation ③, and then in the following turn (④) ask the model to generate feedback.

① You are a computer science professor teaching introductory programming using Python. You are an expert at evaluating programming feedback tailored to novices.

Below is a problem description and an incorrect program written by a student (i.e., it does not pass all test cases).

[problem description](#)>, [student code](#)

Below is the feedback written by a teaching assistant (TA), which includes an explain and fixes for the bugs in the program. As well as a hint for the first bug.

[feedback](#)

Your task is to evaluate the quality of the TA's feedback according to the grading criteria outlined below.

[grading criteria](#)

This evaluation will be conducted in two parts

1. Reasoning: Reflect on the quality of the TA's feedback.
 - Reflect on the quality of the feedback, using the grading criteria as a guide.
 - Discuss strengths and weaknesses in the explanation and hint.
2. Grading List: Conclude with your final assessment for each criterion.
 - If the criterion is fully met, respond with "true"; otherwise, respond with "false".

Please provide your answer using a JSON format with two keys:

- "reasoning": your detailed written analysis
- "grading": a dictionary with each criterion as a key and your final answer (true or false) as the value.

Use only true or false (no other qualifiers) for each grading criterion in the JSON output.

①

Figure 6: **Judging prompt.** We provide our three LLM judges with a ① system description describing their role, ① a description of the judging task, and the specification of the response format in json.

Analyzing Interview Questions via Bloom’s Taxonomy to Enhance the Design Thinking Process

Fatemeh Kazemi Vanhari and Christopher Anand and Charles Welch

Department of Computing and Software

McMaster University

Hamilton, Ontario, Canada

{kazemivf, anandc, cwelch}@mcmaster.ca

Abstract

Interviews are central to the Empathy phase of Design Thinking, helping designers uncover user needs and experience. Although interviews are widely used to support human-centered innovation, evaluating their quality, especially from a cognitive perspective, remains underexplored. This study introduces a structured framework for evaluating interview quality in the context of Design Thinking, using Bloom’s Taxonomy as a foundation. We propose the Cognitive Interview Quality Score, a composite metric that integrates three dimensions: Effectiveness, Bloom Coverage, and Distribution Balance Score. Using human-annotations, we assessed 15 interviews across three domains to measure cognitive diversity and structure. We compared CIQS-based rankings with human experts and found that the Bloom Coverage Score aligned more closely with expert judgments. We evaluated the performance of LMA-3-8B-Instruct and GPT-4o-mini, using zero-shot, few-shot, and chain-of-thought prompting, finding GPT-4o-mini, especially in zero-shot mode, showed the highest correlation with human annotations in all domains. Error analysis revealed that models struggled more with mid-level cognitive tasks (e.g., Apply, Analyze) and performed better on Create, likely due to clearer linguistic cues. These findings highlight both the promise and limitations of using NLP models for automated cognitive classification and underscore the importance of combining cognitive metrics with qualitative insights to comprehensively assess interview quality.

1 Introduction

Design Thinking is a widely adopted framework for creative problem-solving, particularly in areas that require deep user understanding and human-centered innovation. It typically progresses through five iterative stages: Empathize, Define, Ideate, Prototype, and Test. At the heart of this process

is the first stage, “Empathize”, enabling designers to deeply understand users’ experiences, emotions, and needs. It distinguishes Design Thinking from purely analytical approaches by emphasizing a human-centered perspective. This phase often involves interviews, observations, and immersive techniques, such as simulating real user experiences, to uncover pain points and inform meaningful design interventions (Brown, 2009; Org, 2015). Among these methods, interviews play a vital role by fostering open-ended, direct dialogue between researchers and users. The quality of the interview questions during this phase is especially critical, as it shapes the depth, clarity, and diversity of responses, and ultimately influences the effectiveness of the application’s design.

Despite the central role of interviews, there is limited systematic guidance on how to structure interview questions to encourage deeper cognitive engagement. Many interviews rely on intuitive or ad hoc question writing, often leading to unbalanced questioning that skews toward lower-order thinking, such as remembering or understanding, while neglecting higher-order processes such as analyzing, evaluating, and creating (Anderson and Krathwohl, 2001).

To address this gap, we propose a novel approach that takes advantage of Bloom’s Taxonomy, a widely used hierarchical framework that classifies cognitive tasks into six categories: Remember, Understand, Apply, Analyze, Evaluate, and Create (Bloom, 1956; Anderson and Krathwohl, 2001). Originally developed for educational settings, Bloom’s Taxonomy has been effectively adapted in recent years for use in question generation (Hwang et al., 2023), question classification (Mohammed and Omar, 2018; Gani et al., 2023), and curriculum evaluation (and, 2002). In this study, we apply our approach to the domain of interview question design within the context of Design Thinking. Specifically, we investigate whether

covering different levels of Bloom’s Taxonomy in interview questions and responses has an effect on the overall quality of interviews and contributes to enhancing the Design Thinking process. Our analysis covers three interview subjects: AI Regulation, Math Visualizer, and Grandfather Game.

We use Large Language Models (LLMs) to automatically classify both interview questions and their responses according to Bloom’s Taxonomy. Leveraging recent advances in prompt engineering techniques including zero-shot (Brown et al., 2020), few-shot (Liu et al., 2023), and chain-of-thought (CoT) prompting (Wei et al., 2022), we use LLaMA-3-8B-Instruct, an instruction-tuned open-source LLM, and GPT-4o-mini, a lightweight proprietary model from OpenAI optimized for fast reasoning tasks, to assign Bloom levels based on the cognitive demands of the text. We also introduce a composite evaluation metric, the Cognitive Interview Quality Score (CIQS), which integrates Bloom effectiveness, coverage, and distribution balance scores into a single measure to assess the overall quality of interview questions.

To guide our investigation into the cognitive quality of interviews and the role of automated classification, this study is driven by the following research questions:

RQ1: Does covering multiple levels of Bloom’s Taxonomy in interview questions and responses contribute to higher-quality interviews within the Design Thinking process?

RQ2: Can LLMs, such as LLaMA-3-8B-Instruct and GPT-4o-mini, reliably classify interview content into Bloom’s cognitive levels across different prompting strategies?

RQ3: To what extent do our CIQS-based automated rankings of interview quality align with expert human evaluations across diverse interview subjects?

2 Related Work

2.1 Automated Classification of Questions Using Bloom’s Taxonomy

Bloom’s Taxonomy, originally introduced by Bloom (1956) and later revised by Anderson and Krathwohl (2001), has long served as a framework for classifying learning objectives and designing educational assessments. Numerous studies have leveraged this taxonomy to guide the construction of questions that effectively target various cognitive levels, from simple

recall (Remember) to complex creative tasks (Create). Chang and Chung (2009) developed a keyword-based system aimed at automatically classifying teachers’ questions according to Bloom’s Taxonomy. By constructing a dictionary that maps specific keywords to corresponding cognitive levels, their system achieved a 75% accuracy in identifying questions at the Remember level. However, its performance declined for higher-order levels, with accuracy ranging between 25% and 59%. Yahya and Osman (2011) explored the effectiveness of machine learning techniques by employing TF-IDF features combined with Support Vector Machine (SVM) classifiers to categorize 190 exam questions across Bloom’s six cognitive categories. Haris and Omar (2012) employed a rule-based classifier to categorize 135 computer programming examination questions according to Bloom’s Taxonomy.

Building upon these methodologies, Mohammed and Omar (2020) introduced an enhanced classification model incorporating TFPOS-IDF, a variation of TF-IDF that considers part-of-speech information, and pretrained word2vec embeddings to capture semantic relationships. They evaluated their model using kNN, Logistic Regression, and SVM classifiers on datasets containing 141 and 600 questions. The SVM classifier exhibited superior performance, achieving weighted F1-scores of 83.7% and 89.7% on the respective datasets, highlighting the efficacy of integrating syntactic and semantic features in question classification.

Li et al. (2022) conducted a study to automate the classification of learning objectives according to Bloom’s Taxonomy. They compiled 21,380 learning objectives from 5,558 courses at an Australian university, manually labeled these objectives based on Bloom’s six cognitive levels, and applied five conventional machine learning algorithms—Naive Bayes, Logistic Regression, Support Vector Machine, Random Forest, and XGBoost—as well as a deep learning approach using the pretrained BERT language model. Their findings demonstrated that BERT-based classifiers outperformed others across all cognitive levels, achieving Cohen’s κ up to 0.93 and F1 scores up to 0.95. Additionally, SVM, Random Forest, and XGBoost models delivered performance comparable to BERT-based classifiers. The study also revealed that constructing separate binary classifiers for each cognitive level slightly outperformed a single multi-class, multi-label classifier, suggesting that individualized models for

each cognitive level may enhance classification accuracy.

Gani et al. (2023) focused on automating the classification of exam questions by evaluating various pretrained word embedding techniques, both contextual and non-contextual, across two datasets. Their study highlighted that while deep learning and contextual embeddings improved classification performance, their effectiveness was significantly influenced by dataset characteristics. Similarly, Al Faraby et al. (2024) assessed the capability of ChatGPT in classifying and generating questions. They found that in generating questions from reading sections, the differences with human-generated questions were not significant, indicating ChatGPT's potential for educational content creation.

2.2 Automatic Evaluation of Questions

Recent advancements in natural language processing have facilitated the automated evaluation of open-ended question complexity using Bloom's Taxonomy. Raz et al. (2024) employed a fine-tuned LLM to predict human ratings of question complexity, demonstrating a strong correlation ($r = 0.73$) between LLM-generated scores and human assessments, outperforming traditional baseline measures such as semantic distance and word count. Simone A Luchini and Beaty (2025) investigated the use of LLMs to assess the originality of narratives across multiple languages. They trained three distinct LLMs to predict human originality ratings of short stories written in 11 languages. The first model, trained exclusively on English narratives, achieved a robust correlation ($r = 0.81$) with human assessments. When this model was applied to multilingual stories translated into English, it maintained strong predictive performance ($r \geq 0.73$). Additionally, a multilingual model trained on narratives in their original languages reliably predicted human originality scores across all languages ($r \geq 0.72$). Hwang et al. (2023), explored an AI-driven approach to generating and evaluating multiple-choice questions in introductory chemistry and biology, focusing on alignment with Bloom's Taxonomy. They employed zero-shot prompting with GPT-3.5 to create questions, validated their cognitive levels using RoBERTa, and assessed question quality based on Item Writing Flaws Moore et al. (2023). The findings indicate that GPT-3.5 is capable of generating questions at various cognitive levels, particularly excelling at producing higher-order thinking questions at the

Evaluation level. However, discrepancies between AI-generated and human-assessed Bloom levels suggest the need for further refinement in question generation methodologies. Additionally, the study highlights an inverse correlation between Bloom's level and perceived question quality, indicating that while AI can generate complex questions, it may struggle with nuances in cognitive distinction and clarity at higher taxonomic levels.

3 Methodology

3.1 Dataset

This study is based on a dataset of transcribed interviews collected to evaluate the cognitive depth of questions and responses used during the "Empathy" phase of the Design Thinking process. In this study, the interviews focused on three distinct subject areas: Grandfather Game Application, Math Visualizer Software, and AI Regulation (for a description of each area see Appendix B). These topics were selected to ensure a variety of user perspectives and cognitive demands, ranging from personal storytelling to educational technology and policy discussions.

A total of 15 semi-structured interviews were conducted. Each interview consisted of both high-level and low-level open-ended questions. Not all questions were equally well-structured, as the goal was to intentionally support a range of cognitive levels in line with Bloom's Taxonomy, enabling analysis across varying depths of reasoning and understanding. The interviews were audio-recorded with participant consent, transcribed using Microsoft Teams, and manually reviewed for accuracy. Transcripts were anonymized and structured by role (interviewer/interviewee).

While the original transcripts included more entries, we removed manually segments that were not suitable for cognitive classification. This included ice-breaker exchanges (for example, "Hi, how are you today?", "Thanks for joining us!"), affirmations (for example, "yes", "okay"), and expressions of appreciation (for example, "thank you"), all of which could not be meaningfully assigned a Bloom's level. After this filtering process, the final dataset consisted of 726 entries, comprising 363 interview questions and their corresponding 363 responses. All questions and responses were manually classified by one of the authors familiar with Bloom's Taxonomy levels. Our analysis spans the three interview subjects: AI Regulation

(274 entries), Math Visualizer (244 entries), and Grandfather Game (208 entries).

3.2 Bloom-Level Classification Process

To classify each interview question and response according to Bloom’s Taxonomy, we employed a prompt-based strategy using two LLMs: LLaMA-3-8B-Instruct and GPT-4o-mini. We applied three prompting techniques including: zero-shot, few-shot, and CoT to guide the models’ responses.

In the zero-shot prompting approach, the model receives a direct instruction to classify the input into one of the six Bloom levels including: Remember, Understand, Apply, Analyze, Evaluate, or Create, without being given any prior examples. This method tests the model’s ability to rely on its internalized knowledge of Bloom’s Taxonomy and produces a fast baseline classification.

In few-shot prompting, we provide the model with one labeled example for each Bloom’s Taxonomy level before introducing the target input. These examples help calibrate the model’s understanding of the classification task.

Finally, we apply CoT prompting, which instructs the model to explain its reasoning before presenting a final classification. This method encourages step-by-step cognitive processing, making the model’s decision-making process transparent and auditable.

The purpose of this classifications is to evaluate their alignment with human judgment and to inform future efforts toward automating cognitive-level assessment in interviews (see Section 4.1 for the results).

3.3 Evaluation Framework

To assess the cognitive quality of interviews, we used human-annotated Bloom’s Taxonomy classifications for each question and response. Based on these annotations, we calculated three key evaluation metrics: Effectiveness Score (ES), Bloom Coverage Score (BCS), and Distribution Balance Score (BDS), developed by the authors to capture different dimensions of cognitive engagement. Together, these metrics represent the Cognitive Interview Quality Score (CIQS), a composite measure reflecting the cognitive richness and structural diversity of each interview.

In this study, CIQS and its components were derived from human classifications due to their higher reliability. The following sections review the components of the CIQS metric.

3.3.1 Effective Score (ES)

The Effectiveness Score measures how well each interview question succeeds in eliciting the intended level of cognitive engagement, as defined by Bloom’s Taxonomy. Rather than evaluating the question in isolation, this score is grounded in a comparison between the cognitive level of the question and the cognitive depth observed in the interviewee’s response. This approach aligns with the goals of the “Empathy” phase in Design Thinking, where the primary objective is not only to ask meaningful questions but to generate equally meaningful insights [Brown \(2009\)](#).

To calculate ES, first each question–response pair is evaluated by comparing the intended cognitive level of the question with the actual level of the response, and rated according to this criteria:

- Highly Effective (2 points): The response exceeds the intended cognitive level (for example, a question aimed at “Analysis” receives a “Creative” response).
- Effective (1 point): The response matches the intended cognitive level of the question.
- Needs Improvement (0 points): The response falls below the intended level, indicating limited cognitive engagement.

After assigning these numerical values to each pair, the ES for each interview is calculated as the average score across all pairs:

$$\text{Effectiveness Score} = \frac{\sum_{i=1}^n s_i}{n} \quad (1)$$

where $s_i \in \{0, 1, 2\}$ is the score assigned to the i -th question–response pair based on the mentioned criteria, and n is the total number of pairs in the interview. The resulting score ranges from 0 (all questions need improvement) to 2 (all questions are highly effective). This metric captures not only the cognitive intent behind the questions but also their real-world impact as demonstrated through participant responses.

3.3.2 Bloom Coverage Score (BCS)

The Bloom Coverage Score evaluates the extent to which an interview engages participants across the six levels of Bloom’s Taxonomy. A higher BCS indicates greater cognitive diversity, reflecting an intentional design that stimulates a broad range of thinking processes.

This diversity is particularly important in the context of Design Thinking, where complex problem-solving requires movement across multiple cognitive domains. Wu et al. (2021) propose a design thinking model explicitly structured around Bloom’s Taxonomy, arguing that design thinking can be taught and structured through cognitive processes, from basic understanding to advanced creative generation. They emphasize that aligning design tasks with Bloom’s full spectrum enables learners and participants to progress systematically from comprehension to innovation.

We define BCS as the number of cognitive levels covered in the interview divided by the total number of levels (6). The ideal BCS is 1.0, indicating that all six Bloom’s levels are present at least once. The metric focuses on whether each level appears, not how often, encouraging diverse cognitive coverage in interview design.

3.3.3 Distribution Balance Score (BDS)

While the BCS measures the number of Bloom’s cognitive levels represented in an interview, it does not reflect how evenly those levels are distributed. A cognitively rich interview is not only diverse in coverage but also balanced, ensuring that no single level dominates. To address this, we introduce BDS, which quantifies the uniformity of the cognitive distribution across Bloom’s levels.

Let p_i represent the proportion of questions classified into the i -th Bloom’s Taxonomy level, and let n be the total number of Bloom levels ($n = 6$). The BDS is defined as:

$$\text{BDS} = 1 - \frac{\sum_{i=1}^n \left(p_i - \frac{1}{n}\right)^2}{\frac{n-1}{n}} \quad (2)$$

This formula computes the squared deviation of the observed distribution $\{p_i\}$ from a uniform distribution $\frac{1}{n}$, and normalizes it by the maximum possible imbalance, which occurs when all items are concentrated in a single Bloom level. The squared term ensures that both over and underrepresentation contribute equally to the imbalance score, while penalizing larger deviations more. The BDS value ranges between 0 and 1.0. A BDS of 1.0 indicates a perfectly balanced distribution across all Bloom levels, reflecting equal representation. Conversely, a BDS of 0 signifies complete imbalance, with all items concentrated in a single Bloom level.

The formulation of the BDS is adapted from Pielou’s Evenness Index Pielou (1966), tradition-

ally used in ecology to assess distributional uniformity. We apply this concept to measure cognitive balance across Bloom’s levels. Unlike entropy-based alternatives, our variance-based approach offers greater simplicity and sensitivity to cognitive imbalances. This metric encourages interviews that span multiple cognitive levels in a well-distributed and cognitively meaningful way.

3.3.4 Cognitive Interview Quality Score (CIQS)

To provide a comprehensive assessment of interview quality from a cognitive perspective, we propose the Cognitive Interview Quality Score. This metric combines three core dimensions: practical effectiveness, cognitive coverage, and structural balance. CIQS is calculated using the following weighted formula:

$$\text{CIQS} = 0.5 \times \text{ES} + 0.3 \times \text{BCS} + 0.2 \times \text{BDS} \quad (3)$$

In this formula, Effectiveness is emphasized most heavily to reflect the importance of empirical success: questions must not only be well-designed but must also stimulate the intended cognitive engagement, as evidenced by actual responses Anderson and Krathwohl (2001). Bloom Coverage receives moderate emphasis for its role in encouraging diverse thinking pathways, while Distribution Balance contributes structural integrity without dominating the evaluation. The weighting scheme (0.5 for ES, 0.3 for BCS, and 0.2 for BDS) was determined to prioritize cognitive alignment in actual responses while still valuing breadth and balance. This design is informed by principles from educational assessment and cognitive taxonomy theory Anderson and Krathwohl (2001), though the metric itself is introduced as part of this work. The CIQS serves as a unified cognitive quality rating for each interview, enabling systematic comparison across topics or participant groups while supporting iterative improvement in interview design.

3.4 Human Evaluation of Interview Quality

To validate the CIQS framework, we conducted a human evaluation in which an expert (tenured Professor) in design thinking independently ranked the interviews across all three subjects. The expert ranked each interview based on its effectiveness in uncovering useful information about the user and their practices and needs. This qualitative judgment served as a benchmark to assess how well CIQS scores aligned with human-perceived

interview quality. Comparing the CIQS rankings with the expert’s rankings helps determine whether cognitively focused metrics reflect what a human evaluator considers a high-quality, informative interview.

4 Experiments & Results

4.1 Evaluating Human–LLM Cognitive Classification Agreement

One of the authors annotated all question-response pairs in our dataset for their Bloom’s Taxonomy level. To measure the agreement between LLM-assigned and human-assigned Bloom’s Taxonomy levels, we used Kendall’s Tau (τ), which is well-suited for ordinal data and provides a robust estimate of correlation, particularly with small sample sizes and tied ranks (Kendall, 1938). The results are presented in Table 1, indicate that the zero-shot GPT-4o-mini achieved the strongest alignment with human judgments in all domains: AI Regulation ($\tau = 0.58$), Math Visualizer ($\tau = 0.47$), and Grandfather Game ($\tau = 0.56$). Among LLaMA-3-8B-Instruct models, the few-shot prompting yielded the highest correlations overall, particularly in AI Regulation ($\tau = 0.33$). In contrast, zero-shot prompting under LLaMA showed very weak agreement across subjects.

These findings suggest that GPT-4o-mini, especially in zero-shot, is more reliable for capturing cognitive-level distinctions in interview data, while open-source LLaMA models show more limited alignment with expert assessments. Performance differences can be attributed to the models’ architectures and training methodologies. GPT-4o-mini (OpenAI’s distilled model) balances efficiency and advanced reasoning, excelling in nuanced tasks.¹ LLaMA-3-8B-Instruct, while optimized for dialogue and instruction-following, may require further fine-tuning to match the classification accuracy demonstrated by GPT-4o-mini in this study.²

To identify which Bloom’s levels posed the greatest challenges for LLMs, we generated separate confusion matrices comparing the aggregated predictions of LLaMA-3-8B-Instruct models and GPT-4o-mini models against human classifications across Bloom’s Taxonomy levels, as presented in Figures 1 and 2. The LLaMA ensemble, based on

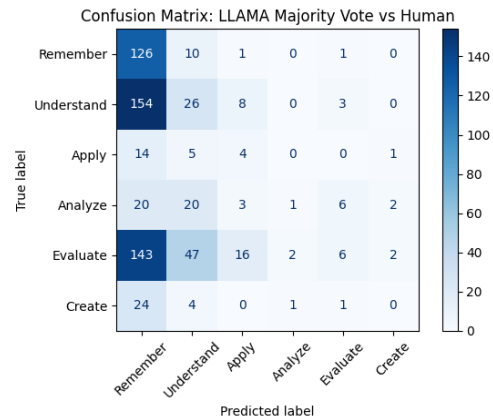


Figure 1: Confusion Matrix: LLaMA-3-8B-Instruct Majority Vote Vs Human Classification.

majority voting, exhibited a strong overprediction of the “Remember” category, leading to widespread misclassification of responses originally labeled as “Understand”, “Evaluate”, and “Create”. This pattern suggests a tendency to default to lower-order cognitive categories. In contrast, the GPT-4o-mini ensemble produced a more balanced distribution across predicted classes, with higher accuracy in identifying “Remember”, “Understand” and “Evaluate”, and notably less confusion between the levels.

These findings are further supported by the quantitative results reported in Tables 2 and 3. The LLaMA-3-8B-Instruct models showed limited alignment with human labels, with accuracy ranging from 23.9% to 29.1% and macro F1-scores below 0.19. Their highest macro precision and recall were 0.312 and 0.226, respectively, under the Chain-of-Thought setting. In contrast, all GPT-4o-mini variants outperformed LLaMA across metrics. The Zero-shot GPT model achieved 53.7% accuracy and a macro F1-score of 0.511, while Few-shot prompting reached a macro precision of 0.642. GPT models also showed stronger weighted F1-scores, indicating better overall balance across Bloom levels.

4.2 Evaluating Cognitive Dimensions of Interviews with CIQS

To evaluate and compare the cognitive quality of interviews across different topics, we applied our proposed scoring framework, the Cognitive Interview Quality Score, which combines three key dimensions: Effectiveness Score, Bloom Coverage Score, and Distribution Balance Score. As illustrated in Table 4, AI Regulation achieved the highest CIQS (0.88), supported by strong effectiveness (ES =

¹<https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence>

²<https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>

Model and Prompting Technique	AI Regulation	Math Visualizer	Grandfather Game
LLaMA-3-8B-Instruct Zero-shot	0.08	-0.01	0.01
LLaMA-3-8B-Instruct Few-shot	0.33	0.26	0.26
LLaMA-3-8B-Instruct Chain-of-Thought	0.18	0.09	0.33
GPT-4o-mini Zero-shot	0.58	0.47	0.56
GPT-4o-mini Few-shot	0.45	0.41	0.51
GPT-4o-mini Chain-of-Thought	0.52	0.41	0.47

Table 1: Kendall’s Tau (τ) correlation coefficients between model predictions and human annotations with highest scoring models in bold.

Model	Accuracy	Macro Precision	Macro Recall	Macro F1	Weighted F1
LLaMA-3-8B-Instruct Zero-shot	0.239	0.218	0.171	0.129	0.180
LLaMA-3-8B-Instruct Few-shot	0.285	0.222	0.225	0.155	0.205
LLaMA-3-8B-Instruct Chain-of-Thought	0.291	0.312	0.226	0.185	0.230

Table 2: Performance of LLaMA-3-8B-Instruct models across prompting techniques.

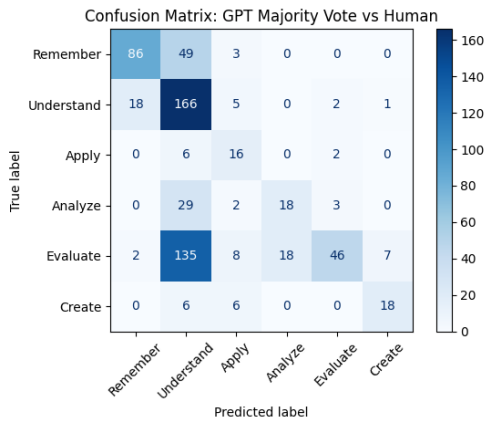


Figure 2: Confusion Matrix: GPT-4o-mini Majority Vote vs Human Classification.

1.01) and distribution balance (BDS = 0.80), despite slightly lower Bloom coverage (BCS = 0.71). This suggests that responses in AI-related interviews were well-aligned with the intended cognitive levels and well-distributed, though not all Bloom levels were equally represented. In contrast, Math Visualizer interviews exhibited the lowest CIQS (0.82), mainly due to a lower effectiveness score, suggesting that responses did not consistently reach the cognitive depth expected from the questions. Grandfather Game fell in the middle CIQS (0.84), showing relatively strong alignment but narrower cognitive coverage.

This automated scoring approach enables an objective comparison of interviews based on cognitive dimensions. However, cognitive depth is only one aspect of interview quality. As part of future work, we aim to explore additional metrics, such as emotional engagement, relevance to interview goals, procedural coverage, and question neutrality. These dimensions emerged from the feedback we

received during our interview sessions on different topics, where participants highlighted aspects that contributed to more meaningful and engaging conversations. These dimensions may offer a more complete view of interview quality beyond what Bloom’s taxonomy captures.

4.3 Evaluating the Alignment Between CIQS and Human Rankings

Figures 3-5 compare CIQS-based rankings with human expert judgments. Each CIQS score reflects a weighted combination of ES, BCS, and BDS.

In AI Regulation, the expert ranked Interview 3 as the most effective and Interview 1 as the least, while our CIQS-based scoring produced the opposite order and ranked Interview 3 as the most effective, highlighting a misalignment between cognitive structure (as captured by CIQS) and the expert’s judgment, which was based on how well each interview uncovered useful information about the user and their practices and needs. For Math Visualizer, Interview 5 ranked highest by CIQS due to perfect BCS and strong ES, while the expert preferred Interview 2 for its insightfulness. In Grandfather Game, both approaches aligned on Interview 1 as the best, though discrepancies appeared in the middle ranks. Notably, further analysis revealed that Bloom Coverage Score more closely aligned with human expert rankings than CIQS or other individual metrics. BCS showed moderate to strong correlations with expert judgments across all domains: ($\rho = 0.50$) in Math Visualizer, ($\rho = 0.90$) in Grandfather Game, and ($\rho = 0.71$) in AI Regulation. These results suggest that interviews with broader cognitive coverage were more likely to be perceived as informative and high-quality by experts, contradicting our initial hypothesis that

Model	Accuracy	Macro Precision	Macro Recall	Macro F1	Weighted F1
GPT-4o-mini Zero-shot	0.537	0.584	0.537	0.511	0.521
GPT-4o-mini Few-shot	0.491	0.642	0.481	0.477	0.453
GPT-4o-mini Chain-of-Thought	0.518	0.529	0.527	0.494	0.518

Table 3: Performance of GPT-4o-mini models across prompting techniques.

Metric	AI Regulation	Math Visualizer	Grandfather Game
Effectiveness Score	1.01	0.84	1.00
Bloom Coverage Score	0.71	0.80	0.64
Distribution Balance Score	0.80	0.82	0.75
Cognitive Interview Quality Score	0.88	0.82	0.84

Table 4: Cognitive evaluation scores across interview subjects with highest scores in bold.

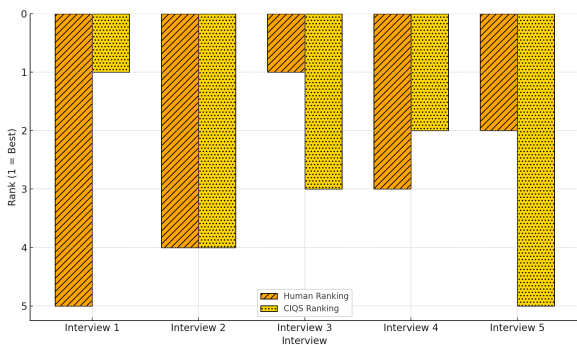


Figure 3: AI Regulation: CIQS vs Human Rankings.

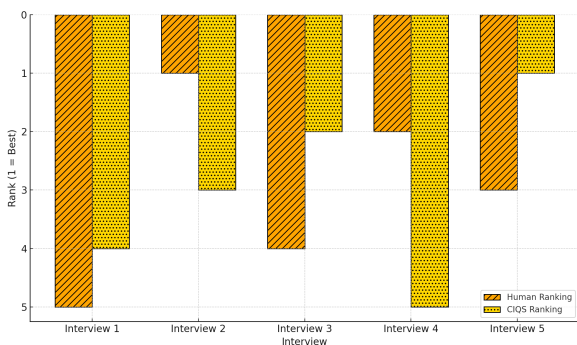


Figure 4: Math Visualizer: CIQS vs Human Rankings.

Effectiveness Score would play the most influential role in overall evaluation. To further investigate this we performed a linear regression to learn the coefficients for Equation 3 that best align with the human expert rankings. We found that BCS had the highest coefficient but that values varied across domains with ES and BDS less consistently in their impact. While more work is needed to determine which factors most correlate with human judgments, these preliminary results suggest that BCS is more impactful and that other attributes of the topic may be relevant in expert decisions (for full regression details, see Appendix C).

The results suggest that While CIQS captures

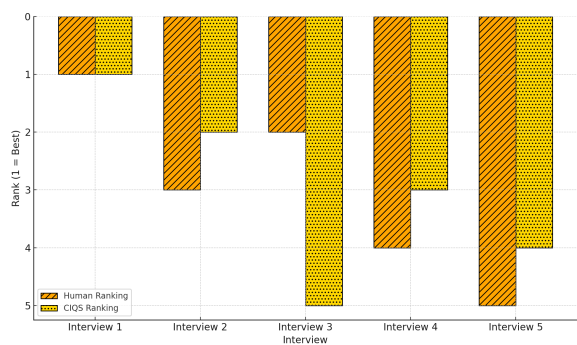


Figure 5: Grandfather Game: CIQS vs Human Rankings.

the cognitive structure of interviews, human evaluations often consider additional factors such as relevance, clarity, emotional engagement, and procedural detail. This highlights the value of combining cognitive metrics with qualitative insights for a more complete assessment of interview quality.

5 Discussion

RQ1: Our results suggest that interviews covering a broader range of Bloom’s cognitive levels (higher BCS) tend to be ranked more favorably by the human expert, indicating greater cognitive diversity. This supports the hypothesis that cognitive richness, particularly through varied questioning strategies, enhances the quality of interviews in the Design Thinking context. However, alignment with human expert rankings was not always consistent, implying that additional qualitative dimensions (for example, emotional engagement and the inclusion of procedural information) also influence perceived interview quality.

RQ2: The outputs from LLaMA and GPT-4o-mini demonstrated partial alignment with human annotations, showing that LLMs have the potential to support cognitive level classification. GPT-

4o-mini, in particular, showed stronger agreement with human labels across prompting strategies, especially in zero-shot settings. However, inconsistencies across domains and between models reveal that current LLMs are not yet fully reliable as standalone evaluators. While their performance is promising for future automation efforts, fine-tuning and prompt engineering may be necessary to achieve consistent, human-comparable accuracy.

RQ3: The CIQS rankings showed partial alignment with expert human evaluations, with higher consistency in the Grandfather Game domain ($\tau = 0.40$) and greater divergence in Math Visualizer ($\tau = -0.8$) domain. These differences suggest that while CIQS effectively captures the cognitive structure and balance of interviews, human experts often consider additional qualitative dimensions, such as emotional engagement, relevance to user needs, and the inclusion of procedural information, that are not directly encoded in cognitive metrics. As such, CIQS serves as a valuable and scalable starting point for evaluating interview quality, but it should complement qualitative assessments.

6 Conclusion & Future Work

This study introduced a cognitive evaluation framework for interview quality based on Bloom’s Taxonomy, applied within the context of Design Thinking. We proposed the CIQS, a composite metric incorporating effectiveness, coverage, and distribution of cognitive levels. Using human-annotations, we collected and evaluated 15 interviews across three domains to measure the cognitive diversity and structure of interview content. We compared CIQS rankings with expert judgments, finding that while they are partially aligned, BCS correlates more strongly with human rankings than CIQS or other individual metrics, suggesting that breadth is especially valued by experts. GPT-4o-mini, particularly in zero-shot, showed the highest agreement with human Bloom level annotations (up to $\tau = 0.58$), outperforming LLaMA-3-8B-Instruct.

These findings suggest that while CIQS effectively captures the cognitive structure of interviews, human evaluations often prioritize additional factors such as relevance to user needs, clarity, emotional engagement, and procedural depth. This highlights the importance of complementing cognitive metrics with broader qualitative dimensions for a more comprehensive assessment of interview quality. In future work, we plan to refine CIQS by

exploring alternative weighting, incorporating additional qualitative indicators, and fine-tuning LLMs for more accurate, autonomous classification of interview content based on Bloom’s Taxonomy. To support continued research, we will release our corpus of 726 question–response pairs spanning three domains to support future work.

Limitations

The main challenge of this and any study of Design Thinking effectiveness is the maxim “savour surprises”, by which design thinkers mean that the most important information is usually the information which was not anticipated and not planned for. This is because this information is the most likely to invalidate a design made without in-depth user interviews, or to lead to a new product category which was not previously contemplated [Furr and Dyer \(2014\)](#). At this stage, we are not trying to identify such surprises, but ultimately, a research program aiming to improve design education will have to address it.

A more immediate limitation of this study is the use of a single human expert. Experts in teaching and evaluating design thinking are uncommon and in demand in academia and industry. To increase the number of evaluators, it will be necessary to streamline the process so that it is less time-consuming.

Another limitation of this study is that even the human evaluator is not evaluating what we ultimately care about: the acceleration of the innovation process through better design interviews. We do not know whether interviews ranked highly by human experts actually lead to higher rates of innovation. Once automated metrics are found with higher levels of agreement with human experts, validation studies including the full development cycle from initial interviews to product validation will be necessary.

Role-playing can be challenging depending on the task. For our AI interviews, we noticed a lack of procedural information and emotion, where we expected more of both. We think it is not trivial for most people to role-play older individuals or versions of themselves. We suggest future work in this direction borrows from more established fields to set up experiments involving perspective-taking, e.g. work on empathy [Batson et al. \(2002\)](#).

Finally, since our intermediate goal is to produce tools useful for teaching design skills, it is disap-

pointing that the proprietary LLM greatly outperformed the open-source LLM. Many school boards and higher education institutions will be reluctant to submit their students' data to proprietary LLMs which they cannot control. In our study, we used role-playing by sophisticated professionals, graduate students and upper-year undergraduate students to produce a data set for training and evaluation. In teaching scenarios, it would be much harder to insure that personal information would not lead into the interviews. Moreover, when you initially describe the data set, you need to use similar language to say that this data is designed to not include personal information. Finally, a data set generated using role-playing may be fundamentally different from real design interviews in a way which effects the validity of the metrics.

While Bloom's Taxonomy provides a useful scaffold for assessing cognitive engagement, it has known limitations. The taxonomy does not explicitly model underlying mental processes such as perception, memory, and intuition, and some categories may overlap in practice—for example, extrapolation under "Understand" often resembles "Apply." Furthermore, the hierarchy between "Apply", "Analyze", and "Create" has been critiqued as insufficiently nuanced. Future extensions could explore integrating more adaptive taxonomies that better capture the fluid and context-dependent nature of reasoning in design interviews (Madaus et al., 1973).

References

- Said Al Faraby, Ade Romadhony, and Adiwijaya. 2024. Analysis of LLMs for educational question classification and generation. *Computers and Education: Artificial Intelligence*, 7:100298.
- David R. Krathwohl and. 2002. A Revision of Bloom's Taxonomy: An Overview. *Theory Into Practice*, 41(4):212–218.
- Lorin W Anderson and David R Krathwohl. 2001. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition*. Addison Wesley Longman, Inc.
- C Daniel Batson, Nadia Ahmad, David A Lishner, J Tsang, CR Snyder, and SJ Lopez. 2002. Empathy and altruism. *The Oxford handbook of hypo-egoic phenomena*, pages 161–174.
- Benjamin S. Bloom. 1956. Taxonomy of educational objectives: The classification of educational goals. *Handbook; Cognitive domain*, 1.
- Tim Brown. 2009. *Change by Design: How Design Thinking Creates New Alternatives for Business and Society*. Harper Business, New York, NY.
- 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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Wen-Chih Chang and Ming-Shun Chung. 2009. Automatic applying Bloom's Taxonomy to classify and analysis the cognition level of English question items. In *2009 Joint Conferences on Pervasive Computing (JCPC)*, pages 727–734. IEEE.
- Nathan R Furr and Jeff Dyer. 2014. *The innovator's method: bringing the lean start-up into your organization*. Harvard Business Press.
- Mohammed Osman Gani, Ramesh Kumar Ayyasamy, Anbuselvan Sangodiah, and Yong Tien Fui. 2023. Bloom's Taxonomy-based exam question classification: The outcome of CNN and optimal pre-trained word embedding technique. *Education and Information Technologies*, 28(12):15893–15914.
- Syahidah Sufi Haris and Nazlia Omar. 2012. A rule-based approach in Bloom's Taxonomy question classification through natural language processing. *2012 7th International Conference on Computing and Convergence Technology (ICCT)*, pages 410–414.
- Kevin Hwang, Sai Challagundla, Maryam Alomair, Lujie Karen Chen, and Fow-Sen Choa. 2023. Towards AI-assisted multiple choice question generation and quality evaluation at scale: Aligning with Bloom's Taxonomy. In *Workshop on Generative AI for Education*.
- M. G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika*, 30(1-2):81–93.
- Yuheng Li, Mladen Rakovic, Boon Xin Poh, Dragan Gasevic, and Guanliang Chen. 2022. Automatic Classification of Learning Objectives Based on Bloom's Taxonomy. In *Proceedings of the 15th International Conference on Educational Data Mining*, pages 530–537, Durham, United Kingdom. International Educational Data Mining Society.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.
- George F. Madaus, Elinor M. Woods, and Ronald L. Nuttall. 1973. A causal model analysis of bloom's taxonomy. *American Educational Research Journal*, 10(4):253–262.

Manal Mohammed and Nazlia Omar. 2018. [Question classification based on Bloom's Taxonomy using enhanced TF-IDF](#). *International Journal on Advanced Science, Engineering and Information Technology*, 8(4-2):1679.

Manal Mohammed and Nazlia Omar. 2020. [Question classification based on Bloom's Taxonomy cognitive domain using modified TF-IDF and word2vec](#). *PloS one*, 15(3):e0230442.

Steven Moore, Huy A Nguyen, Tianying Chen, and John Stamper. 2023. [Assessing the quality of multiple-choice questions using GPT-4 and rule-based methods](#). In *European conference on technology enhanced learning*, pages 229–245. Springer.

IDEO Org. 2015. *The field guide to human centered design*. Ideo Org.

E.C. Pielou. 1966. [The measurement of diversity in different types of biological collections](#). *Journal of Theoretical Biology*, 13:131–144.

Tuval Raz, Simone Luchini, Roger Beaty, and Yoed Kenett. 2024. [Automated Scoring of Open-Ended Question Complexity: A Large Language Model Approach](#). *Proceedings of the Annual Meeting of the Cognitive Science Society*, 46.

John D. Patterson Dan Richard Johnson Matthijs Baas Baptiste Barbot Iana P. Bashmakova Mathias Benedek Qunlin Chen Giovanni Emanuele Corazza Boris Forthmann Benjamin Goecke Sameh Said Ibrahim Maciej Karwowski Yoed Kenett Izabela Lebuda Todd Lubart Kirill G. Miroshnik Felix Kingsley Obialo Marcela Ovando-Tellez Ricardo Primi Rogelio Puente Diaz Claire Stevenson Emmanuelle Volle Aleksandra Zielińska Janet van Hell Yin Wenpeng Simone A Luchini, Moosa Ibraheem Muhammad and Roger Beaty. 2025. [Automated assessment of creativity in multilingual narratives](#). *Psychology of Aesthetics, Creativity, and the Arts*.

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.

Fan Wu, Yang Cheng Lin, and Peng Lu. 2021. [A new design thinking model based on Bloom's Taxonomy](#). *Learn X Design 2021: Engaging with challenges in design education*.

Anwar Ali Yahya and Addin Osman. 2011. [Automatic classification of questions into Bloom's cognitive levels using support vector machines](#). *Computer Science, Education*.

7 Appendix

A Prompts Used for LLM Classification

This appendix provides the exact prompts used to classify interview questions and responses into

cognitive levels according to Revised Bloom's Taxonomy, using different prompting techniques.

Zero-Shot Prompt

You are a cognitive science expert that categorizes text into one of the Revised Bloom's Taxonomy levels.

You must respond with only one word: one of the following levels: Remember, Understand, Apply, Analyze, Evaluate, or Create.

Do not provide any explanation, reasoning, or additional text. Only return the level name in the following format.

Classification: <One of the six Bloom's levels>

Few-Shot Prompt

You are a cognitive science expert trained in Revised Bloom's Taxonomy.

Classify the following text according to Revised Bloom's Taxonomy levels: Remember, Understand, Apply, Analyze, Evaluate, or Create.

Examples:

Text: "List the main components of design thinking."

Classification: Remember

Text: "Explain the theory of cognitive load."

Classification: Understand

Text: "How would you apply Pythagoras' theorem to calculate the height of a building?"

Classification: Apply

Text: "Identify patterns in customer behavior based on the provided dataset."

Classification: Analyze

Text: "Evaluate the effectiveness of renewable energy sources compared to fossil fuels."

Classification: Evaluate

Text: "Design a new marketing strategy for launching a product."

Classification: Create

Do not provide any explanation, reasoning, or additional text. Only return the level name in the following format.

Classification:<One of the six Bloom's levels>

Chain-of-Thought Prompt

You are a cognitive science expert in Revised Bloom's Taxonomy.

Your task is to classify a given text into one of the Revised Bloom's Taxonomy cognitive levels:

Remember, Understand, Apply, Analyze, Evaluate, or Create.

Text: {input-text}

First, explain your reasoning step by step based on what the text requires cognitively.

Then, based on your explanation, select the most appropriate Bloom's level from (Remember, Understand, Apply, Analyze, Evaluate, Create) using the following format:

Classification: <One of the six Bloom's levels>

Note: For all classifications, the following model parameters were used:

Temperature = 0.0, Max tokens = 500.

B Description of Interview Topics

This study includes interviews conducted on three different design topics, each selected to represent different cognitive and contextual demands. The topics were used to simulate early-stage Design Thinking sessions and assess the cognitive quality of interview interactions. The participants in this study included a mix of students or recent graduates and university professors, some of whom had prior experience with Design Thinking. To maintain anonymity, they were instructed to avoid disclosing any real personal information. Depending on the interview topic, participants were asked to adopt specific roles. In the Math Visualizer interviews, they were asked to act as university students; in the Grandfather Game interviews, they assumed the perspective of older adults; and in the AI Regulation interviews, they portrayed individuals using AI platforms in organizations such as schools or businesses. Interviewers were instructed to engage naturally while focusing on uncovering user needs and generating meaningful insights.

B.1 AI Regulation

This topic explores public perceptions, concerns, and expectations surrounding the regulation of artificial intelligence. The interviewees were asked about their understanding of AI technologies, trust in regulatory frameworks, and suggestions for ethical oversight. The domain encourages abstract reasoning and evaluative thinking about policy and technology.

B.2 Math Visualizer

This topic focuses on the use of visualization tools in learning mathematics. Participants discussed their personal experiences with visual learning, the challenges they face in understanding mathematical concepts, and ideas for improving visual interfaces.

B.3 Grandfather Game

This topic centers on designing a game that would appeal to older adults. Participants were asked to reflect on their childhood memories, personal interests, and previous gaming experiences to inform the creation of engaging and age-appropriate game concepts.

C Linear Regression for CIQS

We can predict the human rankings of design thinking interviews with the CIQS score by learning coefficients for Equation 3. We predict the coefficients with intercept for each conversation topic. The equation for the Grandfather Game topic is shown in Equation 4 and yields $R^2 = 0.51$ which matches human ranking. Similarly, for AI-Regulation, we get $R^2 = 0.99$ for Equation 5. Lastly, for the Math Visualiser we get Equation 6 with $R^2 = 0.58$. Here we are predicting the rank (lower is better) with the same terms, which is different than ranking by the maximum score as we did in the main part of the paper, however, it serves the same function and supports our claims that BCS is the most important term and the impact of factors appears to vary with the domain.

$$\begin{aligned} \text{CIQS} = & -11.0 \times \text{ES} - 25.0 \times \text{BCS} \\ & + 9.2 \times \text{BDS} + 22.9 \end{aligned} \quad (4)$$

$$\begin{aligned} \text{CIQS} = & -13.6 \times \text{ES} - 3.6 \times \text{BCS} \\ & + 53.5 \times \text{BDS} - 23.1 \end{aligned} \quad (5)$$

$$\begin{aligned} \text{CIQS} = & 21.0 \times \text{ES} - 18.8 \times \text{BCS} \\ & - 13.0 \times \text{BDS} + 11.1 \end{aligned} \quad (6)$$

Estimation of Text Difficulty in the Context of Language Learning

Anisia Katinskaia,^{†‡} Anh-Duc Vu,^{†‡} Jue Hou,^{†‡} Ulla Vanhatalo,[◇]
Yiheng Wu,[‡] Roman Yangarber[‡]

[†]Department of Computer Science, [‡]Department of Digital Humanities

[◇]Department of Finnish, Finno-Ugrian and Scandinavian studies

University of Helsinki, Finland

first.last@helsinki.fi

Abstract

Easy language and text simplification are currently topical research questions, with important applications in many contexts, and with various approaches under active investigation, including prompt-based methods. The estimation of the level of difficulty of a text becomes crucial when the estimator is employed inside a simplification workflow as a quality-control mechanism. It can act as a *critic* in frameworks where it can guide other models, which are responsible for generating text at a specified level of difficulty, as determined by the user's needs. We present our work in the context of simplified Finnish. We discuss problems in collecting corpora for training models for estimation of text difficulty, and our experiments with estimation models. The results of the experiments are promising: the models appear usable both for assessment and for deployment as a component in a larger simplification framework.

1 Introduction

In the US¹ and in the European Union,² legal pressures are emerging with laws that require government-affiliated agencies, as well as private-sector organizations in certain situations, to use clear communication that members of the public can understand. Workflows that involve easy language are already in official use at various levels of functioning in the public and private sectors in 20 countries in the EU. Easy language also plays a key role in second-language (L2) education, in particular—simplification of text to a level appropriate for a given learner is a key component of *personalization* in teaching. Simplification itself is a widely researched area in NLP.

In this paper, we take the position that methods for evaluating and *assessing* the difficulty level of a piece of text are *prerequisite* to methods for

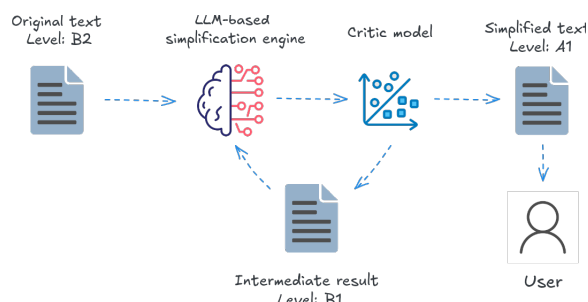


Figure 1: Text simplification using GPT-4o guided by level-aware feedback from a difficulty classifier as critic.

simplification—since in the absence of effective evaluation, simplification methods cannot be effectively validated or falsified.

The task of assessing the level of difficulty of the text can be framed as classification or (more appropriately) as regression—labeling a piece of text with a difficulty level, such as, e.g., a CEFR level.³ We will refer to models performing this task as *difficulty models*. These models can serve various purposes in language learning, such as estimating the difficulty level of texts that learners encounter. In this work, we use difficulty models to guide and evaluate text simplification pipelines performed by a large language model (LLM), specifically GPT-4o from OpenAI (Hurst et al., 2024).

Our simplification pipeline (Figure 1) employs a difficulty model that serves as a *critic*: it evaluates the difficulty level of the LLM output. If the resulting text exceeds the target level, feedback will be sent to the LLM to try again. The feedback includes the resulting text and its estimated level. The pipeline runs several iterations; if the resulting text remains harder than the target level after N iterations, the process terminates, and an error message is returned to the user.

We train two BERT-based difficulty models:

¹PlanLanguage.gov

²European Accessibility Act

³CEFR: Common European Framework of Reference for Languages.

one *regression* model, which predicts continuous scores that are later mapped to CEFR levels, and one *ordinal classification* model, which directly predicts the CEFR level of the input text. Our results show that both models improve the performance of the simplification pipeline over a baseline that runs without any critical guidance. The ordinal classification model proves to be a more effective critic for the LLM. Our hypothesis is that it aligns better with the difficulty assessment task because of its ordinal (ranking) nature.

The paper is organized as follows: Section 2 presents an overview of related work; Section 3 discusses the data we use to train the assessment models; Section 4 presents the experimental setup for the difficulty assessment; Section 5 presents the experiments with controlling the behavior of the LLM via a critic that assesses difficulty; Section 6 discusses the results and concludes the paper.

2 Related Work

Assessment of text difficulty, often referred to as readability assessment, has a long history in both education and in NLP. Traditional readability formulas, such as the Flesch-Kincaid Grade Level and Flesch Reading Ease, and the Lexile framework, based on item response theory (IRT), provide simple numeric scores for text difficulty (Kincaid et al., 1975; Stenner, 1996). These methods are easy to apply, but they rely on surface-level features and do not directly account for deeper lexical or syntactic complexity.

Early NLP readability systems used supervised models with hand-crafted linguistic features, including frequency word lists, depth of parse trees, grammatical constructions, and discourse structures. Collins-Thompson and Callan (2004) introduced a language modeling approach to predict reading difficulty for a tutoring system. Vajjala and Meurers (2012) incorporated features from Second Language Acquisition research to better serve language learners. For Russian, Laposhina et al. (2018) introduced a feature-based readability tool available online and widely used by L2 teachers.

Azpiazu and Pera (2019) present a multilingual readability model using a hierarchical attention network that learns to attend to difficult parts of a text and can implicitly learn factors like semantic difficulty or subtle syntactic cues. These models can be trained on proficiency-labeled data (e.g., with CEFR levels) to detect nuances of text difficulty

specific to L2 readers (e.g., idiomatic language). Recent work has shown that a fine-tuned BERT can outperform strong feature-based baselines by a significant margin in classifying texts by grade level or proficiency level (Martinc et al., 2021). Sharoff (2022) investigated compared the performance of Transformer-based models for predicting text difficulty vs. assessment using linguistic features, such as frequency of conjunctions, discourse particles, etc., for English and Russian.

Early pipeline approaches used readability classifiers to decide when to simplify: for example, Gasperin et al. (2009) trained a model to identify sentences that need simplification based on linguistic complexity features. Aluísio et al. (2010) developed readability assessment tools to support simplifying texts for low-literacy readers. Readability metrics have also served as simplification objectives in rule-based systems—Woodsend and Lapata (2011) incorporate a Flesch-Kincaid grade formula into an optimization-based simplifier.

Readability predictors have been used as feedback in generation loops—Alkaldi and Inkpen (2023) use a readability classifier in a reinforcement learning framework to iteratively simplify a text until it reaches the desired difficulty. More recently, large-scale neural systems have combined reading level prediction with controllable generation techniques (Agrawal and Carpuat, 2023).

3 Data

First, we describe the data used for training and evaluating the difficulty models and for the simplification pipeline. A major challenge is the scarcity of annotated data in Finnish for text simplification and difficulty prediction. To address this, we use a combination of Finnish texts annotated with difficulty levels (“native” data), and Russian texts annotated with difficulty levels and then translated into Finnish using machine-translation models.

3.1 Native Data

We use two collections of native Finnish data. The first consists of 1113 documents manually annotated by teachers of Finnish as a second language (L2), see “Manual” in Table 1. These are primarily informative and literary texts: the former covering topics such as human rights, social benefits, etc.; the latter feature classic Finnish literature and fragments of the Bible. The “Score” column in Table 1 shows the numerical values we assign to CEFR

Source	Level	Score	# Docs	# Words	# Sent.
SM	easy	1.5	153	294	9.3
YLE-selko	medium	3.5	766	249	8.7
HS	hard	5.5	715	598	13.7
YLE	hard	5.5	703	480	14.5
Manual	A2	2.0	363	237	10.9
	B1	3.0	229	204	11.0
	B2	4.0	154	221	11.8
	C1	5.0	192	272	17.5
	C2	6.0	175	189	19.9

Table 1: Native Finnish data.

levels, which are later used in regression models.

The second collection contains 2337 texts from *Suomen Mestari* (SM), a Finnish textbook, and *YLE selkosuomeksi* news,⁴ as well as news articles from the major newspapers *YLE* and *Helsingin Sanomat* (HS). These texts were not manually annotated. Instead, we make a coarse assumption based on the source: all texts from SM are labeled as easy, texts from YLE-selko as medium, and texts from YLE and HS as hard. We then suppose these difficulty levels roughly correspond to CEFR levels A1-A2, B1-B2, and C1-C2, respectively. Although this source-based annotation is a simplification—individual texts may vary in difficulty—it provides a practical heuristic in the context of limited human resources for annotating data.

3.2 Translated Data

Having some amount of Russian data annotated for difficulty, we translate it into Finnish to extend the size of the training set.

We use two sources of annotated Russian texts: 1. the *RuFoLa* corpus (Laposhina, 2020), which contains texts from coursebooks designed for learners of Russian as a foreign language; 2. the *RuAdapt* corpus (Dmitrieva and Tiedemann, 2021), a *parallel* Russian–Simple Russian dataset of texts adapted for learners of Russian as a foreign language. For our study, we use only the literary (*Zlatoust*) and encyclopedic sub-corpora, see Table 2. The “Score” column again shows the mapping between CEFR levels and numeric labels used later for a BERT-based regression model.

We filter out texts shorter than 10 words, as such a short context can negatively affect translation quality. We translated the Russian texts into Finnish using a model from OpusMT.⁵ We should

⁴News in Simple Finnish: yle.fi/selkouutiset

⁵The Tatoeba model for Slavic-Finnish.

Source	Level	Score	# Docs	# Words	# Sent.
RuFoLa	A1	1.0	301	136	8.8
Encyclop.	A1-A2	1.5	282	31	12.3
RuFoLa	A2	2.0	466	183	10.5
Zlatoust	A2-B1	2.5	96	50	8.2
RuFoLa	B1	3.0	3300	91	12.2
Zlatoust	B1-B2	3.5	1677	54	15.8
Zlatoust	B2	4.0	834	228	12.8
RuFoLa	C1	5.0	485	363	14.9
RuFoLa	C2	6.0	29	385	16.5

Table 2: Annotated documents in Russian.

Split	# Documents	Source
Training	6248	MT
	2221	Native
Validation	1222	MT
	364	Native
Test	865	Native

Table 3: Data splits.

note that machine translation does not guarantee that a text in Russian will remain at the same difficulty level after translation into Finnish. This problem merits a dedicated research experiment. The entire dataset was split into 3 sets: training, validation, and test, see Table 3. The test set—860 texts—contains only native documents, and most documents are manually annotated.

4 Experiments

To establish an interpretable baseline for document-level difficulty prediction, we first train a feature-based regression model. This allows us to evaluate how well linguistic features alone can capture text difficulty, and later compare its performance to that of less interpretable deep-learning approaches.

4.1 Feature-based Regression

In this experiment, we use only native Finnish texts to train a Ridge regression model that predicts the difficulty level of a document. The target labels are mapped to the following numeric values: 0.0 (easy), 1.0 (medium), and 2.0 (hard). Manually annotated documents are mapped to the same numeric values: A1-A2 to 0.0, B1-B2 to 1.0, and C1-C2 to 2.0. We use these numeric values instead of the scores presented in Table 1 for simplicity.

We use 179 features to capture linguistic characteristics of the texts. These include normalized averages of count of POS tags, depth of parse

tree, sentence length, word distribution across ten frequency bins, and the proportion of out-of-vocabulary (OOV) words.⁶ The features also include the counts of over 160 linguistic constructs, covering grammatical features—e.g., tense, case, number, etc.—and syntactic patterns—e.g., necessity constructions, government structures, etc. The extraction of constructs from text is performed using the text processing pipeline in the Revita language learning system (Katinskaia et al., 2018, 2017; Hou et al., 2019); see examples of linguistic constructs and how they are extracted from text in (Katinskaia et al., 2023). Details of the features appear in Appendix D.

We evaluate three variants of the baseline model:

- (A) using all 179 features,
- (B) using a bootstrap selection of 104 features,
- (C) performing feature selection by training a Lasso regression model.

More details on the models are presented in Appendix A. As all models exhibited comparable performance, we adopt model (B) as the baseline in subsequent analyses due to its smallest feature set.

4.1.1 Results

Evaluation was performed using only native Finnish texts. The baseline model (B) achieved a mean absolute error (MAE) of 0.27 and a root mean squared error (RMSE) of 0.35. Figure 2 shows the distribution of the predicted scores in the three coarse levels of difficulty. The plot shows that easy texts tend to get scores higher than 0.0. This could be explained by the fact that we have much fewer easy texts in the native corpus, as well as by the assumption that all SM texts should be labeled easy, while in fact some of these texts are of intermediate difficulty. Nevertheless, the results provide a strong baseline for comparison with more complex models used in subsequent experiments, which offer less interpretability.

4.2 BERT-based Regression

We extend the BERT model for regression-based difficulty prediction, integrating custom loss weighting to handle class imbalances in the training data. The model is based on BERT, whose output layer is replaced with: (a) a pre-classification layer that projects BERT’s pooled output into a lower-dimensional space, has ReLU activation and

⁶Based on a large Finnish corpus, we build a list of words sorted by frequency and grouped into frequency bins.

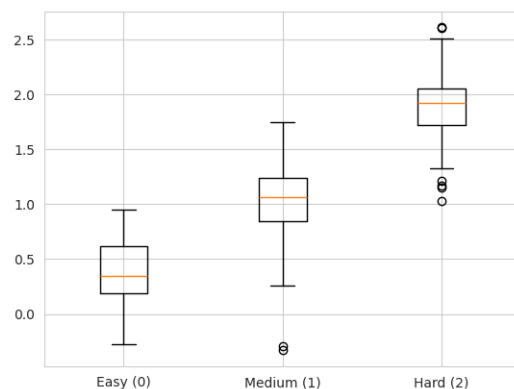


Figure 2: Feature-based regression baseline model (C). Predicted regression scores across difficulty levels: Easy (0), Medium (1), and Hard (2).

Dropout, and (b) a final feedforward regression head that predicts a continuous difficulty score.

The model is trained using weighted mean squared error (MSE) loss. To prevent the model from being biased toward the most frequent difficulty levels in the training data, sample weights are computed inversely proportional to the frequency of each difficulty level. These weights are then normalized to ensure they sum to 1.

The model was trained on all training data presented in Table 3, using the Adam optimizer, separate optimization parameters for the BERT parameters and the linear layers, weight decay = 0.01, cosine scheduler for the learning rate, and early stopping.

4.2.1 Results

The evaluation was again performed on the test set containing native Finnish texts. The BERT-based regression model achieved MAE 0.13 and RMSE 0.29. Figure 3 shows the distribution of the predicted scores at all CEFR levels in the test set. The number of documents per level is shown in the “Support” column of Table 4. As we can see from the plot, for some of the difficulty levels (particularly, for level A1-A2), predicted scores tend to be higher than the true labels, indicating some bias toward overestimation.

To assess the classification performance, we map real-valued predictions to the nearest CEFR level. The resulting confusion matrix is in Figure 4. Class-wise precision, recall, and F1-scores are in Table 4. Overall, the model performs well across most CEFR levels. The lowest F1-score is observed for the A1-A2 level, which also has the smallest number of examples in the test set.

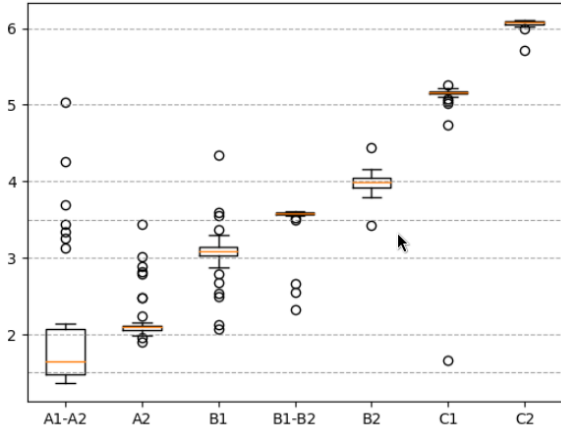


Figure 3: Predicted regression scores across difficulty levels using BERT-based regression model.

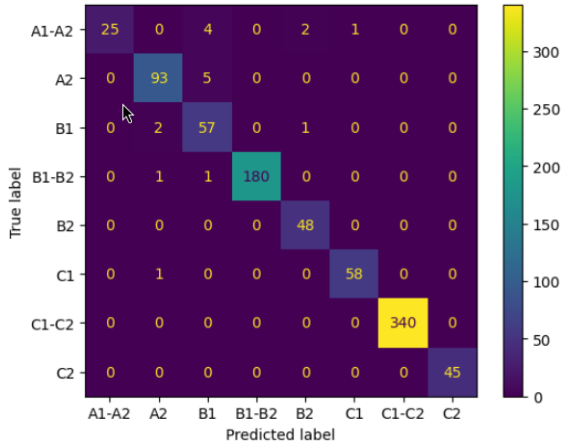


Figure 4: Confusion matrix after mapping difficulty scores to CEFR levels.

Only 1.3% of the test documents were assigned a predicted level that differs from the true level by *more than one* CEFR level. We consider deviations within one level to be acceptable, given the inherent difficulty and subjectivity of the task.

We examine the agreement between the feature-based regression model and the BERT-based regression on the test set, and the agreement of both models with the true labels. The predictions of the two models show a strong correlation: Spearman’s rank correlation is 0.76, and Pearson’s correlation is 0.83; Quadratic Weighted Kappa (QWK) is 0.84.⁷ This suggests a high degree of both rank-order and linear agreement between the models, despite being trained on different datasets and using different features.

The BERT-based model achieves near-perfect

⁷Predictions from the feature-based model were linearly rescaled to match the 1–6 scale of the BERT-based regression.

Level	Precision	Recall	F1-score	Support
A1-A2	1.00	0.78	0.88	32
A2	0.96	0.95	0.95	98
B1	0.85	0.95	0.90	60
B1-B2	1.00	0.98	0.99	183
B2	0.94	1.00	0.97	48
C1	0.98	0.98	0.98	59
C1-C2	1.00	1.00	1.00	340
C2	1.00	1.00	1.00	45

Table 4: Performance on the test set after mapping difficulty scores to CEFR levels.

Model	Pearson	Spearman	QWK
BERT vs. True	0.98	0.95	0.98
Feature vs. True	0.84	0.83	0.81

Table 5: Correlation and agreement of feature-based baseline model and BERT-based regression model with true labels.

agreement with the true difficulty labels (see Table 5), where the gain in QWK suggests that BERT is particularly better at matching difficulty levels. In contrast, the feature-based model demonstrates good but notably lower performance (0.98 vs. 0.81). Both models are available for testing.⁸

4.3 BERT-based Ordinal Classification

This model extends BERT for rank-consistent ordinal regression (Cao et al., 2020), a task in which labels have a meaningful order but unknown interval distances. Unlike standard classification, ordinal regression models the probability of a response exceeding certain thresholds, making it particularly useful for difficulty assessment.

The model predicts $\mathbb{P}(Y > k)$ for each threshold k using a modified BERT architecture, where a linear classifier estimates the probability that the input exceeds a set of ordinal thresholds. In particular, given an input sequence, we pass the pooled output of BERT through a dropout layer and a linear classification head of size (hidden_dim $\rightarrow K - 1$), where K is the number of CEFR levels.

For K ordinal labels, the model outputs $K - 1$ logits for each threshold. Each logit represents the probability:

$$\mathbb{P}(Y > k | X)$$

for each difficulty threshold k , where X represents the BERT-generated input representation.

⁸revita.helsinki.fi/selkomitta

Since ordinal regression differs from standard classification, we use a binary cross-entropy (BCE) loss adapted for ordinal constraints:

- **Ordinal Target Construction:** For a batch of size N and K classes, we construct a binary target matrix $\mathbf{T} \in \{0, 1\}^{N \times (K-1)}$, where each element $T_{i,k} = \mathbb{I}[y_i > k]$ indicates whether the true label exceeds threshold k .
- **Weighting Mechanism:** A weight matrix $\mathbf{W} \in \mathbb{R}^{N \times (K-1)}$ assigns higher penalties to more severe misclassifications. This can be scaled by a global hyperparameter α to control the influence of the weighting.

The weighted ordinal loss function is defined as:

$$L = \frac{1}{N} \sum_{i=1}^N s_i \sum_{j=1}^{K-1} w_{i,j} \cdot \text{BCE}(\sigma(z_{i,j}), t_{i,j})$$

where:

- N is the batch size.
- $z_{i,j}$ is the model logit for level j .
- $\sigma(z)$ is the sigmoid function.
- $t_{i,j}$ is the binary target: 1 if the true label exceeds threshold j , 0 otherwise.
- $w_{i,j}$ is a weight penalty based on label distance
- s_i is an additional sample-level weight to address class imbalance.

The weights w_{ij} are given by:

$$w_{ij} = 1 + \alpha \cdot |y_i - j|, \quad \alpha > 0$$

where y_i is the true ordinal label. This weighting penalizes predictions that are farther from the correct class more heavily. In our experiments, we set $\alpha = 0.5$.

By modeling thresholds rather than treating classes as independent, the loss preserves ordinal relations. Furthermore, the model learns a probability distribution over ranks, capturing uncertainty rather than committing to hard class decisions.

To obtain the predicted ordinal class, we apply a sigmoid activation to the model’s output logits, yielding threshold probabilities $\mathbb{P}(Y > k)$ for each $k = 1, \dots, K - 1$. The predicted class \hat{y} is then calculated by counting how many of these probabilities exceed the threshold of 0.5:

$$\hat{y} = \sum_{k=1}^{K-1} \mathbb{I}[\mathbb{P}(Y > k) > 0.5]$$

Accuracy	0.76	RMSE	0.57
MAE	0.28	ρ	0.89
QWK	0.87	τ	0.83

Table 6: Results of ordinal classification.

Here, $\mathbb{I}[\cdot]$ denotes the indicator function, which returns 1 if the condition is true and 0 otherwise.

Intuitively, this approach treats the predicted class as the number of ordinal thresholds that the input is likely to exceed with confidence greater than 0.5—higher classes correspond to exceeding more difficulty levels.

During training, we apply different learning rates for BERT layers and for the classifier head. Optimization is performed using AdamW. The learning rate is scheduled using a cosine annealing strategy with a linear warm-up over the first 10% of the training steps. The model is trained using the same data as for BERT-based regression. Since our data is not balanced over many classes for classification, we map the labels to 6 classes only: A1, A2, B1, B2, C1, and C2.

4.3.1 Results

The ordinal critic performs worse in terms of standard classification metrics on the same test set of 865 documents, see Table 10 and Figure 7 in Appendix B. The model achieves an accuracy of 0.76, see Table 6. However, metrics such as accuracy do not fully capture ordering information.

To better account for the severity of misclassifications, we report the Mean Absolute Error (MAE), which measures the average absolute difference between the predicted and the true labels—penalizing larger mistakes more heavily than smaller ones. MAE of 0.28 indicates that, on average, the predicted level deviates from the ground truth by about a quarter of a CEFR level. Analyzing the prediction errors in more detail, we find that 76% of the predictions exactly match the true levels, while 20% of the predictions are within one level of the ground truth. Only 4% of the documents are misclassified by more than one level—a deviation we consider “intolerable” due to the impact on downstream applications. The RMSE of 0.57, which penalizes larger errors more heavily, confirms the relatively low deviation.

In addition to accuracy, we report three metrics that better reflect the ordinal nature of CEFR levels; they include absolute- and rank-based measures, as well as agreement-based metrics.

Setup	# Documents	# Simplifications	Accuracy (%)
Baseline (no critic)	209	627	41.18
Regression Critic	212	634	50.00
Ordinal Classifier Critic	196	588	71.12

Table 7: Accuracy of simplification across different critic strategies. Each document is simplified to 3 target levels: A1, A2, and B1. A simplification is considered correct if the critic assesses it to match the target level.

- **Spearman’s rank correlation coefficient** ($\rho = 0.89$), which suggests a strong monotonic relationship between the predicted and true rankings. A higher ρ value indicates better ordinal agreement.
- **Kendall’s Tau** ($\tau = 0.83$), which confirms high ordinal agreement and is especially robust for small test sets.
- **QWK** of 0.87, which reflects substantial agreement between the predicted and true labels, while penalizing larger errors more heavily than smaller ones.

Taken together, these results indicate that the model not only achieves a high proportion of exact matches, but also preserves the ordinal structure of the CEFR scale with strong rank correlation and consistent agreement.

5 LLM-based Text Simplification

In this section, we describe how we use BERT-based difficulty models to assist LLM-based text simplification. These models act as critics to guide the simplification pipeline (see Figure 1):

- The original level of the input text is either assessed by the critic or manually labeled.
- The LLM receives the input text, the target level, and a prompt describing the target level.
- The LLM attempts to generate a simplified version of the text.
- The critic assesses the difficulty level of the output.
- If the target level is reached, the process is terminated.
- Otherwise, the LLM receives its previous output, the achieved level, the target level, and an updated prompt.
- The process is repeated for a maximum of 5 iterations.

When using the BERT-based regression model as a critic, its continuous difficulty scores are

mapped to discrete CEFR levels for compatibility with the feedback loop. When using the ordinal classification model, predictions can be used directly without mapping.

If the output is still above the target adjective after 5 iterations, the process stops. At each step, the LLM gets the feedback: This is your previous attempt to simplify the text to level X. The critic says your simplification is Y. Try harder to reach X.

5.1 Evaluation with and without Critic

We evaluate three variants of our guided text simplification pipeline: (1) **Baseline**, where the model performs one-shot simplification without critic feedback; (2) **Regression-based (REG) Critic**, where the critic is a BERT-based regression model; and (3) **Ordinal Classification (ORD) Critic**, where the critic is an ordinal classification model.

The evaluation was conducted on 220 manually annotated documents from the test set, whose original levels are above B2. Simplifications were generated to 3 target CEFR levels: A1, A2, and B1. The results are summarized in Table 7.⁹

The baseline system frequently produced simplifications that were off by one CEFR level, with common confusions such as A1 vs. A2 or A2 vs. B1. Adding the REG critic led to a moderate improvement in accuracy (+9%), suggesting that iterative refinement is beneficial. However, the most substantial improvement came from the ORD critic, which achieved 71.12% accuracy—nearly 30 percentage points higher than the baseline.

These results indicate that feedback from the ordinal critic aligns more effectively with the CEFR framework and better guides the LLM toward the target level. Table 8 shows that the LLM generates more correctly simplified outputs with the ordinal critic than with the regression critic, except for the B1 target level—where it tends to generate more

⁹Several simplification pipelines failed due to random reasons; they were not restarted, hence the number of simplification experiments in Table 7 is different for different critics.

Target	Generated	BL	REG	ORD
A1	A2	18.9	7.9	8.0
A1	A2-B1	—	8.7	—
A1	B1	5.6	7.9	0.0
A1	B1-B2	—	1.6	—
A2	A1	4.6	0.0	3.4
A2	B1	9.9	9.8	1.1
A2	B1-B2	—	5.5	—
A2	B2	0.0	1.3	0.0
B1	A1	1.9	0.0	0.9
B1	A2	13.6	1.4	15.1
B1	B2	2.9	3.9	0.0
B1	B2-C1	—	1.4	—

Table 8: Percentage of generating simplifications at an incorrect level, across three simplification pipelines. Each row indicates “incorrect” simplifications, where the generated level does not match the target level.

Target Level	Critic	Average Iterations	Maximum Iterations
A1	Regression	2.95	5
	Ordinal	2.71	5
A2	Regression	2.86	5
	Ordinal	2.21	5
B1	Regression	2.74	5
	Ordinal	1.87	4

Table 9: Average number of simplification iterations per target CEFR level using regression vs. ordinal critic.

A2-level outputs when guided by the ordinal critic. We also report the average number of iterations required to reach the target level in Table 9: using the ORD critic requires fewer iterations on average, especially for the B1 target.

For all test documents, we tracked the simplification process performed by the LLM by measuring the *intermediate* CEFR levels at each iteration, and the cosine similarity between the intermediate simplification and the original input.¹⁰ Figures 5 and 6 present the mean and standard deviation of difficulty and similarity scores across all documents, with the X-axis representing the iteration number, the left Y-axis showing difficulty scores, and the right Y-axis showing similarity scores.

Note that unlike in Tables 1 and 2, the scores produced by the ORD critic range from 0 to 5. Target level A1 in Figure 5 should be around 1 and in Figure 6—around 0. The plots show that, with the

¹⁰huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

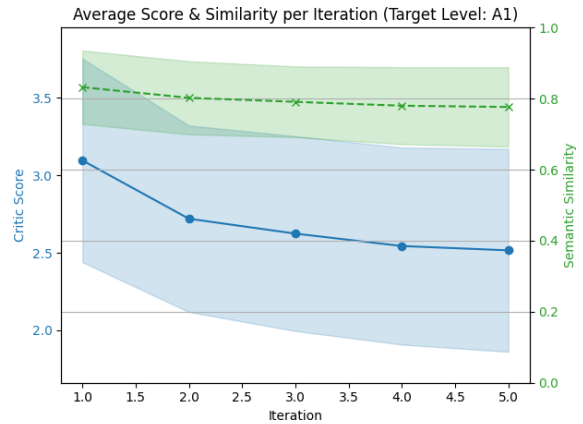


Figure 5: Regression critic with target level A1: average score and cosine similarity per simplification iteration

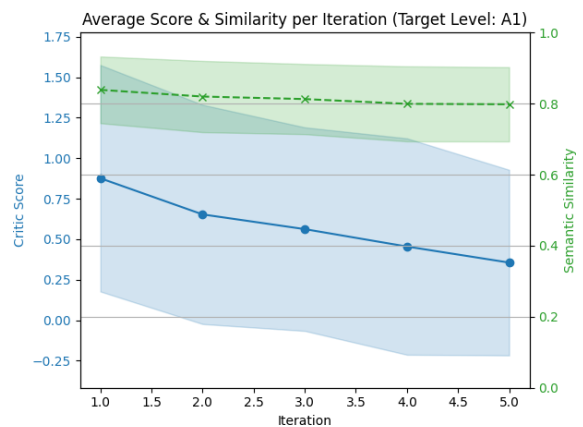


Figure 6: Ordinal critic with target level A1: average score and cosine similarity per simplification iteration

ORD critic, the difficulty of the first intermediate output is already below A2 (i.e., below 1.0), while for the REG critic it remains around B1 (around 3.0). We see a similar gain in performance of ORD over REG critic when the target level is A2 (Figures 8, 9) and B1 (Figures 10, 11) in the Appendix. Cosine similarity stays consistently at or above 0.8 in both pipelines, with slightly higher values when using the ORD critic.

5.2 Evaluation on Parallel Data

We further evaluate our approach using the Parallel Corpus of Standard Finnish–Easy Finnish (Dmitrieva and Kononova, 2023).¹¹ The Easy Finnish dataset includes news articles from the Yle archive, and consists of 1,919 manually verified pairs, each comprising an article in Easy Finnish and its corresponding article in Standard Finnish (the *source* article). We extracted 300 document pairs that are longer than 10 words and have Lev-

¹¹clarino.uib.no/comedi/editor/lb-2022111625

enshtein distance greater than 10, in order to focus on longer contexts that can be meaningfully simplified. As the dataset does not include difficulty annotations, we estimate the difficulty levels of the selected documents using the REG model and the ORD model. Both models indicate that in approximately 65% of the selected pairs, the source document is indeed more difficult than its simplified version.

We processed all source documents through the simplification pipeline once with REG and once with the ORD critic. The outputs generated by the LLM were then compared to the Easy Finnish articles using the SARI metric (Xu et al., 2016), which has been shown to correlate with human judgment of simplicity. The SARI metric is 40.6 for simplification with the BERT-based regression, and 43.1—with the ordinal model.

5.3 Manual Evaluation

An expert in teaching Finnish performed a preliminary manual analysis of the simplification results described above. We randomly selected 24 pairs of source texts and their simplified versions, generated by the pipeline with the REG and ORD models as critics. The annotator’s task was to assess whether the simplified text was indeed simpler than the source in terms of lexicon, grammar, sentence structure, and content. Although a more systematic analysis would require a larger sample and deeper investigation, several qualitative patterns emerged. Table 13 and 14 present manually analyzed pairs generated by these two simplification pipelines.

Both pipelines generally demonstrate a strong ability to simplify text: in all 24 cases, at least some parts of each sentence were successfully simplified, and in many cases, the entire sentence was made simpler (e.g., see Example 3 in Table 13). Lexical simplifications include, for instance, “*tivistää vientiponnisteluja*” (*intensify export efforts*) → “*lisätä vientiä*” (*increase exports*), “*kehittyvät taloudet*” (*developing economies*) → “*kehitysmaat*” (*developing countries*).

The REG pipeline frequently adds explanatory or contextual information, e.g., by fronting reporting clauses or expounding on the original content (see Examples 8 and 10 in Table 14). While longer texts are not necessarily more complex, such additions may increase the risks of hallucinations. In contrast, the ORD pipeline is often more effective at removing redundant information, resulting in more concise sentences. In some instances, how-

ever, the simplifications were simply paraphrases that did not reduce the overall difficulty. Whether a change constitutes a genuine simplification often depends on the reader and may require closer inspection. Both pipelines also occasionally miss clear opportunities for simplification.

In several cases, both models produced “simplified” sentences that were arguably more complex than the original; such cases are highlighted in red in the tables. For example, the verb “*tuplaantua*” (*to double*) may be easier for L2 learners than the synonym “*kaksinkertaistua*,” even though both are correct. Also, a few minor grammar problems are seen in the outputs, such as incorrect case usage in Finnish noun phrases. In other cases, the simplified sentence introduced factual ambiguities or errors, due to the model’s misunderstanding of the context or reference. More details on the results are in Appendix E.

6 Discussion and Conclusion

Our experiments with difficulty models demonstrate that small models can effectively guide text simplification performed by a large language model. Although both BERT-based difficulty models were trained on a *mix* of native and translated data, they significantly improve over the zero-shot baseline.

While the ordinal classifier performs worse on standard classification metrics, it proves more effective as critic in the simplification pipeline. We hypothesize several reasons for this. First, the regression model requires mapping floating-point difficulty scores to discrete CEFR levels, which may lose meaningful distinctions—especially during iterative simplification, where small improvements may be obscured by rounding. Second, regression assumes linear distances between levels, e.g., that the distance between A1 and A2 is equal to the distance between C1 and C2. This assumption is not required by ordinal classification.

An additional benefit of the ORD critic, currently unused, is its ability to estimate *probabilities* for CEFR thresholds—which could be interpreted as a confidence of a text being A1, A2, etc., and enable more fine-grained feedback for the LLM.

In future work, we plan to integrate feature-based and Transformer-based models, enabling the LLM to receive targeted feedback about which linguistic features in the intermediate texts do not match the desired difficulty level.

7 Acknowledgements

This work was supported in part by Business-Finland: Agency for Technology and Innovation, Project “*Easy Language for accessible workplace communication*” (Grant 4173/31/2024). We are grateful to Tiina Onikki-Rantajääskö for her insightful feedback.

8 Limitations and Ethical Considerations

While our results show that difficulty models can effectively guide LLM-based text simplification, several limitations remain. First, the models are trained and evaluated on a small dataset. Working only with Finnish may limit generalizability to other languages or domains. Second, the mapping from regression scores to CEFR levels introduces discretization errors that may obscure nuanced improvements. Third, the simplification pipeline is constrained to five iterations, which may be insufficient for particularly complex texts, and more iterations are expensive to run. Finally, we use a fixed prompt template for LLM interactions; future work could explore adaptive or dynamically generated prompts.

This work focuses on improving language accessibility, particularly for second-language (L2) learners, and aims to reduce linguistic barriers in education and communication. However, several ethical considerations must be acknowledged. First, automated simplification tools may reinforce biases present in the training data, especially if texts from specific groups or dialects are under-represented. Second, over-reliance on automated systems may inadvertently reduce the role of human educators in assessing learner needs. Lastly, misuse of simplification systems—e.g., to manipulate or oversimplify critical content—could have adverse effects. We emphasize that these systems should be used as assistive tools, not as replacements for human judgment in the context of education or public communication.

References

Sweta Agrawal and Marine Carpuat. 2023. Controlling pre-trained language models for grade-specific text simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12807–12819, Singapore.

Wejdan Alkaldi and Diana Inkpen. 2023. [Text simplification to specific readability levels](#). *Mathematics*, 11(9):2063.

Sandra Aluísio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. 2010. Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9, Los Angeles, California.

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.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331.

Kevyn Collins-Thompson and James P. Callan. 2004. A language modeling approach to predicting reading difficulty. In *HLT-NAACL 2004: Proceedings of the Human Language Technology Conference of the NAACL*, pages 193–200.

Anna Dmitrieva and Aleksandra Konovalova. 2023. [Creating a parallel Finnish-Easy Finnish dataset from news articles](#). In *Proceedings of the 1st Workshop on Open Community-Driven Machine Translation*, pages 21–26, Tampere, Finland.

Anna Dmitrieva and Jörg Tiedemann. 2021. [Creating an aligned Russian text simplification dataset from language learner data](#). In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 73–79, Kiyv, Ukraine. Association for Computational Linguistics.

Caroline Gasperin, Lucia Specia, Tiago F. Pereira, and Sandra M. Aluísio. 2009. Learning when to simplify sentences for natural text simplification. In *Proceedings of the Encontro Nacional de Inteligência Artificial (ENIA)*, Bento Gonçalves, Brazil.

Jue Hou, Maximilian W Koppatz, José Maria Hoya Quecedo, Nataliya Stoyanova, Mikhail Kopotev, and Roman Yangarber. 2019. Modeling language learning using specialized Elo ratings. In *BEA: 14th Workshop on Innovative Use of NLP for Building Educational Applications, ACL: 56th annual meeting of Association for Computational Linguistics*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

Anisia Katinskaia, Jue Hou, Anh-duc Vu, and Roman Yangarber. 2023. [Linguistic constructs represent the domain model in intelligent language tutoring](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 136–144, Dubrovnik, Croatia.

Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2017. Revita: a system for language learning and supporting endangered languages. In *6th Workshop on NLP for CALL and 2nd Workshop on NLP for Research on Language Acquisition, at NoDaLiDa*, Gothenburg, Sweden.

Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2018. Revita: a language-learning platform at the intersection of ITS and CALL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

J. Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Benjamin S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Air Station Memphis (Research Branch Report 8-75).

Antonina Laposhina. 2020. A corpus of Russian textbook materials for foreign students as an instrument of an educational content analysis. *Russian Language Abroad*, 6(283):22–28.

Antonina Laposhina, Tatiana Veselovskaya, Maria Lebedeva, and Olga Kupreshchenko. 2018. Automated text readability assessment for Russian second language learners. In *Computational Linguistics and Intellectual Technologies*, pages 403–413.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 47(1):141–179.

Serge Sharoff. 2022. What neural networks know about linguistic complexity. *Russian Journal of Linguistics*, 26(2):371–390.

A. Jackson Stenner. 1996. Measuring reading comprehension with the Lexile framework. Technical report, MetaMetrics Inc., Durham, NC.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP (BEA)*.

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.

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.

Level	Precision	Recall	F1-score	Support
A1	0.70	0.22	0.33	32
A2	0.65	0.63	0.64	98
B1	0.72	0.77	0.75	243
B2	0.14	0.19	0.16	48
C1	0.89	0.93	0.91	399
C2	0.95	0.47	0.63	45

Table 10: Performance of ORD classifier on test set

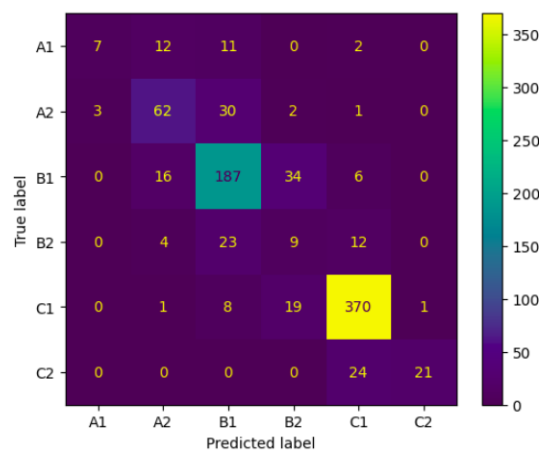


Figure 7: Confusion matrix for ORD classifier

A Baseline classification: feature-based regression

Models (A) and (B) were trained with a regularization strength $\alpha = 1.0$. For (B), we fit a Ridge regression model to $N_{\text{boot}} = 1000$ bootstrap samples of the training set, each time recording the feature coefficients. For each feature, we calculate the mean and standard deviation of its coefficient across bootstraps. The signal-to-noise ratio is defined as the absolute mean divided by the standard deviation. Features with a signal-to-noise ratio above a threshold (e.g., ≥ 1) are selected, ensuring selection of features with stable and consistently strong effects across resampled datasets.

We fit a Lasso regression model (C), which was employed for feature selection due to its ability to perform both regularization and automatic variable selection. Features with nonzero coefficients are selected, while those with coefficients shrunk to zero are excluded. The regularization parameter α for the Lasso model was selected via cross-validation using the LassoCV procedure, optimizing for mean squared error on held-out validation folds.

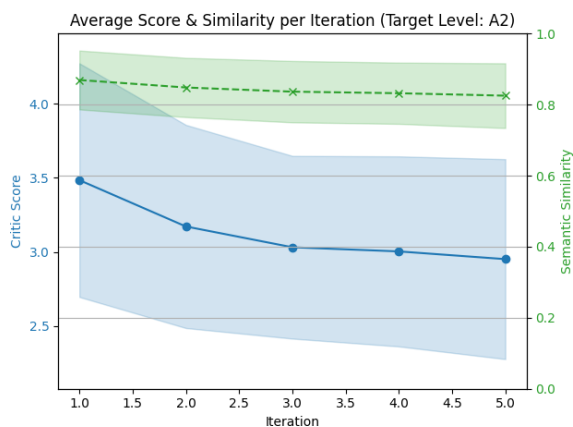


Figure 8: Regression critic with target level A2: average score and cosine similarity per simplification iteration.

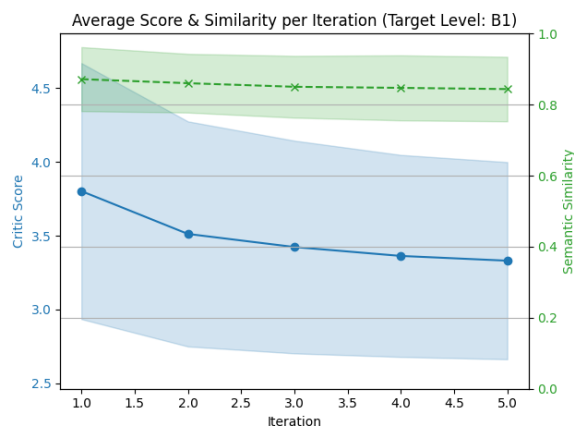


Figure 10: Regression critic, target level B1: average score and cosine similarity per simplification iteration.

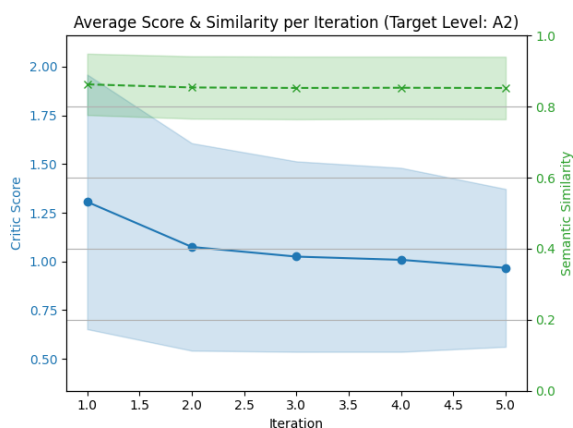


Figure 9: Ordinal critic with target level A2: average score and cosine similarity per simplification iteration.

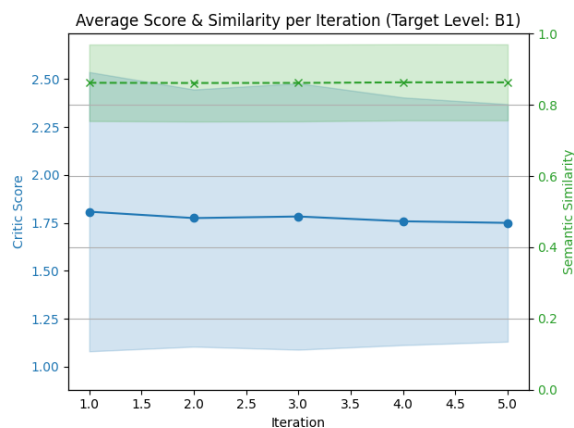


Figure 11: Ordinal critic with target level B1: average score and cosine similarity per simplification iteration.

B Ordinal classification performance

Table 10 and Figure 7 show classification metrics for the BERT-based ordinal classification difficulty model.

C LLM Prompt Templates

Below we list the CEFR-level-specific prompts used to guide GPT-4o in the simplification task. The prompts were formulated based on the definitions of CEFR levels.¹² Each prompt instructs the model to return a JSON object containing a single key "SIMPLIFICATION", with text adapted to the specified proficiency level.

Common Prompt Structure:

You must always output a JSON object with a "SIMPLIFICATION" key. You are

¹²www.coe.int/en/web/common-european-framework-reference-languages/table-1-cefr-3.3-common-reference-levels-global-scale

an expert in Finnish language and language teaching. You will be given a text in Finnish. Your task is to read it first and then to provide an adaptation into CEFR level X. Please do not significantly change the meaning of the input text. [Level-specific instructions] This is the text to simplify: {text}

Imagine that you are teaching a X learner, your adaptation should fit their proficiency level.

Level-specific Instructions:

A1 Prompt

A1 is the simplest level for beginners. The texts in A1 should be simple, with short sentences and easy grammar. The definition of a learner with A1 level is: "Can understand and use familiar every-

day expressions and very basic phrases aimed at the satisfaction of needs of a concrete type. Can introduce themselves and others and can ask and answer questions about personal details such as where someone lives, people they know and things they have. Can interact in a simple way provided the other person talks slowly and clearly and is prepared to help.”

A2 Prompt

A2 is just above the beginner level. The text in A2 should be simple and have relatively easy grammar. The definition of a learner with A1 level is: “Can understand sentences and frequently used expressions related to areas of most immediate relevance (e.g. very basic personal and family information, shopping, local geography, employment). Can communicate in simple and routine tasks requiring a simple and direct exchange of information on familiar and routine matters. Can describe in simple terms aspects of his/her background, immediate environment and matters in areas of immediate need.”

B1 Prompt

B1 is an intermediate level. The definition of a learner with B1 level is: “Can understand the main points of clear standard input on familiar matters regularly encountered in work, school, leisure, etc. Can produce simple connected text on topics which are familiar, or of personal interest. Can describe experiences and events, dreams, hopes & ambitions and briefly give reasons and explanations for opinions and plans.”

D Feature List

Tables 11 and 12 present the features used to train the feature-based models. Features shown in bold in both tables were selected via bootstrap feature selection. The features include morphophonemic, grammatical, lexical, and syntactic features. Details regarding how these features (or constructs) are detected in text can be found in (Katinskaia et al., 2023).

Consonant gradations features are identified using rule-based methods. The label “Inactive” indicates that gradation *does not* occur in the given form, e.g.: compare “nukkuu” (infinitive *to sleep*) and “nukkua” (3rd person singular *she/he sleeps*)—no gradation. The label “Active” indicates that the gradation is present, e.g.: compare “nukkua” (*to sleep*) and “nukun” (1st person singular *I sleep*)—gradation **kk** → **k**.

Lexical features include groups denoting temporal concepts, e.g.: time of the day in allative case—“aamulla” (*in the morning*), “yöllä” (*at night*); but months in inessive case: “elokuussa” (*in August*), “kesäkuussa” (*in June*), etc.

Vocabulary bags—from 1 to 10—represent frequency bins constructed from a list of over 20,000 lemmas, sorted by their frequency. The feature OOV coverage measures the proportion of words in the text whose lemmas are not found in any frequency bins, averaged over the text length.

E Simplification Results

Table 13 shows results of simplification with LLM-based pipeline guided by the ordinal classification model.

E.1 Simplification Pipeline guided by Ordinal Classification

In Example 1, the simplification was achieved by splitting the original sentence into two, grammar was simplified by replacing conditional mood with indicative: “maksaisi” (*would cost*) → “maksaa” (*costs*); “pienenisivät” (*would decrease*) → “pienenevät” (*decrease*).

Examples 2 and 3 demonstrate the removal of unnecessary information. The simplification in Example 4 resulted in a simpler information structure, as “Brittania” was moved to the beginning of the sentence. However, two sentences were combined into one, which made the overall structure more complex. The lexicon was also simplified; see the blue highlights in the simplified sentence.

Example 5 illustrates a case where the “simplified” version was actually more complex in some respects: “aiotaan nosta” (*is going to be increased*) → “suunnitellaan korotettavaksi” (*is planned to be raised*); “prosenttia” (*percent*) → “prosentilla” (*by percent*).

In Example 7, the grammar was improved: “katsoo päätöksessään” (*considers in its decision*) → “päätettiin” (*decided*); “ettei” (*that not—contracted*)

→ “että ei” (*that not—expanded*); “ei ollut syytä epäillä” (*had no reason to suspect*) → “ei voinut epäillä” (*could not suspect*). However, some parts were made more difficult: “rekrytointia hoitanut mies” (*the man who handled the recruitment*) → “rekrytoinnista vastannut mies” (*the man responsible for recruitment*).

E.2 Simplification Pipeline guided by Regression

Although the simplification in Example 8 improved contextual information—by adding “puolue” (*party*) and “lakialoite” (*legislative initiative*)—it also contains an error in orthography (a missing hyphen between “Perussuomalaiset” and “puolue”). Additionally, it introduces unnecessary and grammatically complex information, such as “lain voimaantulon jälkeen sen aiheuttamat [kustannukset]” (*the [costs] caused by it after the law comes into force*).

Example 9 demonstrates changes that made the lexicon more difficult: “yli” (*over*) → “ylittäen” (*exceeding*); “on kasvanut paljon” (*has grown a lot*) → “lisääntynyt huomattavasti” (*increased significantly*).

The simplified text in Example 10 illustrates the removal of the unnecessary word “käytännössä” and the simplification of some grammatical forms: “voisivat” (*could*) → “voivat” (*can*); “tiivistää vientiponnisteluja” (*intensify export efforts*) → “parantaa yhteistyötä viennissä” (*improve cooperation in exports*). However, it also introduces new information not present in the source (see red highlight).

The red highlights in Examples 11–13 indicate cases where the forms were made lexically, grammatically, or syntactically more complex than in the source texts.

Feature Set 1	Feature Set 2
Comparative adjective form	Consonant gradation (A type, nouns, active)
Positive adjective form	Consonant gradation (A type, nouns, inactive)
Superlative adjective form	Consonant gradation (A type, verbs, active)
Abessive case	Consonant gradation (A type, verbs, inactive)
Ablative case	Consonant gradation (“lki” A type, active)
Accusative case	Consonant gradation (“lki” A type, inactive)
Adessive case	Consonant gradation (A type ending with “uku”, active)
	Consonant gradation (A type ending with “uku”, inactive)
Allative case	Consonant gradation (B type, nouns, active)
Comitative case	Consonant gradation (B type, nouns, inactive)
Elative case	Consonant gradation (B type, verbs, active)
Essive case	Consonant gradation (B type, verbs, inactive)
Genitive case	Compound noun inflection
Illative case	Lists of confusable nouns
Inessive case	Noun paradigm “aihe”
Instructive case	Noun paradigm “bussi”
Nominative case	Noun paradigm “kala”
Partitive case	Noun paradigm “kannel”
Translative case	Noun paradigm “koditon”
Clitics of emphasis	Noun paradigm “koira”
Clitics of negation	Noun paradigm “kysymys”
Clitics of question	Noun paradigm “maa”
Clitics han	Noun paradigm “manner”
Clitics pa	Noun paradigm “mansikka”
Construction with differen factors	Noun paradigm “nainen”
Construction of type “ESSA” (Temporaalirakenne)	Noun paradigm “olut”
Construction with “Että”, perfect	Noun paradigm “ovi”
Construction with “Että”, present	Noun paradigm “puhelin”
Construction with “Että”, different actors	Noun paradigm “talo”
Construction with “Että”, same actors	Noun paradigm “uusi”
Existential construction	Noun paradigm “uutuus”
Existential construction, negative	Noun paradigm “valas”
Existential construction, positive	Noun possessive suffixes
Negative construction	Noun of time
Necessity Construction	Noun of time (day, essive)
Permission Construction	Nouns of time (hour, adessive)
Construction of possession	Noun of time (month, inessive)
Construction of possession, negative	Noun of time (season, adessive, essive)
Construction of possession, positive	Noun of time (time of the day, adessive, essive)
Construction with same actors	Noun of time (week, adessive)
Construction with “TUA” (Temporaalirakenne)	Noun of time (year, essive)
Government by adjective	Plural number
Government by noun	Singular number
Government by verb	Cardinal numeral
Government by adposition	Cardinal numeral, long
Infinitive 1	Cardinal numeral, short
Infinitive 2	Ordinal numeral
Infinitive 3	Ordinal numeral, long
Infinitive 4	Ordinal numeral, short
Infinitive 5	Agentive participle
Infinitive TUA	Perfect active participle
Conditional mood	Perfect passive participle
Conditional passive mood	Participle with possessive suffixes
Imperative mood	Present active participle
Indicative mood	Present passive participle
Potential mood	Person 1
Potential passive mood	Person 2
Possessiveness	Person 3
Negative polarity	OOV coverage
Average dependency tree depth	

Table 11: Combined linguistic feature sets for the feature-based regression model.

Feature Set 3

Demonstrative pronoun
Indefinite pronoun
Indefinite pronoun “joku”
Indefinite pronoun “kukaan”
Interrogative pronoun
Interrogative pronoun “kumpi”
Personal pronoun
Reflexive pronoun
Relative pronoun
Active object
Object of infinitive
Genitive modifier
Object of imperative
Object of passive
Object in ablative of “sense” verbs
Object in ablative of “source” verbs
Object in adessive “instrument” verbs
Object in allative of “sense” verbs
Object in allative of “communication” verbs
Object in allative of “possession” verbs
Complement construction
Object of negative
Object of verb of “feeling” (Tunnekausatiivi)
Imperfect tense
Imperfect indicative tense
Perfect tense
Perfect Indicative tense
Pluperfect tense
Pluperfect indicative tense
Present tense
List of confusable verbs
Infinitive
Infinitive bases
Verb paradigm “haluta”
Verb paradigm “juosta”
Verb paradigm “nousta”
Verb paradigm “saada”
Verb paradigm “sanoa”
Verb paradigm “tarvita”
Verb paradigm “tehdä”
Verb paradigm “tulla”
Verb paradigm “vanheta”
Participle
Active voice
Passive voice
Passive imperfect voice
Passive perfect voice
Passive pluperfect voice
Passive present voice
Average number of POS
Average sentence length
Vocabulary bag 1
Vocabulary bag 2
Vocabulary bag 3
Vocabulary bag 4
Vocabulary bag 5
Vocabulary bag 6
Vocabulary bag 7
Vocabulary bag 8
Vocabulary bag 9
Vocabulary bag 10

Table 12: Linguistic features used for the feature-based regression model.

Ordinal classifier as critic:

ID: 1

Source: Perussuomalaiset arvioivat, että “Lex Lindström” maksaisi noin 20-30 miljoonaa euroa, ja menot pienensivät vuosittain.

Level: C1

Simplified: Perussuomalaiset-puolue arvioi, että “Lex Lindström” maksaa noin 20-30 miljoonaa euroa. He arvioivat, että kulut pienenevät vuosittain.

Level: B2

ID: 2

Source: Jo täällä asuvilla maahanmuuttajilla on Nafisa Yeasminin mukaan tärkeää rooli uusien tulijoiden sopeutumisessa Suomeen. — Me tunnemme jo suomalaisen kulttuurin ja kielen.

Level: B2

Simplified: Suomessa asuvat maahanmuuttajat auttavat uusia ihmisiä. — Me tunnemme Suomen kulttuurin ja kielen.

Level: A2

ID: 3

Source: Käytännössä Pohjoismaat voisivat esimerkiksi tiivistää puhtaan teknologian vientiponnisteluja kehittyvissä talouksissa.

Level: B2

Simplified: Pohjoismaat voisivat esimerkiksi lisätä puhtaan teknologian vientiä kehitysmaihin.

Level: A2

ID: 4

Source: Siirtymäkauden ajan Britannia säilyy EU:n sisämarkkinoilla, maksaa maksuja ja soveltaa EU-lainsäädäntöä. Se ei kuitenkaan osallistu päätöksentekoon.

Level: B2

Simplified: Britannia on vielä vähän aikaa EU:ssa mukana, noudattaa EU:n sääntöjä ja maksaa EU:lle, mutta ei voi osallistua päätöksiin.

Level: A2

ID: 5

Source: Korkeinta kokopäivähoidon maksua aiotaan nostaa peräti 22 prosenttia. Tällä hetkellä ylin maksu on 283 euroa kuukaudessa ensimmäisestä lapsesta.

Level: C1

Simplified: Korkeinta kokopäivähoidon maksua suunnitellaan korotettavaksi jopa 22 prosentilla. Tällä hetkellä maksetaan enintään 283 euroa kuukaudessa ensimmäisestä lapsesta.

Level: B2

ID: 6

Source: Oikeus katsoo päätöksessään, ettei rekrytointia hoitaneella miehellä ollut syytä epäillä saamaansa ohjetta. — Olemme tyytyväisiä.

Level: C1

Simplified: Oikeus päättyi, että rekrytoinnista vastannut mies ei voinut epäillä saamaansa ohjetta. — Olemme tyytyväisiä.

Level: B1

ID: 7

Source: Suomalaisten testamenttilahjoittajien sydäntä lähellä on erityisesti lasten ja naisten hyvinvointi ja koulutus niin Suomessa kuin kehitysmaissa. — Lahjoittajat sanovat, että kun satsaa tyttöihin ja naisiin, niin yhteiskuntakin muuttuu vahvasti.

Level: B2

Simplified: Suomalaiset lahjoittajat haluavat auttaa lapsia ja naisia Suomessa ja muissa maissa. He sanovat, että kun auttaa tyttöjä ja naisia, myös yhteiskunta paranee.

Level: A2

Table 13: Pairs of source texts and their simplified versions. Simplifications were guided by the ordinal classifier. Underlined text in the source indicates which parts were changed; blue highlights mark successful simplification. Red highlights parts which are erroneous or harder than the source.

Regression model as critic:

ID: 8

Source: Perussuomalaiset arvioivat, että "Lex Lindström" maksaisi noin 20-30 miljoonaa euroa, ja menot pienenisivät vuosittain.

Level: B2-C1

Simplified: Perussuomalaiset puolue on arvioinut, että 'Lex Lindström' — lakialoitteen taloudellinen vaikutus olisi noin 20-30 miljoonaa euroa. Lain voimaantulon jälkeen sen aiheuttamat kustannukset pienentyisivät vuosittain.

Level: B2-C1

ID: 9

Source: Kaiken kaikkiaan lemmikkien lääkintä — ja hoitomenot ovat tuplaantuneet kymmenessä vuodessa yli 200 miljoonaan euroon. Lemmikkien määrä on kasvanut samassa ajassa paljon hitaammin.

Level: B2-C1

Simplified: Yhteensä lemmikkien lääkintä — ja hoitokulut ovat kasvaneet kaksinkertaisiksi kymmenessä vuodessa ylittäen 200 miljoonaa euroa. Samaan aikaan lemmikkien määrä on lisääntynyt huomattavasti hitaammin.

Level: B2

ID: 10

Source: Käytännössä Pohjoismaat voisivat esimerkiksi tiivistää puhtaan teknologian vientiponnisteluja kehittyvissä talouksissa.

Level: B2-C1

Simplified: Pohjoismaat voivat parantaa yhteistyötä puhtaan teknologian viennissä kehittyviin talouksiin esimerkiksi jakamalla innovaatioita ja edistämällä yhteisiä hankkeita.

Level: B2-C1

ID: 11

Source: Korkeinta kokopäivähoidon maksua aiotaan nostaa peräti 22 prosenttia. Tällä hetkellä ylin maksu on 283 euroa kuukaudessa ensimmäisestä lapsesta.

Level: B2-C1

Simplified: Kokopäivähoidon korkeinta maksua suunnitellaan nostettavaksi jopa 22 prosenttia. Nyt korkein maksu on 283 euroa kuukaudessa ensimmäisestä lapsesta.

Level: B2

ID: 12

Source: Jämsä ja Liukkokin ovat muuttaneet kulutustottumuksiaan. — Kaupassa voin jättää muovipussin ostamatta ja ruokaillessa syön mieluummin kasviksia tai kanaa naudanlihan sijaan, Jämsä kertoo.

Level: B1-B2

Simplified: Jämsä ja Liukko ovat muuttaneet tapojaan kuluttaa. Jämsä sanoo, että kaupassa hän voi päättää olla ostamatta muovipussia ja ruokailutilanteissa hän valitsee mieluummin kasvikset tai kanan naudanlihan sijasta.

Level: B1-B2

ID: 13

Source: Suomalaisten testamenttilahjoittajien sydäntä lähellä on erityisesti lasten ja naisten hyvinvointi ja koulutus niin Suomessa kuin kehitysmaissa. — Lahjoittajat sanovat, että kun satsaa tyttöihin ja naisiin, niin yhteiskuntakin muuttuu vahvasti.

Level: B2

Simplified: Suomalaiset testamenttilahjoittajat haluavat parantaa lasten ja naisten elämää ja koulutusta Suomessa ja kehitysmaissa. He uskovat, että kun tukee tyttöjä ja naisia, koko yhteiskunta voi kehittyä vahvemaksi.

Level: B1

Table 14: Pairs of source texts and their simplified versions. Simplifications were guided by the regression model. Underlined text in the source indicates which parts were changed; blue highlights mark successful simplification. Red highlights parts which are erroneous or harder than the source.

Are Large Language Models for Education Reliable for All Languages?

Vansh Gupta^{*†} Sankalan Pal Chowdhury^{*†}
Vilém Zouhar[†] Donya Rooein[‡] Mrinmaya Sachan[†]

{guptav, spalchowd, vzouhar, msachan}@ethz.ch donya.rooein@unibocconi.it

[†]ETH Zurich [‡]Bocconi University

Abstract

Large language models (LLMs) are increasingly being adopted in educational settings. These applications expand beyond English, though current LLMs remain primarily English-centric. In this work, we ascertain if their use in education settings in non-English languages is warranted. We evaluated the performance of popular LLMs on four educational tasks: identifying student misconceptions, providing targeted feedback, interactive tutoring, and grading translations in eight languages (Mandarin, Hindi, Arabic, German, Farsi, Telugu, Ukrainian, Czech) in addition to English. We find that the performance on these tasks somewhat corresponds to the amount of language represented in training data, with lower-resource languages having poorer task performance. However, at least some models are able to more or less maintain their levels of performance across all languages. Thus, we recommend that practitioners first verify that the LLM works well in the target language for their educational task before deployment.

1 Introduction

Education is a multilingual, multicultural endeavour. AI-based technologies have recently shown the potential to improve students' learning experiences, and educational systems worldwide are increasingly adopting these tools (Gligorea et al., 2023). From personalized instruction and targeted feedback to appropriate content generation and interactive tutoring, these tools offer solutions to key educational challenges (Leon, 2024; Rooein et al., 2024; Mosher et al., 2024). Large language models such as GPT, Gemini, and Llama (OpenAI, 2023; Team, 2024; Roulmliotis et al., 2023) have become

particularly influential, with early evidence suggesting their ability to support teachers or scaffold student learning (Kasneci et al., 2023; Alqahtani et al., 2023).

Although most of these LLMs are trained on multilingual corpora (OpenAI, 2019; Nvidia, 2022; Peng et al., 2023; Gu and Dao, 2023), they are still overwhelmingly English-centric (Argoub, 2022; Ruder et al., 2022, Table 1). Inadequate adaptation to local languages in an educational setting risks diminishing their utility and exacerbating existing inequalities by privileging dominant languages and cultures. The question of multilingualism arises in every domain where LLMs are applied (Lai et al., 2023; Ahuja et al., 2023, 2024). However, it is especially important in the field of education, which has seen wide use of LLMs despite the high stakes (Alhafni et al., 2024; Raheja et al., 2023; Naismith et al., 2023). Without rigorous evaluation tailored to educational tasks across languages, deploying LLMs in classrooms may introduce new forms of harm, including misinformation, misalignment with curricula, or culturally inappropriate content (Almasoud et al., 2025).

In this work, we present an empirical investigation of the capabilities of frontier LLMs on educational tasks across several languages. We identify four education-related tasks (identifying student misconceptions, providing targeted feedback, interactive tutoring, and translation grading) with well-defined language-agnostic metrics. We then evaluate several frontier LLMs (Claude, Gemini, GPT4o, Llama, and Mistral) on these tasks in eight languages (Mandarin, Hindi, Arabic, German, Farsi, Telugu, Ukrainian, and Czech) in addition to English.

Our results show that though performance in English still dominates, other languages are not too far behind, at least for GPT4o and Gemini-2.0-flash, which emerge as the best models. We also find that using prompts in the language of the task

^{*}Equal Contribution

⁰We release the collected dataset and code at github.com/eth-lre/multilingual-educational-llm-bias. The dataset comprises 313,500 automatically evaluated model outputs across seven languages, four tasks, and six models.

is rarely helpful compared to English prompts.

2 Methods

We select our set of tasks based on 3 desiderata:

- **Relevant to Education:** We focus on tasks that LLMs would encounter specifically in the role of tutors, teachers, or teaching assistants. We do not cover tasks like question answering or solving math questions, which, while possibly being relevant to education, are more general tasks that are primarily studied in other contexts.
- **Have a Language Component:** We avoid tasks whose formulation uses purely notation, for example, solving a math equation. If the equation is provided in mathematical notation, the task would remain unchanged between different languages, making the question of multilingual performance moot.
- **Language Invariant Evaluation:** Finally, we need evaluation metrics that remain comparable across languages to compare performance across different languages efficiently. This means we cannot rely on language-dependent metrics like BLEU or COMET (Papineni et al., 2002; Rei et al., 2020).

Based on these, we selected following four tasks:

Task 1: Misconception identification. An important aspect of teaching is fixing student misconceptions, which first requires identifying the student misconception (Liu et al., 2023). We build this task on the EEDI Math Questions Dataset, which contains thousands of multiple-choice questions with four answer choices. For many of the wrong choices, we have expert-annotated misconceptions that could lead to a student picking the said choice. We leverage these to build our task. The LLM is given a multiple-choice question, the student’s (incorrect) answer, and four possible misconceptions. The candidate misconceptions include the true misconception identified by experts and three distractors chosen at random from the other misconceptions present in the dataset. The LLM must pick the correct misconception from these four options (see Example 1 for an example). We evaluate the LLM performance by reporting accuracy in predicting the student misconception. Since the model must pick one of four options, a random baseline has an accuracy of 25%.

Task 2: Feedback selection. A key step towards fixing students’ misconceptions is generating feedback to alleviate them. The EEDI dataset discussed above also includes feedback for all the choices we use for this part. The LLM is again given a multiple-choice question, the student’s answer, and this time, a set of four possible feedbacks, out of which the LLM must select the feedback corresponding to the student’s answer. Note that while there are 4 possible feedbacks, one corresponds to the correct answer. This one is easily identifiable as it reinforces the student’s answer, while the feedbacks corresponding to wrong answers all try to make the student realize their mistake. As an example, see Option C in both parts of 2, which are the only options in their respective questions that do not start with a negative tone. Therefore, if the selected answer is also the correct answer to the problem, the LLM might be able to pick the correct feedback using some shallow semantics, which we want to avoid. Therefore, we ensure that the selected answer is always incorrect. The random baseline has an accuracy of 25%, or 33% if choosing among responses to the wrong answer.

Task 3: Tutoring. For more complex misconceptions, a single-turn feedback often does not suffice, and fixing the misconception requires a multi-turn conversation between the student and the teacher, also known as tutoring. (Bloom, 1984; Cohen et al., 1982) This involves a teacher LLM trying to help the student identify and fix an error in their solution. We evaluate the tutoring ability of the LLM by having it tutor a weaker LLM, which acts as the student. Both the teacher and the student are given the question, but only the teacher LLM can access the correct answer. The student LLM is instructed to stick to the wrong solution unless it sees strong justification to shift. The teacher and the student take turns to send messages, with the teacher’s goal being to get the student model to the correct answer, without revealing the answer themselves. The teacher LLM is considered to get a *success* if the student LLM states the answer. If the teacher reveals the answer before the student has gotten to it, it is counted as *telling*. An *adjusted success* occurs when there is a success but no telling. The task is finally evaluated by Tutoring score (Pal Chowdhury et al., 2024), which is the harmonic mean between success rate and adjusted success rate.

This task differs from the other tasks on this list

	Language family	Script	Wikipedia	CommonCrawl	Speakers
English	Germanic	Latin	6973K	42.8%	1500M
Mandarin	Sino-tibetan	Hanzi ¹	1480K	5.8%	1184M
Hindi	Indo-Iranian	Brahmic	165K	0.20%	609M
Arabic	Afro-Asiatic	Abjad	1259K	0.68%	411M
German	Germanic	Latin	3021K	5.5%	411M
Farsi	Indo-Iranian	Abjad	1034K	0.74%	134M
Telugu	Dravidian	Brahmic	111K	0.02%	96M
Ukrainian	Slavic	Cyrillic	1371K	0.62%	39M
Czech	Slavic	Latin	566M	0.10%	12M

Table 1: Language information, number of speakers (Ethnologue 2025), and global representations of tested languages in NLP (Wikipedia Articles and proportion in CommonCrawl in March 2025).

in at least two significant ways. First, it is a multi-turn conversation task, so there is no scope for guessing the answer. Secondly, the final evaluation depends on the performance of the student LLM, so the multilingual capabilities of the student LLM also restrict the applicability of this task. These factors make this task both slower to run and more complex for the LLMs.

Task 4: Translation grading. A common field of education that has seen an increase in the use of LLMs is Language learning (Klimova et al., 2024; Zhu et al., 2024). A representative task from this field is to assign a grade to a translation provided by a student. While we lack proper datasets across languages with translations and their appropriate grades, we can approximate this task by the fact that *the machine translation of a sentence should receive a higher grade than the exact translation with one word replaced by a random word*. We use English sentences from Duolingo’s English→Spanish SLAM dataset (Settles, 2018), which are machine translated to other languages. We chose this dataset because it is meant to be used for translation, so it should contain fewer hard-to-translate sentences. We filter out simple sentences that do not end with a full stop or have fewer than five words. For each translated sentence, we then create a corresponding *perturbed translation* by replacing one of the words in the sentence with a different word selected at random from the other sentences in the dataset, disrupting both the fluency and adequacy of the translation. The LLM judges both the original and perturbed versions on a scale from 1 (completely incorrect) to 5 (perfect), with the expectation that it should assign a strictly lower score to the perturbed version. A model assigning all scores at random would therefore score around 40%.

¹Alternately referred to as Kanji, Hanja or Hantu

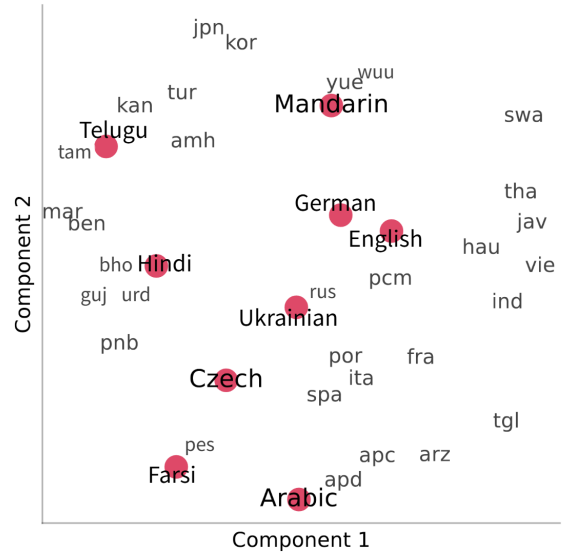


Figure 1: Multidimensional Scaling projection of languages based on syntax features from URIEL/lang2vec. Languages used in our experiments are highlighted and shown with full names, others are in ISO 639/set 2.

Language selection. We choose eight languages for experiments: Mandarin, Hindi, Arabic, German, Farsi, Telugu, Ukrainian, Czech in addition to English for comparison. This language selection reflects diverse linguistic properties, varying levels of representation in training data, and different language families (Foundation, 2024), see Table 1. Hindi (Indo-Aryan) and Telugu (Dravidian) represent major languages from the Indian subcontinent that use the Brahmic script and are under-represented in both CommonCrawl and Wikipedia. German and Mandarin, on the other hand, are examples of languages well represented in both CommonCrawl and Wikipedia. Farsi and Arabic offer insights into LLM performance on a right-to-left Abjad script, whereas Ukrainian and Czech allow us to study generalisation in medium resource morphologically rich languages, using the Cyrillic and

Language	Questions	Misconception	Feedback	Translation
Mandarin	0.593	0.607	0.666	0.781
Hindi	0.455	0.546	0.593	0.831
Arabic	0.596	0.659	0.605	0.793
German	0.623	0.642	0.697	0.792
Farsi	0.574	0.644	0.708	0.840
Telugu	0.518	0.569	0.578	0.639
Ukrainian	0.607	0.642	0.682	0.821
Czech	0.611	0.626	0.663	0.805

Table 2: Average COMET₂₃^{DA,XL} scores for different languages for different components of the tasks. The **Questions** are used for both Misconception and Feedback tasks. The Tutoring task is not translated.

Latin scripts, respectively.

To assess the typological diversity of our selected languages, we used the URIEL typological database (Littell et al., 2017) with `lang2vec`, which provides dense vector representations of languages based on a range of typological, phylogenetic, and geographical features. As recommended by the package, we extracted syntax features with k-NN predictions for the missing values for a set of 40 languages, constructed as the union of our core experimental languages and the most widely spoken languages worldwide according to Ethnologue (Eberhard et al., 2025). We projected each language feature vector into two dimensions using Multidimensional Scaling, producing a 2D language similarity plot. This allows us to visualise (see Figure 1) the relative syntactic diversity of our selected languages and confirm that they span a broad typological space. The visualisation demonstrates that our language selection (highlighted) is well distributed across the typological landscape.

Translation. We obtain our tasks in all the above-mentioned languages by machine translation. Following the GPT4 Technical Report (OpenAI, 2023, Figure 5), we use Azure Translate to translate all our examples to the target languages. However, this introduces an additional noise source for tasks performed in languages other than English. In fact, after reviewing some of the translations manually, it does look like the translations, though decent, are not as easy to follow as their English counterparts. This finding is further corroborated by COMET₂₃^{DA,XL} (Rei et al., 2023) scores of the translations (see Table 7). This means that any differences we observe between English and other-language performance cannot be conclusively attributed to the LLM being tested. However, we can still compare the performance of different LLMs across the same language, as the same translation

was used for all LLMs. Further, if at least one LLM performs well in a task on a given language, we can be reasonably certain that the translation for that task-language pair was also good enough.

Models and prompts. We evaluate six state-of-the-art LLMs praised for their multilingual capabilities: GPT-4o (OpenAI, 2023), Gemini 2.0 Flash (Team, 2024), Claude 3.7 Sonnet (Anthropic, 2024), Llama 3.1 405B (Grattafiori et al., 2024), Mistral Large 2407 (AI, 2024; Jiang et al., 2023), and Command-A (Cohere et al., 2025). We leave all sampling parameters to their defaults. For prompts, we use a simple chain of thought prompting method, where the model is first asked to explain why it would pick a certain answer, and then asked to choose it in a separate prompt. Based on literature (Mondshine et al., 2024; Huang et al., 2023), it is unclear whether or not it is beneficial to translate the prompt itself to the target language or keep it in English, so we try both options.^{2,3}

For each task, we use 1000 examples for reporting our results, sampled at random from the dataset, except for 200 examples in the tutoring task, which is multi-turn.

3 Results

In this section, we describe the results of five popular large language models on the four tasks described in Section 2. The main results are shown in Tables 3 to 6.

English is easiest for LLMs. The gap between English and other languages is large in general. On

²A weaker model roleplays the student model used in the tutoring task to be consistent with the original work. We only use the original prompts because it does not work well with non-English prompts.

³We machine-translate the prompts and manually verify (with L1/L2 language knowledge) the translation adequacy.

Input	Options
<p>Question: Which number is the greatest? Student Answer: 5.0001 Right Answer: 5.2</p>	<p>A: Believes the mean is total frequency divided by something, B (correct): Thinks the more digits a number has the greater it is, regardless of place value, C: Believes parallel lines have gradients that multiply to give -1, D: When multiplying by a multiple of 10, gives an answer 10 times bigger than it should be</p>
<p>Question: What is the lowest common multiple of 8 and 4? Student answer: 4 Right Answer: 8</p>	<p>A: Subtracts instead of adds when answering worded problems, B (correct): Confuses factors and multiples, C: Rounds up instead of down, D: Adds instead of multiplying when expanding bracket</p>

Example 1: Two examples of the misconception identification task (English).

Input	Options
<p>Question: 6 pencils cost £1.50. How much do 3 pencils cost? Student answer: 25p</p>	<p>A: I think you have made an arithmetic error when halving £1.50. Use short division to divide by two, B: I think you have used the incorrect notation for money. Consider how the monetary values in the question are written, C (correct answer): If 6 pencils cost £1.50, then 3 pencils cost half of £1.50, which is £0.75 or 75p., D (student answer): I think you have found the cost for one pencil. The question asks for the cost of 3 pencils.</p>
<p>Question: A film starts at 8.50pm. The film lasts 2 hours and 52 minutes. What time does the film finish? Student answer: 11.02pm</p>	<p>A (student answer): This isn't quite right. Remember that there are 60 minutes in an hour, not 100 :), B: I think you've confused your method a little. Noticing that 2 hours and 52 minutes is just 8 minutes less than 3 hours is super, just make sure you add and subtract in the correct directions though :), C: Almost there! Take care to notice how many hours and minutes you're adding here. Is your answer 2 hours and 52 minutes later than 8.50pm?, D (correct answer): Adding 2 hours to 8.50pm gives 10.50pm. Adding 10 minutes on takes us to 11.00pm, and adding the remaining 42 minutes gives 11.42pm.</p>

Example 2: Two examples of the feedback selection task (English).

Math Problem	Student's (Incorrect) Solution	Correct Solution
<p>Sam sells bread. He has a target of selling 120 crates of bread in a week. One week he was closed on Monday and Friday. Over the weekend he sold 20 crates. On Tuesday he sold 15 crates, on Wednesday 12 crates, and Thursday 18 crates. By how many crates was Sam off from his target for the week?</p>	<p>Sam had 5 days to sell bread because he was closed on Monday and Friday. He sold a total of $20 + 15 + 12 + 18 = 65$ crates of bread from Tuesday to Thursday. Adding the 20 crates he sold over the weekend, Sam sold a total of $65 + 20 = 85$ crates of bread in a week. Sam was off from his target by $120 - 85 = 35$ crates of bread.</p>	<p>During the whole week Sam sold $15 + 12 + 18 + 20 = 65$ crates. Sam was off his target by $120 - 65 = 55$ crates.</p>
<p>Sophia is thinking of taking a road trip in her car, and would like to know how far she can drive on a single tank of gas. She has traveled 100 miles since last filling her tank, and she needed to put in 4 gallons of gas to fill it up again. The owner's manual for her car says that her tank holds 12 gallons of gas. How many miles can Sophia drive on a single tank of gas?</p>	<p>Sophia used 4 out of the 12 gallons of gas in her tank, so there are $12 - 4 = 8$ gallons of gas left in the tank. If Sophia can drive 100 miles on 4 gallons of gas, then she can drive $100/4 = 25$ miles per gallon. Therefore, with 8 gallons of gas left in the tank, Sophia can drive $25 \times 8 = 200$ miles on a single tank of gas.</p>	<p>To find miles per gallon, divide 100 miles / 4 gallons = 25 miles per gallon. To find how far Olivia can go on a single tank, multiply 25 miles per gallon \times 12 gallons = 300 miles.</p>

Example 3: Two examples of the tutoring task.

English Source	Original Translation	Perturbed Translation	Language
It is a kind of tomato.	它是一种番茄	这位工程师有一个家庭	Mandarin
	वह एक तरह का टमाटर है।	भाई एक तरह का टमाटर है।	Hindi
	هذا نوع من الطماطم.	هذا نوع من التفاح.	Arabic
	Es ist eine Art Tomate	Katze ist eine Art Tomate	German
	این نوعی گوجهفرنگی است.	این رنگ گوجهفرنگی است.	Farsi
	ಇದಿ ಒಕ ರಕಮೈನ ಟಮಾಟಾ.	ಇದಿ ಕನುಗೊಂಟಾಢು ರಕಮೈನ ಟಮಾಟಾ.	Telugu
Він впливають різновидом томатів.	Він є різновидом томатів.	Ukrainian	
Je to druh rajčete.	matka to druh rajčete.	Czech	

Example 4: A single example of the translation grading task for non-English languages.

Language	English prompt						Translated prompt					
	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A
English	97.6%	96.2%	95.1%	94.0%	95.0%	95.3%	97.6%	96.2%	95.1%	94.0%	95.0%	95.3%
Mandarin	·95.8%	95.2%	·92.5%	·92.1%	·92.8%	·92.9%	96.5%	95.0%	·91.7%	93.8%	·92.9%	94.1%
Hindi	·94.5%	·93.2%	·91.9%	·89.8%	·91.8%	·93.2%	·95.5%	·93.6%	·89.6%	·90.4%	·90.6%	·91.4%
Arabic	·95.9%	·93.0%	·92.0%	·86.0%	·92.6%	·93.4%	·95.9%	·93.0%	·92.8%	·90.9%	·92.0%	94.0%
German	·96.0%	96.2%	94.6%	·84.6%	95.1%	95.2%	·95.9%	96.6%	94.0%	★74.0%	94.9%	95.2%
Farsi	·94.8%	·93.3%	·93.0%	·87.5%	·92.7%	·93.1%	·95.1%	·94.4%	★68.0%	·88.3%	★66.9%	·93.6%
Telugu	·95.2%	·92.2%	·89.9%	·86.9%	·89.7%	·85.5%	·94.2%	·90.8%	★68.6%	·83.6%	★35.5%	★77.9%
Ukranian	·95.7%	94.9%	·92.9%	93.3%	94.4%	94.9%	·95.6%	·94.3%	★56.6%	·90.4%	94.2%	93.9%
Czech	96.9%	95.1%	94.5%	92.3%	94.5%	94.1%	96.6%	95.8%	★70.2%	·81.6%	★41.0%	94.5%

Table 3: Results (accuracy) for the **misconception identification** task. We mark results significantly lower (at least 10%=★, at least 5%=*, otherwise ·) than English with a one-sided 95% confidence t-test.

Language	English prompt						Translated prompt					
	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A
English	53.4%	38.2%	17.0%	51.1%	48.5%	39.7%	53.4%	38.2%	17.0%	51.1%	48.5%	39.7%
Mandarin	·49.6%	★29.7%	·12.3%	·43.0%	·40.1%	·31.8%	★41.1%	★19.2%	★5.8%	★30.3%	★30.3%	★27.8%
Hindi	·48.7%	35.6%	·13.0%	·43.6%	·40.5%	·31.6%	★32.1%	★13.4%	★6.2%	·44.3%	★18.6%	★18.8%
Arabic	·49.6%	★28.7%	·13.9%	·45.3%	★38.8%	·33.3%	·48.8%	★10.7%	16.3%	48.1%	★27.8%	★28.9%
German	52.5%	·32.1%	15.0%	·46.4%	·42.4%	·32.8%	50.6%	·30.8%	15.6%	·44.4%	★39.4%	37.6%
Farsi	50.2%	★27.9%	·11.3%	·44.9%	·41.3%	★30.9%	·45.9%	·31.6%	16.3%	·44.0%	★33.5%	·35.5%
Telugu	·45.2%	★27.6%	·10.4%	·43.4%	★34.0%	★26.3%	★13.9%	★12.7%	★6.1%	★37.7%	★15.5%	★9.5%
Ukranian	50.3%	·33.2%	·13.0%	·44.8%	·41.3%	·32.2%	★35.9%	★19.6%	★8.1%	52.8%	★31.0%	★27.2%
Czech	49.9%	37.8%	·14.1%	·46.5%	·41.6%	★30.7%	★42.7%	★26.1%	19.2%	·46.6%	★35.5%	·35.6%

Table 4: Results (accuracy) for the **feedback selection** task. We mark results significantly lower (at least 10%=★, at least 5%=*, otherwise ·) than English with a one-sided 95% confidence t-test.

Language	Harmonic mean						Success/1-Telling					
	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A
English	94.7%	97.0%	22.1%	93.0%	82.0%	95.5%	96.0/2.5%	97.5/1.0%	96.5/84.0%	93.5/1.0%	82.0/0.0%	96.0/1.0%
Mandarin	89.8%	89.0%	26.4%	79.7%	79.7%	88.2%	94.0/8.0%	90.5/3.0%	90.0/74.5%	80.5/1.5%	80.0/0.5%	93.0/9.0%
Hindi	90.5%	92.7%	24.2%	★72.2%	73.5%	·88.4%	95.0/8.5%	93.0/0.5%	89.5/75.5%	77.5/10.0%	73.5/0.0%	91.0/5.0%
Arabic	91.4%	89.7%	24.3%	·84.2%	75.2%	87.4%	94.5/5.9%	90.0/0.5%	91.0/77.0%	86.0/3.5%	75.5/0.5%	93.0/10.5%
German	90.7%	91.2%	23.4%	84.2%	77.2%	·86.3%	92.5/3.5%	92.0/1.5%	88.0/74.5%	85.0/1.5%	77.5/0.5%	90.5/8.0%
Farsi	·85.6%	★81.3%	28.7%	77.2%	·65.8%	·77.8%	89.0/6.5%	87.5/11.5%	91.5/74.5%	78.0/1.5%	69.5/7.0%	91.0/23.0%
Telugu	★50.1%	★39.5%	27.7%	·58.9%	★2.9%	★40.7%	77.5/40.5%	77.5/51.0%	85.5/69.0%	61.0/4.0%	59.0/57.5%	63.5/33.5%
Ukranian	91.2%	91.5%	23.5%	·81.2%	71.5%	90.9%	93.0/3.5%	92.0/1.0%	91.5/78.0%	84.0/5.5%	71.5/0.0%	93.5/5.0%
Czech	★43.8%	★44.1%	17.2%	70.2%	★2.9%	★21.5%	65.5/32.5%	73.5/42.0%	90.0/80.5%	71.5/2.5%	52.5/51.0%	77.0/64.5%

Table 5: Results (harmonic mean, success, and telling) for the **tutoring** task. We mark results significantly lower (at least 10%=★, at least 5%=*, otherwise ·) than English with a one-sided 95% confidence t-test when occurring in both success and telling. Telling is flipped such that higher is better.

Language	English prompt						Translated prompt					
	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A
Mandarin	100.0%	99.3%	98.9%	99.9%	99.5%	99.9%	99.9%	99.4%	24.8%	99.6%	99.4%	99.9%
Hindi	91.5%	74.1%	92.1%	77.6%	82.4%	77.9%	93.8%	88.5%	56.5%	86.5%	87.6%	81.3%
Arabic	98.6%	97.9%	99.2%	98.8%	97.5%	99.0%	98.8%	98.3%	67.2%	98.6%	97.8%	97.9%
German	98.2%	97.9%	97.9%	98.2%	98.2%	98.2%	98.5%	98.3%	29.9%	98.0%	98.3%	97.8%
Farsi	95.3%	93.5%	96.0%	96.4%	92.3%	96.6%	96.8%	96.0%	67.0%	96.4%	94.1%	96.2%
Telugu	77.2%	33.7%	81.0%	51.9%	48.7%	25.2%	82.8%	46.8%	40.7%	82.1%	67.1%	15.6%
Ukranian	98.0%	97.3%	96.9%	96.5%	97.3%	98.3%	98.1%	97.9%	85.3%	97.7%	98.4%	98.2%
Czech	98.7%	98.3%	98.9%	98.3%	97.5%	98.8%	99.3%	98.8%	80.8%	98.7%	99.5%	99.2%

Table 6: Results (accuracy) for the **translation grading** task.

average⁴ across all tasks (excluding translation) and models, English has 70.9%, in contrast to 63.1% (Hindi), 55.3% (Czech), 67.8% (Ukranian), 49.7% (Telugu), 66.2% (Farsi), 66.8% (German), 64.6% (Mandarin) and 67.4% (Arabic). This in itself does not make it clear if the loss is due to the LLMs be-

ing weak or the translation quality being poor. The poor performance on Telugu is largely driven by Command-A and Mistral. The former is unsurprising as Telugu is the only language in our list that is not officially supported by it (Cohere et al., 2025). On the other hand, Mistral lists only 12 supported languages of which we test only Hindi, Arabic, German and Chinese. Telugu also has the lowest representation in CommonCrawl and Wikipedia,

⁴Averaging here is done to give a general idea, but we must note that the scores are not equivalent. We use Accuracy for tasks 1 and 2 but Tutoring Score for Task 3

Language	Questions	Misconception	Feedback	Translation
Mandarin	0.593	0.607	0.666	0.781
Hindi	0.455	0.546	0.593	0.831
Arabic	0.596	0.659	0.605	0.793
German	0.623	0.642	0.697	0.792
Farsi	0.574	0.644	0.708	0.840
Telugu	0.518	0.569	0.578	0.639
Ukrainian	0.607	0.642	0.682	0.821
Czech	0.611	0.626	0.663	0.805

Table 7: Average COMET₂₃^{DA,XL} scores for different languages for different components of the tasks. The **Questions** are used for both Misconception and Feedback tasks. The Tutoring task is not translated.

so the result is expected. Manual analysis of the low tutoring performance for Czech reveals that the interactions switch between various language formality styles, to the point that it becomes distracting. Additionally, the language used in Czech classrooms is particular and likely not represented on the internet.

Model performance and consistency. Mistral is the most inconsistent across non-English languages (average deviation⁵=0.186). For example, it completely fails the tutoring task for both Czech and Telugu, despite performing reasonably on other languages in the same task. Command-A is not much better (average deviation⁵=0.161). On the other hand, Gemini is the most consistent (average deviation⁵=0.078) and also has the second-best performance (average score 75.0%). GPT4o, is the best performing model (average 78.6%) while Claude performs the worst (average 49.3%) mostly due to Feedback and Tutoring tasks.

Task difficulty. The worst performance is observed in the Feedback task despite the similarity to the Misconception identification task. While Claude is still the standout worst performer with a worse-than-random performance, all models struggle. Further analysis in Table 11 shows that all models tended to default the feedback corresponding to the correct answer, with the models’ chain of thoughts being “regardless of the student’s mistake, this is the feedback that gives the student the most information about the correct answer.” Most models perform well in the Translation evaluation task, with the accuracy being even higher than human annotators, who were presented with attention checks with similar perturbations (Kocmi et al., 2024; Zouhar et al., 2025). They also do well in the Misconceptions task, with most percentage scores

⁵We calculate the standard deviation across the six languages for each task and then calculate the mean.

(at least in the English prompt setting) being in the 90s. The tutoring task seems to have the most inconsistent performance across models and languages. In general, all models struggle in Czech and Telugu, while Claude struggles in all languages. Avoiding telling seems to be the more challenging part of the problem for all the models, although success rates are not very consistent either.

English and translated prompts. Excluding for the Tutoring task (which did not use native prompts), using English prompts yields better performance than using translated prompts (averages 72.7% and 67.2%). The exceptions to these are GPT, Llama, Gemini, and Mistral in the translation task though in most cases, the difference is not very large. Note that some of the poor performance could be attributed to the prompts being translated and checked for correctness rather than being written in the target language directly, which could introduce some translationese. Regardless, we believe it is best to keep prompts in English. As a further note for English-speaking developers designing multilingual applications, keeping prompts in English ensures that the chains-of-thought remain English, making it easier to run sanity checks.

4 Related Work

LLMs, trained on vast multilingual texts, have dominated tasks such as text generation, translation, and dialogue (Brown et al., 2020), making them promising tools in Intelligent Tutoring Systems (ITS; Corbett et al., 1997; Pal Chowdhury et al., 2024). Prior work explores their use in educational contexts, such as dynamic student interactions (Schmucker et al., 2023), simulating expert and novice behavior (Liu et al., 2023), and math word problem reasoning (Opedal et al., 2023).

Beyond mathematical context, LLMs have also been explored for other forms of learning. Cui

and Sachan (2023) investigate LLMs in adaptive and personalized exercise generation for language learners, while (Wang et al., 2023) examines how conversational tutoring strategies can aid student understanding. Additionally, LLMs have been used to assess grammatical correctness and translation accuracy (Kocmi and Federmann, 2023; Omelianchuk et al., 2024; Freitag et al., 2024), facilitate automated essay scoring (Pack et al., 2024), and provide corrective feedback in second language writing (Han et al., 2024). While LLMs excel in English, their abilities in other languages often vary, reflecting an over-representation of high-resource languages in pre-training corpora. For example, Koto et al. (2023) introduces IndoMMLU, which reveals significant performance disparities between Indonesian and English contexts. Similarly, Holtermann et al. (2024) examines LLMs across 137 languages and attributes discrepancies in performance to tokenisation strategies. Li et al. (2024); Armengol-Estapé et al. (2022) further find a strong correlation between pre-training data proportions and performance, reaffirming the gap between high- and low-resource languages. For Catalan, Armengol-Estapé et al. (2022) find that while GPT-3 performed well in generative tasks, its comprehension capabilities were limited by the language’s moderate representation.

Recent research has increasingly explored the application of LLMs in multilingual educational contexts, though challenges persist in balancing performance across languages. Systematic reviews of AI-based language learning tools highlight the prevalence of NLP and machine learning techniques for error correction, feedback provision, and assessment in non-English contexts, though they note persistent gaps in dialogic competence and teacher preparedness (Alhusaiyan, 2025). Studies evaluating LLMs’ cross-lingual capabilities reveal performance disparities, with models demonstrating stronger skill tagging accuracy for English-centric curricula compared to underrepresented languages like Irish or Marathi (Kwak and Pardos, 2024). Bibliometric analyses indicate growing research interest in AI for foreign language education, particularly in vocabulary acquisition and writing support, though most studies still focus on high-resource European and Asian languages (Doğan and Talan). These works collectively underscore both the transformative potential and current limitations of LLMs in achieving equitable multilingual educational support.

To address multilingual education more directly, projects like Kaleidoscope (Salazar et al., 2025) and Aya (Üstün et al., 2024) by Cohere For AI aim to support culturally diverse languages, while SEA-HELM (Susanto et al., 2025) and ECLeKTic (Goldman et al., 2025) emphasise culturally grounded evaluations in Southeast Asian and cross-lingual contexts, respectively. These efforts highlight the need for multilingual benchmarks that move beyond English-centric evaluations.

Prior pedagogical studies tend to assess single LLMs in monolingual settings. We fill this gap by benchmarking LLMs in multiple tasks. Specifically, we conduct zero-shot experiments across multiple models and languages to better analyze their real-world applicability.

5 Conclusion

We analyse the performance of six well-known state-of-the-art LLMs across six languages other than English on four educational tasks. We find that while performance in English continues to be better than in other languages, the drop to other models is not always large. In particular, we find that GPT4o and Gemini 2.0 perform consistently well across all languages, with a few exceptions. We also note that English prompts work as well, if not better, than prompts written in the target language, when solving multilingual tasks. This opens up opportunities for porting applications developed for English into different languages. However, we note that certain models perform poorly in some tasks and languages, so **we recommend** first verifying that a model works well in a particular language on a specific educational task before deployment. However, to answer the question posed by the title, we believe that *atleast some* language models **are** reliable across languages.

Limitations

The shown experiments could naturally be better extended to more languages. The selected languages reflect a balance between author familiarity, which is necessary for meaningful qualitative analysis, and linguistic diversity, as evidenced by their spread in URIEL feature space. Similarly, we only covered six LLMs. In both cases, the cost of experiments (see Table 8) becomes prohibitively expensive, which motivated the data release in this paper to enable further research.

Additionally, translation quality remains a con-

Model	API	Total	Miconception	Feedback	Tutoring	Translation
Mistral	Mistral API	\$530	\$170	\$170	\$120	\$70
Claude	Anthropic	\$600	\$190	\$190	\$135	\$85
Command	Cohere	\$520	\$165	\$165	\$120	\$70
Llama	Together.ai	\$600	\$190	\$190	\$135	\$80
GPT4o	Open AI	\$80	\$25	\$25	\$18	\$12
Gemini	Google Genai	\$30	\$10	\$10	\$6	\$4

Table 8: Approximate costs for the experiments. Does not include taxes or currency conversion charges. The total is about \$2360 with approximately an additional \$500 spent on preliminary experiments.

cern, as previously discussed. A more thorough evaluation would involve human translations for every task, similar to the MMLU multilingual benchmark (Xuan et al., 2025), but doing so for all our tasks would be resource-intensive.

Finally, the set of tasks is not a complete representation of problems in the education space, primarily because most of the more complex tasks lack well-defined language-agnostic metrics.

Acknowledgements

Sankalan Pal Chowdhury is partially funded by the ETH-EPFL JDPLS Program. Donya Rooein is supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 949944, INTEGRATOR).

References

- Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. [MEGA: Multilingual evaluation of generative AI](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267. Association for Computational Linguistics.
- Sanchit Ahuja, Divyanshu Aggarwal, Varun Gumma, Ishaan Watts, Ashutosh Sathé, Millicent Ochieng, Rishav Hada, Prachi Jain, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2024. [MEGAVERSE: Benchmarking large language models across languages, modalities, models and tasks](#). 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 2598–2637. Association for Computational Linguistics.
- Mistral AI. 2024. [Large enough: Introducing mistral large 2](#). Accessed: 2024-09-08.
- Bashar Alhafni, Sowmya Vajjala, Stefano Bannò, Kaushal Kumar Maurya, and Ekaterina Kochmar. 2024. [LLMs in education: Novel perspectives, challenges, and opportunities](#). *Preprint*, arXiv:2409.11917.
- Eman Alhusaiyan. 2025. A systematic review of current trends in artificial intelligence in foreign language learning. *Saudi Journal of Language Studies*, 5(1):1–16.
- Abdullah M. Almasoud, Muhammad Rafay Naeem, Muhammad Imran Taj, Ibrahim Ghaznavi, and Junaid Qadir. 2025. [Toward inclusive educational AI: Auditing frontier LLMs through a multiplexity lens](#). *ArXiv*, abs/2501.03259.
- Tariq Alqahtani, H. Badreldin, Mohammed A. Alrashed, Abdulrahman I. Alshaya, S. Alghamdi, Khalid Bin saleh, Shuroug A. Alowais, Omar A. Alshaya, I. Rahman, Majed S Al Yami, and Abdulkareem M. Al-bekairy. 2023. [The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research](#). *Research in social & administrative pharmacy : RSAP*.
- Anthropic. 2024. [Introducing claude 3.5 sonnet](#). Accessed: 2024-09-08.
- Sabrina Argoub. 2022. [The NLP divide: English is not the only natural language - polis](#).
- Jordi Armengol-Estapé, Ona de Gibert Bonet, and Maite Melero. 2022. [On the multilingual capabilities of very large-scale English language models](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3056–3068. European Language Resources Association.
- 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.
- 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](#). *Preprint*, arXiv:2005.14165.

- Peter A. Cohen, James A. Kulik, and Chen-Lin C. Kulik. 1982. [Educational outcomes of tutoring: A meta-analysis of findings](#). *American Educational Research Journal*, 19(2):237–248.
- Team Cohere, Arash Ahmadian, Marwan Ahmed, Jay Alammam, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, Raphaël Avalos, and 1 others. 2025. [Command a: An enterprise-ready large language model](#). *arXiv preprint arXiv:2504.00698*.
- Albert T. Corbett, Kenneth R. Koedinger, and John R. Anderson. 1997. [Chapter 37 - intelligent tutoring systems](#). In Marting G. Helander, Thomas K. Landauer, and Prasad V. Prabhu, editors, *Handbook of Human-Computer Interaction (Second Edition)*, second edition, pages 849–874. North-Holland, Amsterdam.
- Peng Cui and Mrinmaya Sachan. 2023. [Adaptive and personalized exercise generation for online language learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10184–10198. Association for Computational Linguistics.
- Yunus Doğan and Tark Talan. [Artificial intelligence in foreign language learning: A bibliometric analysis](#). *Journal of Pedagogical Research*, 9(2):206–230.
- David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2025. [Ethnologue: Languages of the World](#), 28 edition. SIL International, Dallas, Texas. Online version: <http://www.ethnologue.com>.
- Wikimedia Foundation. 2024. [List of wikipeidias by language group](#). Accessed: 2024-09-08.
- Markus Freitag, Nitika Mathur, Daniel Deutsch, Chikiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. [Are LLMs breaking MT metrics? results of the WMT24 metrics shared task](#). In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81. Association for Computational Linguistics.
- Ilie Gligorea, Marius Cioca, Romana Oancea, A. Gorski, Hortensia Gorski, and Paul Tudorache. 2023. [Adaptive learning using artificial intelligence in e-learning: A literature review](#). *Education Sciences*.
- Omer Goldman, Uri Shaham, Dan Malkin, Sivan Eiger, Avinatan Hassidim, Yossi Matias, Joshua Maynez, Adi Mayrav Gilady, Jason Riesa, Shruti Rijhwani, Laura Rimell, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2025. [ECLeKTic: A novel challenge set for evaluation of cross-lingual knowledge transfer](#). *Preprint*, arXiv:2502.21228.
- 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.
- Albert Gu and Tri Dao. 2023. [Mamba: Linear-Time sequence modeling with selective state spaces](#). *Preprint*, arXiv:2312.00752.
- Jieun Han, Haneul Yoo, Junho Myung, Minsun Kim, Hyunseung Lim, Yoonsu Kim, Tak Yeon Lee, Hwajung Hong, Juho Kim, So-Yeon Ahn, and Alice Oh. 2024. [LLM-as-a-tutor in EFL writing education: Focusing on evaluation of student-LLM interaction](#). In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 284–293. Association for Computational Linguistics.
- Carolin Holtermann, Paul Röttger, Timm Dill, and Anne Lauscher. 2024. [Evaluating the elementary multilingual capabilities of large language models with multiq](#). *Preprint*, arXiv:2403.03814.
- Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. [Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting](#). *Preprint*, arXiv:2305.07004.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Enkelejda Kasneci, Kathrin Seifler, S. K uchemann, M. Bannert, Daryna Dementieva, F. Fischer, Urs Gasser, G. Groh, Stephan G nnemann, Eyke H llnermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, J. Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, T. Seidel, and 4 others. 2023. [ChatGPT for good? on opportunities and challenges of large language models for education](#). *Learning and Individual Differences*.
- Blanka Klimova, Marcel Pikhart, and Liqaa Habeb Al-Obaydi. 2024. [Exploring the potential of ChatGPT for foreign language education at the university level](#). *Frontiers in Psychology*, 15:1269319.
- Tom Kocmi and Christian Federmann. 2023. [Large language models are state-of-the-art evaluators of translation quality](#). In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203. European Association for Machine Translation.
- Tom Kocmi, Vil m Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popovi c, Mrinmaya Sachan, and Mariya Shmatova. 2024. [Error span annotation: A balanced approach](#)

- for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1440–1453. Association for Computational Linguistics.
- Fajri Koto, Nurul Aisyah, Haonan Li, and Timothy Baldwin. 2023. Large language models only pass primary school exams in indonesia: A comprehensive test on IndoMMLU. *Preprint*, arXiv:2310.04928.
- Yerin Kwak and Zachary A. Pardos. 2024. Bridging large language model disparities: Skill tagging of multilingual educational content. *British Journal of Educational Technology*, 55(5):2039–2057.
- Viet Dac Lai, Nghia Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13171–13189. Association for Computational Linguistics.
- Maikel Leon. 2024. Leveraging generative AI for on-demand tutoring as a new paradigm in education. *International Journal on Cybernetics & Informatics*.
- Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du. 2024. Quantifying multilingual performance of large language models across languages. *Preprint*, arXiv:2404.11553.
- Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14. Association for Computational Linguistics.
- 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. *Preprint*, arXiv:2310.02439.
- Itai Mondshine, Tzuf Paz-Argaman, Asaf Achi Mordechai, and Reut Tsarfaty. 2024. HeSum: a novel dataset for abstractive text summarization in Hebrew. In *Proceedings of the Seventh Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2024)*, pages 26–36. Association for Computational Linguistics.
- Maggie A. Mosher, Lisa Dieker, and Rebecca Hines. 2024. The past, present, and future use of artificial intelligence in teacher education. *Journal of Special Education Preparation*.
- Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. Automated evaluation of written discourse coherence using GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403. Association for Computational Linguistics.
- Nvidia. 2022. Transformer model.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanyskiy, 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.
- Andreas Opedal, Niklas Stoehr, Abulhair Saparov, and Mrinmaya Sachan. 2023. World models for math story problems. *Preprint*, arXiv:2306.04347.
- OpenAI. 2019. Language models are unsupervised multitask learners.
- OpenAI. 2023. GPT-4 technical report. *Preprint*, arXiv:2303.08774.
- 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.
- 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, L@S '24*, page 5–15, New York, NY, USA. Association for Computing Machinery.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.
- Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, and 15 others. 2023. RWKV: Reinventing RNNs for the transformer era. *Preprint*, arXiv:2305.13048.
- Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. CoEdIT: Text editing by task-specific instruction tuning. *Preprint*, arXiv:2305.09857.
- Ricardo Rei, Nuno M. Guerreiro, Jos textasciitilde A© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848. Association for Computational Linguistics.

- 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.
- 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. Association for Computational Linguistics.
- Konstantinos I. Roumeliotis, Nikolaos D. Tselikas, and Dimitrios K. Nasiopoulos. 2023. [Llama 2: Early adopters’ utilization of meta’s new open-source pre-trained model](#). *Preprints*.
- Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2022. [Square one bias in NLP: Towards a multi-dimensional exploration of the research manifold](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2340–2354. Association for Computational Linguistics.
- Israfael Salazar, Manuel Fernández Burda, Shayekh Bin Islam, Arshia Soltani Moakhar, Shivalika Singh, Fabian Farestam, Angelika Romanou, Danylo Boiko, Dipika Khullar, Mike Zhang, Dominik Krzemiński, Jekaterina Novikova, Luísa Shimabucoro, Joseph Marvin Imperial, Rishabh Maheshwary, Sharad Duwal, Alfonso Amayuelas, Swati Rajwal, Jebish Purbey, and 25 others. 2025. [Kaleidoscope: In-language exams for massively multilingual vision evaluation](#). *Preprint*, arXiv:2504.07072.
- Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2023. [Ruffle&riley: Towards the automated induction of conversational tutoring systems](#). *Preprint*, arXiv:2310.01420.
- Burr Settles. 2018. [Data for the 2018 duolingo shared task on second language acquisition modeling \(SLAM\)](#).
- Yosephine Susanto, Adithya Venkatadri Hulagadri, Jann Railey Montalan, Jian Gang Ngui, Xian Bin Yong, Weiqi Leong, Hamsawardhini Rengaran, Peerat Limkonchotiwat, Yifan Mai, and William Chandra Tjhi. 2025. [SEA-HELM: South-east asian holistic evaluation of language models](#). *Preprint*, arXiv:2502.14301.
- Gemini Team. 2024. [Gemini: A family of highly capable multimodal models](#). *Preprint*, arXiv:2312.11805.
- Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D’souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. [Aya model: An instruction fine-tuned open-access multilingual language model](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939. Association for Computational Linguistics.
- Lingzhi Wang, Mrinmaya Sachan, Xingshan Zeng, and Kam-Fai Wong. 2023. [Strategize before teaching: A conversational tutoring system with pedagogy self-distillation](#). In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2268–2274. Association for Computational Linguistics.
- Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Yun Xing, Junjue Wang, Huitao Li, Xin Li, Kunyu Yu, Nan Liu, Qingyu Chen, Douglas Teodoro, Edison Marrese-Taylor, Shijian Lu, Yusuke Iwasawa, Yutaka Matsuo, and Irene Li. 2025. [MMLU-ProX: A multilingual benchmark for advanced large language model evaluation](#). *Preprint*, arXiv:2503.10497.
- Tiffany Zhu, Kexun Zhang, and William Yang Wang. 2024. [Embracing AI in education: Understanding the surge in large language model use by secondary students](#). *Preprint*, arXiv:2411.18708.
- Vilém Zouhar, Tom Kocmi, and Mrinmaya Sachan. 2025. [AI-assisted human evaluation of machine translation](#). *Preprint*, arXiv:2406.12419.

Language	English prompt						Translated prompt					
	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A
English	0.0%	0.7%	0.1%	2.1%	0.0%	0.9%	0.0%	0.7%	0.1%	2.1%	0.0%	0.9%
Mandarin	0.5%	1.6%	0.0%	3.2%	0.0%	0.2%	0.5%	1.6%	0.0%	3.2%	0.0%	0.2%
Hindi	0.0%	1.6%	0.4%	2.3%	0.0%	0.4%	0.0%	0.9%	0.2%	2.5%	0.0%	0.1%
Arabic	0.1%	1.7%	0.2%	2.1%	0.0%	0.2%	0.0%	1.1%	0.3%	2.2%	0.0%	0.1%
German	0.5%	1.6%	0.3%	2.3%	0.0%	0.2%	0.5%	1.6%	0.3%	2.3%	0.0%	0.2%
Farsi	0.0%	1.8%	0.2%	2.0%	0.0%	0.3%	0.0%	1.6%	0.2%	2.9%	0.0%	0.1%
Telugu	0.0%	0.1%	0.0%	2.2%	0.0%	0.4%	0.0%	0.3%	0.1%	1.7%	0.0%	0.0%
Ukranian	0.1%	1.6%	0.1%	2.2%	0.0%	0.3%	0.0%	1.8%	0.4%	1.6%	0.0%	0.1%
Czech	0.1%	1.6%	0.1%	1.9%	0.0%	0.7%	0.0%	1.4%	0.0%	0.7%	0.0%	0.5%

Table 9: Response error rate for the **misconception identification** task.

Language	English prompt						Translated prompt					
	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A
English	0.0%	0.3%	0.0%	1.3%	0.0%	0.0%	0.0%	0.3%	0.0%	1.3%	0.0%	0.0%
Mandarin	0.0%	0.1%	0.0%	1.5%	0.0%	0.2%	0.0%	0.0%	0.1%	1.6%	0.0%	0.1%
Hindi	0.0%	0.0%	0.0%	1.1%	0.0%	0.1%	0.0%	0.0%	0.0%	1.0%	0.0%	0.2%
Arabic	0.0%	0.0%	0.0%	1.5%	0.0%	0.1%	0.0%	0.0%	0.0%	2.1%	0.0%	0.2%
German	0.0%	0.0%	0.0%	1.1%	0.0%	0.1%	0.0%	0.0%	0.0%	0.8%	0.0%	0.2%
Farsi	0.0%	0.0%	0.0%	1.2%	0.0%	0.0%	0.0%	0.0%	0.0%	1.1%	0.0%	0.1%
Telugu	0.0%	0.0%	0.0%	1.7%	0.0%	0.1%	0.0%	0.2%	0.0%	1.8%	0.0%	0.1%
Ukranian	0.0%	0.0%	0.0%	0.9%	0.0%	0.0%	0.0%	0.0%	0.0%	1.3%	0.0%	0.0%
Czech	0.0%	0.0%	0.0%	1.1%	0.0%	0.0%	0.0%	0.0%	0.0%	3.0%	0.0%	0.0%

Table 10: Response error rate for the **feedback selection** task.

Language	English prompt						Translated prompt					
	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A
English	23.7%	45.8%	75.0%	27.8%	23.0%	35.1%	23.7%	45.8%	75.0%	27.8%	23.0%	35.1%
Mandarin	26.9%	54.5%	81.5%	36.9%	33.8%	47.7%	32.6%	71.9%	89.7%	24.3%	49.3%	54.5%
Hindi	30.3%	42.3%	79.6%	34.9%	29.9%	47.9%	55.5%	78.9%	87.7%	33.2%	68.9%	71.3%
Arabic	28.0%	54.1%	79.5%	35.3%	32.9%	44.0%	21.9%	81.7%	73.6%	22.4%	36.4%	49.5%
German	25.1%	48.8%	79.4%	32.7%	30.4%	45.4%	22.3%	54.5%	77.0%	29.6%	32.9%	36.0%
Farsi	28.5%	52.5%	82.5%	31.6%	32.6%	45.5%	21.6%	52.1%	75.1%	29.3%	30.3%	28.9%
Telugu	29.2%	55.6%	81.5%	35.2%	33.1%	52.4%	78.3%	73.8%	89.5%	37.9%	70.9%	78.8%
Ukranian	27.3%	49.4%	80.3%	32.7%	33.5%	45.2%	47.3%	69.7%	87.1%	20.7%	46.7%	49.7%
Czech	27.9%	39.5%	80.2%	30.2%	31.8%	49.0%	34.9%	59.5%	67.8%	23.0%	33.1%	38.0%

Table 11: Rate of defaulting to the correct answer for the **feedback selection** task.

Language	English prompt						Translated prompt					
	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A	GPT4o	LLama	Claude	Gemini	Mistral	Cmd-A
Mandarin	0.0%	0.1%	0.5%	0.0%	0.0%	0.0%	0.0%	0.1%	4.2%	0.0%	0.0%	0.0%
Hindi	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%
Arabic	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	3.7%	0.0%	0.0%	0.0%
German	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	3.5%	0.0%	0.0%	0.0%
Farsi	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Telugu	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.5%	0.0%	0.0%	0.0%
Ukranian	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.5%	0.0%	0.0%	0.0%
Czech	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.8%	0.0%	0.0%	0.0%

Table 12: Response error rate for the **translation grading** task.

A Experiment Prompts

A.1 Task: Misconception Identification

We used a sequence of 3 prompts:

System prompt:

You are an expert math tutor who knows about all grade-school level math misconceptions. Your task is to select the accurate type of misconceptions your student has based on the (incorrect) answer he/she gives to a multiple-choice math question. You will be given 4 misconceptions types. Your selected misconception type should correspond to the given question and answer. Explain your reasoning

User message 1:

Question: {QUESTION}
Selected Answer: {SELECTED_ANSWER}
Misconceptions:
A. {Misconception 1}
B. {Misconception 2}
C. {Misconception 3}
D. {Misconception 4}

The position of the Misconception corresponding to the selected answer rotates from question to question. The subsequent assistant message is stored as the chain-of-thought. Thereafter, we sent the second user message.

User message 2:

Now based on your above explanation, output the option corresponding to the correct misconception. Only say 'A', 'B', 'C', or 'D' without any other text. Do not say anything else.

The response to this part is the final answer. We regenerate until an answer of 'A', 'B', 'C', or 'D' is received, up to 20 times. If no answer is received, a response of 'E' is saved.

This method is used for all models except Gemini. In case of Gemini, we use the `generate_content` method, which is recommended for non-chat tasks and allows for a single user message. In this case, after obtaining the chain-of-thought, we make a new query with the same system prompt but with the following user message:

Gemini message:

You have previously given the following answer and explanation:
{COT}
Now based on your above explanation, output the option corresponding to the correct misconception. Only say 'A', 'B', 'C', or 'D' without any other text. Do not say anything else.

Note that the last part is identical to User Message 2

When using translated prompts, the System Prompt and, User Message 2 and Gemini Message are translated to the target language.

A.2 Task: Feedback Selection

System prompt:

You are an expert math tutor who specialises in providing precise and helpful feedback for grade-school level math questions. Your task is to select the correct explanation for a student's given answer to a multiple-choice math question.

You will be provided with:

- A math question
- A specific answer chosen by the student (which can be correct or incorrect).
- Four possible explanations (labelled A, B, C, and D).

Your selected explanation should accurately correspond to the given answer. Provide your reasoning for selecting the explanation.

User message 1:

Question: {QUESTION}

Selected Answer: {SELECTED_ANSWER}

Feedbacks:

- A. {Feedback 1}
- B. {Feedback 2}
- C. {Feedback 3}
- D. {Feedback 4}

The position of the Feedback corresponding to the selected answer rotates from question to question. If it is placed at positions A, B, or C, the feedback corresponding to the correct answer is at position D. Otherwise, it is at C. The subsequent assistant message is stored as the chain-of-thought. Thereafter, we sent the second user message.

User message 2:

Now based on your above explanation, output the option corresponding to the correct explanation. Only say 'A', 'B', 'C', or 'D' without any other text. Do not say anything else.

The response to this part is the final answer. We regenerate until an answer of 'A', 'B', 'C', or 'D' is received, up to 20 times. If no answer is received, a response of 'E' is saved.

This method is used for all models except Gemini. In case of Gemini, we use the `generate_content` method, which is recommended for non-chat tasks and allows for a single user message. In this case, after obtaining the chain-of-thought, we make a new query with the same system prompt but with the following user message:

Gemini message:

You have previously given the following answer and explanation:

{COT}

Now based on your above explanation, output the option corresponding to the correct explanation. Only say 'A', 'B', 'C', or 'D' without any other text. Do not say anything else.

Note that the last part is identical to User Message 2. When using translated prompts, the System Prompt, User Message 2, and Gemini Message are translated to the target language. We manually made sure that the formatting was maintained after the translation.

A.3 Task: Tutoring

Student system prompt:

Student Persona: {STUDENT_PERSONA}

Math problem: {MATH_PROBLEM}

Student solution: {STUDENT_SOLUTION}

Context: You need to role-play the student, {STUDENT_NAME}, while the user roleplays the tutor. {STUDENT_NAME} thinks their answer is correct. Only when the teacher provides several good reasoning questions, {STUDENT_NAME} understands the problem and corrects the solution. {STUDENT_NAME} can use calculator and thus makes no calculation errors. Send <EOM> tag at end of the student message.

Teacher system prompt:

A tutor and a student work together to solve the following math word problem.

Math problem: {MATH_PROBLEM}

The correct solution is as follows:

{GROUND_TRUTH}

You need to role-play the tutor while the user roleplays the student, {STUDENT_NAME}. The tutor is a soft-spoken empathetic man who dislikes giving out direct answers to students, and instead likes to answer questions with other questions that would help the student understand the concepts, so that she can solve the problem themselves.

{STUDENT_NAME} has come up with a solution, but it is incorrect. Please start the conversation, one line at a time, aiming to figure out what is {STUDENT_NAME}'s solution and what is wrong with it. Then try to get her to fix it.

The dialogue history was formatted as user-assistant message pairs for teacher and student roles. We manually set the initial messages to initiate conversations in the target language.

A.4 Task: Translation Grading

System prompt:

You are a language translation evaluator. Your task is to assess the quality of a translation from English to {LANGUAGE}. You will be provided with two sentences:

1. An original English sentence.
2. A translated sentence in {LANGUAGE}.

Your goal is to rate the translation on a scale from 1 to 5 based on the following criteria:

- 1: The translation is incorrect, incomprehensible, or completely unrelated to the original English sentence.
- 2: The translation has significant errors and distorts the meaning of the original English sentence.
- 3: The translation is understandable but contains notable errors or awkward phrasing.
- 4: The translation is mostly accurate with minor errors or slightly awkward phrasing.
- 5: The translation is fluent, natural, and accurately conveys the meaning of the original English sentence without errors.

Explain your decision

User message 1:

English: {ENGLISH_SENTENCE}
{LANGUAGE}: {TRANSLATED_SENTENCE}

The subsequent assistant message is stored as the chain-of-thought. Thereafter, we sent the second user message.

User message 2:

Now based on your above explanation, output the final score from 1 to 5. Only say '1', '2', '3', '4', or '5' without any other text. Do not say anything else.

The response to this part is the final answer. We regenerate until an answer of '1', '2', '3', '4', or '5' is received, up to 20 times. If no answer is received, a response of '0' is saved.

This method is used for all models except Gemini. In case of Gemini, we use the `generate_content` method, which is recommended for non-chat tasks and allows for a single user message. In this case, after obtaining the chain-of-thought, we make a new query with the same system prompt but with the following user message:

Gemini message:

You have previously given the following answer and explanation:

{COT}

Now based on your above explanation, output the final score from 1 to 5. Only say '1', '2', '3', '4', or '5' without any other text. Do not say anything else.

Note that the last part is identical to User Message 2

This sequence is repeated twice for each sentence, once with the original translation and once with the perturbed translation. The scores are then compared. When using English prompts, the LANGUAGE fields are set to their English exonyms, i.e., Mandarin, Hindi, Arabic, German, Farsi, Telugu, Ukrainian, and Czech. When using translated prompts, the System Prompt, User Message 2, and Gemini Message are translated to the target language. We manually made sure that the formatting was maintained after the translation. We also use the language endonyms, namely 中文, हिन्दी, العربية, Deutsch, فارسی, తెలుగు, Українська, and Čeština.

B Translation Quality

As we mentioned in Limitations, an LLM performing poorly in a given language does not necessarily mean that the LLM itself is bad. It could also mean that information was lost during translation. This is particularly problematic because the machine translation systems likely suffer from the same resource limitations that plague the LLMs in the first place. As such, we manually investigated a small subset of translated questions for the languages they we are fluent in, namely Persian, Arabic, Czech, and Hindi. For each language, we analysed 10 questions each for the Feedback and Misconception tasks, and 20 questions for the Translation Grading task.

In the case of Persian, the only recurring error was with mathematical notation, particularly that the minus sign gets placed to the right of the numbers instead of the left, where it should be. This, however, seems to be a rendering issue, which is a result of the fact that the minus sign (‘−’, U+2212) is often replaced by the similar-looking hyphen (‘-’, U+002D), confusing the rendering program into believing that it is rendering text. This should not be an issue since LLMs take raw Unicode encodings as input. Beyond this, there were some minor tense errors, but the meanings were clear.

The issue with sign placement was also observed in Arabic. In addition, there seem to be some translation errors. For example, the word ‘travel’ used here in the context of the movement of a graph was translated to ‘liyusaafir’, which is more like ‘taking a trip’. We found no errors in the sentences for the translation task. In Czech, the primary source of errors was improper context-dependent terminology. For example, when translating the word ‘co-interior (angles)’, it missed the ‘co’ prefix and translated only the ‘interior’ part. While this is fine in regular speech, in Mathematical terminology, this can be confusing. Despite making the translation harder to follow, the core meaning of the question is preserved.

In Hindi we found several cases where the Hindi sentence was difficult to follow for the Hindi speaking author due to misinterpretation of polysemes by the translator e.g. the word ‘round’, which was being used in the sense of ‘approximate’ was translated to the sense of ‘circle’ and ‘property’ which was being used in the sense of ‘quality’, was translated as ‘possessions’. Also, the phrase ‘Not Quite’ was translated to something like ‘Not Enough’, perhaps due to the word ‘quite’ not having a Hindi equivalent. However, given the context, using the word for ‘Almost’ would have been more tonally accurate. However, quite a few translations were hard for the annotator to follow, but backtranslating them yielded reasonably good results, meaning there was no information loss.

The translation exercises showed few errors, perhaps due to the sentences being easy to translate by design. There were one or two mistranslations, but otherwise it worked well. One minor issue was that word boundary detection, which was performed in Python using the regex ‘\b\w+\b’, sometimes identified individual characters in Hindi rather than whole words. However, the resulting sentence still had errors, just not the type of errors that we expected.

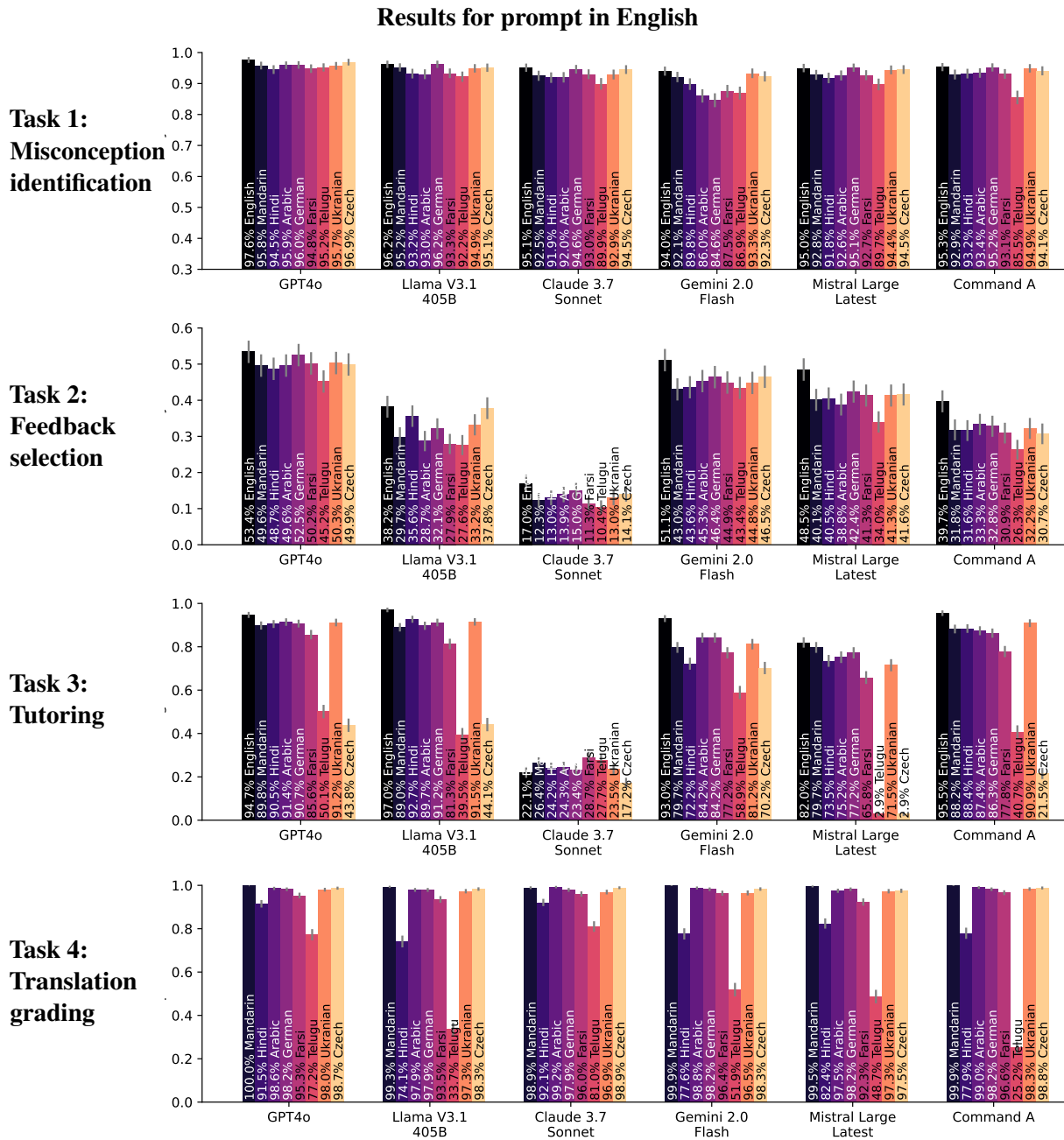


Figure 2: Evaluation results of the four tasks across five large language models. The error bars show a 95% confidence interval (t-test). MathDial Graphs show *tutoring score after five turns*, most models flatline after 5 utterance pairs. The English language column is absent because translation evaluation uses English as the source. All scores range from 0.0 to 1.0, with higher being better, though they are not comparable with each other. Note the truncated y-axes for better detail. Visualizes Tables 3 to 6.

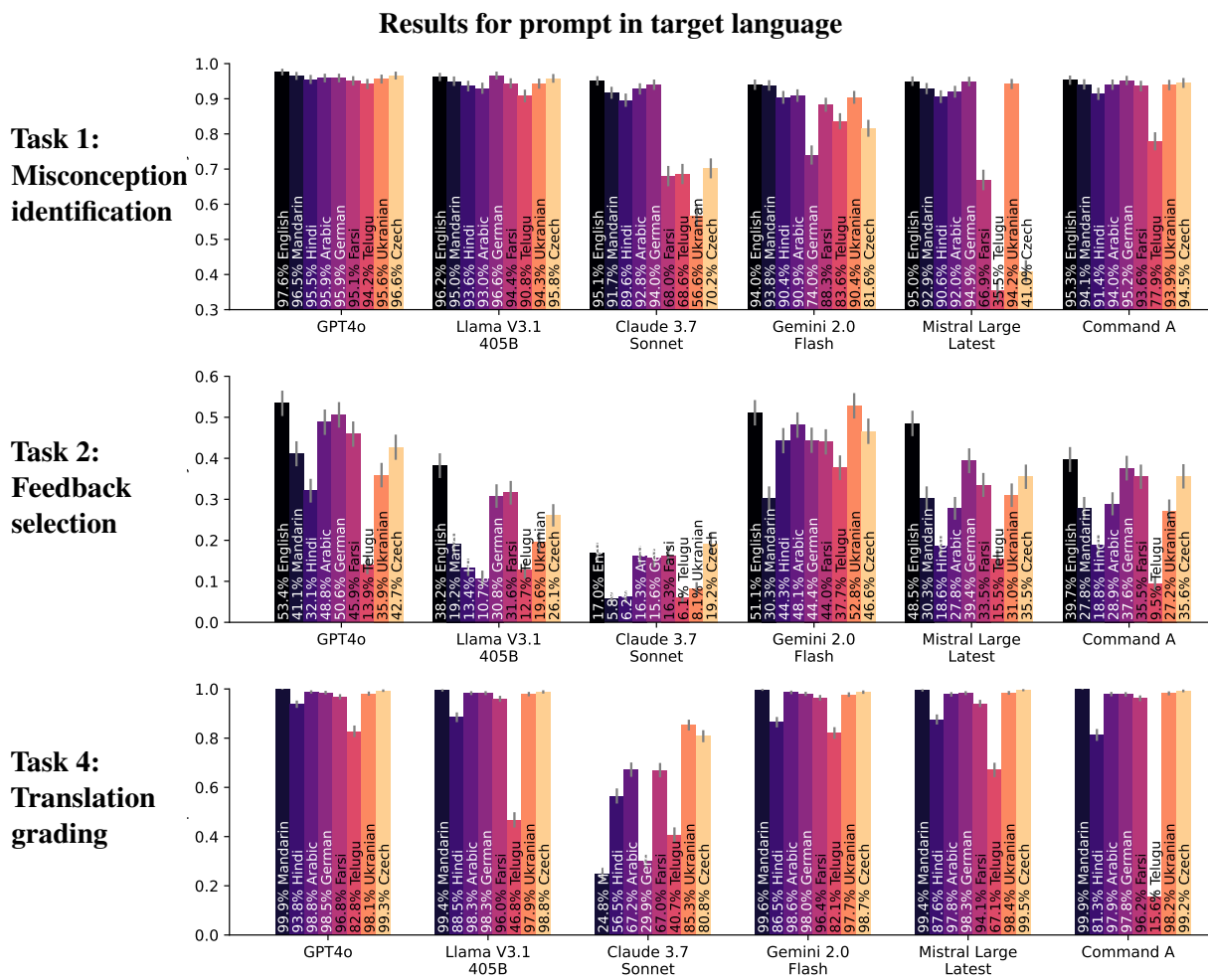


Figure 3: Evaluation results of the four tasks across five large language models. The error bars show 95% confidence interval (t-test). MathDial Graphs show *tutoring score after five turns*, most models flatline after 5 utterance pairs. The English language column is absent because translation evaluation uses English as the source. All scores range from 0.0 to 1.0, with higher being better, though they are not comparable with each other. Note the truncated y-axes for better detail. Visualizes Tables 3 to 6.

Exploiting the English Vocabulary Profile for L2 word-level vocabulary assessment with LLMs

Stefano Bannò, Kate M. Knill, Mark J. F. Gales

ALTA Institute, Department of Engineering, University of Cambridge (UK)

{sb2549, kmk1001, mjfg100}@cam.ac.uk

Abstract

Vocabulary use is a fundamental aspect of second language (L2) proficiency. To date, its assessment by automated systems has typically examined the context-independent, or part-of-speech (PoS) related use of words. This paper introduces a novel approach to enable fine-grained vocabulary evaluation exploiting the precise use of words within a sentence. The scheme combines large language models (LLMs) with the English Vocabulary Profile (EVP). The EVP is a standard lexical resource that enables in-context vocabulary use to be linked with proficiency level. We evaluate the ability of LLMs to assign proficiency levels to individual words as they appear in L2 learner writing, addressing key challenges such as polysemy, contextual variation, and multi-word expressions. We compare LLMs to a PoS-based baseline. LLMs appear to exploit additional semantic information that yields improved performance. We also explore correlations between word-level proficiency and essay-level proficiency. Finally, the approach is applied to examine the consistency of the EVP proficiency levels. Results show that LLMs are well-suited for the task of vocabulary assessment.

1 Introduction

Automated writing evaluation has become an established area of research within natural language processing, playing a key role in language education and computer-assisted language learning (Huawei and Aryadoust, 2023). Within this context, assessing second language (L2) proficiency remains a critical objective, with increasing attention being paid not only to holistic assessment but also to the evaluation of individual aspects of language competence (Weigle, 2002), such as grammar, coherence, and vocabulary. This has proven essential for providing detailed, pedagogically useful feedback to language learners (Hamp-Lyons, 1995). Among these components, vocabulary is the es-

sential building block of language (Schmitt et al., 2001) as words are the main vehicle for expressing meaning (Vermeer, 2001).

Existing vocabulary assessment methods have typically either assigned an overall vocabulary score to learners' texts at the essay- (Crossley et al., 2023; Bannò et al., 2024) or sentence-level (Arase et al., 2022), extracted vocabulary-related features (e.g., lexical diversity or sophistication) (Kyle and Crossley, 2015; Kyle et al., 2018), or attempted to assess the difficulty or appropriateness of individual words (Bax, 2012; Uchida and Negishi, 2018; Settles et al., 2020; Aleksandrova and Pouliot, 2023). The latter, however, has been largely under-explored and presents significant challenges – particularly in dealing with polysemy and contextual variation as well as handling multi-word expressions.

Our work directly addresses this limitation by focusing on word-level, in-context vocabulary assessment in a fully replicable manner. Specifically, we leverage an open-access reference containing information about which lexical items are used at each level of English learning, the English Vocabulary Profile (see Section 3.1). We combine this resource with state-of-the-art large language models (LLMs) to predict the proficiency level of individual words as they are used in L2 learner writing. To the best of our knowledge, this is the first study to apply LLMs for this task, offering a novel and robust approach to vocabulary assessment that explicitly handles semantic ambiguity and contextual nuance.

In Section 2, we outline the theoretical background of L2 vocabulary assessment and review previous studies that have implemented it in automated systems. Section 3 presents the English Vocabulary Profile and the L2 datasets used in our experiments. Section 4 constitutes the core of our study. The first experiment focuses on identifying the intended meaning of polysemous words from the English Vocabulary Profile through the use of

LLMs, using learner example sentences sourced from the same dataset. In our second experiment, we annotate sentences from a learner corpus by assigning each word a proficiency level based on the English Vocabulary Profile – annotations we plan to make publicly available. We then automatically predict these levels and compare the performance of various LLMs against a random baseline and a part-of-speech (PoS)-based model. The third experiment extends this approach to additional L2 learner datasets annotated only at the essay level, investigating the correlation between predicted word-level proficiency and essay-level proficiency. Additionally, at the end of the same section, we use our LLM-based approach to examine the consistency of the annotations in the English Vocabulary Profile. Finally, in Section 5, we present our conclusions and outline directions for future work.

2 Related Work

2.1 Vocabulary assessment

Despite its fundamental importance, the assessment of vocabulary was only selectively investigated at the beginning of the scientific era of L2 assessment, whereas much more attention was paid to the contrastive analysis of sounds and grammar (Lennon, 2008). When vocabulary knowledge was evaluated, it was primarily tested using the discrete-point approach, an assessment method focused on testing one specific linguistic element – phonology, morphology, syntax, and vocabulary – at a time, generally using multiple-choice questions. This approach to vocabulary testing faced various criticisms, as it offered only a limited view of a learner’s vocabulary knowledge, neglected the role of productive language use, disregarded the importance of context in real-world communication, and failed to consider learners’ use of strategies to cope with unfamiliar words (Read, 2000).

The 1980s represented a watershed in vocabulary assessment since a group of researchers started to publish studies on defined procedures aiming at assessing specific aspects of vocabulary use and knowledge (Anderson and Freebody, 1981, 1983; Nation, 1983; Meara and Buxton, 1987). These seminal works were something of an exception, given that, on the one hand, the field of L2 acquisition was primarily concerned with the investigation of the acquisition by learners of morpho-syntactic features, whereas, on the other hand, the advent of the communicative approach shifted the attention

of language assessment researchers from knowledge of grammatical and lexical elements to the performance of real-world-like tasks (Read, 2013).

For our work, we believe it is important to remember Read’s conceptualisation of vocabulary assessment, who classified it according to 6 dimensions arranged in antonymic pairs: discrete versus embedded, selective versus comprehensive, and context-independent versus context-dependent. The first distinguishes whether vocabulary is assessed as an isolated skill (discrete) or as part of broader language proficiency (embedded). The second refers to the scope of lexical items – either a specific set (selective) or the learner’s full vocabulary range (comprehensive). The third dimension captures whether vocabulary is assessed in isolation or within authentic contexts. Due to the widespread acceptance of the communicative approach (Harding, 2014), it is straightforward to conclude that current trends in language testing and assessment tend to privilege *embedded*, *comprehensive*, and *context-dependent* measures of vocabulary assessment. These three characteristics are central to our approach.

2.2 Lexical sophistication

The Common European Framework of Reference (CEFR) (Council of Europe, 2001, 2020), a key benchmark aligned with communicative language teaching, testing and assessment, distinguishes between vocabulary range and vocabulary control, which have generally been operationalised along the dimensions of lexical diversity and lexical sophistication, respectively.

Lexical diversity, concerning the breadth of vocabulary used by learners (Yu, 2010; Lu, 2012), is typically measured through metrics like type-token ratio or number of unique words. Its relationship with L2 writing proficiency has been widely studied (Crossley and McNamara, 2012; Gebril and Plakans, 2016; Treffers-Daller et al., 2018; Woods et al., 2023). While important, lexical diversity is not the primary focus of this paper.

Our work is more closely related to the idea of lexical sophistication. Its focus is the depth of lexical knowledge and is frequently characterised by the presence of relatively rare or uncommon words within a given language sample (Baese-Berk et al., 2021). It is generally operationalised using features related to word frequency and familiarity, such as the Lexical Frequency Profile, which reflects the proportion of a learner’s vocabulary falling within

various frequency bands derived from a reference corpus (Laufer and Nation, 1995). The English Vocabulary Profile (EVP) (Capel, 2015), adopted in our work (see Section 3), is a resource that describes words, phrases, idioms, and collocations used by English learners at different CEFR levels. The study by Leńko-Szymańska (2015), which represents an important precedent for our work, employed it to assign proficiency bands to 90 essays, finding a strong correlation between the clusters obtained using the vocabulary profile and the human-assigned CEFR levels.

Similarly to the EVP, Dürlich and François (2018) created an online database called EFLLex, which presents the distribution of English words across CEFR levels (A1 to C1), mainly derived from the analysis of textbook corpora. The CEFR-J, a Japan-specific adaptation of the CEFR, also features a word list with proficiency levels (Tono, 2013).

2.3 Automated approaches

Yoon et al. (2012) investigated the use of a vocabulary profile to extract features of lexical sophistication for proficiency assessment of spontaneous speech and found interesting correlations with oral proficiency scores. Kyle and Crossley (2015) introduced the Tool for Automatic Analysis of Lexical Sophistication (TAALES), which computes 135 lexical indices. They found that 5 measures of lexical complexity accounted for more than 50% of the variance in the human ratings of the spoken and written datasets considered in their study. Text Inspector (Bax, 2012) is an online tool that appears to use the EVP to assess writing proficiency; however, its implementation is not publicly available and is most likely rule-based, as it presents all possible meanings (and corresponding proficiency levels) of a word to the user when ambiguities arise. The calculation method used by the CVLA (CEFR-based Vocabulary Level Analyzer) is openly available (Uchida and Negishi, 2018); however, the strictly vocabulary-related part of this tool is also rule-based and, like Text Inspector, does not appear to address issues related to polysemous or ambiguous words.

Duolingo developed and released a tool called the CEFR checker, which is now discontinued. This tool allowed users to assess the difficulty level of English and Spanish words and texts. The lexical complexity component of the tool is described in Settles et al. (2020) and features a vocabulary

scale model based on the CEFR framework and a database of 6,823 English words, partly obtained from the EVP. The authors introduce two regression models trained on lexical representations using surface-level features designed to approximate word frequency. However, these models do not appear to account for multi-word expressions or words with multiple meanings.

Garí Soler and Apidianaki (2021) demonstrated that BERT could effectively generate contextual embeddings for polysemous words, laying the groundwork for further research in lexical complexity assessment. Building on this, Aleksandrova and Pouliot (2023) explored how the most frequent senses of polysemous words appear in language learner essays. Their study leverages BERT to develop a CEFR-aligned classifier aimed at evaluating the lexical complexity of both single-word and multi-word expressions in English and French. It is important to note that their classifier is trained to predict the CEFR level of an item in context, but not to explicitly identify or disambiguate the meaning or sense of that expression.¹

While the aforementioned studies partially or entirely fall short in handling polysemy and ambiguity, our study directly addresses this challenge by targeting word-level, in-context vocabulary assessment in a fully replicable manner. Additionally, to the best of our knowledge, ours is the first study to leverage LLMs for this purpose.

3 Data

3.1 English Vocabulary Profile

The English Vocabulary Profile (EVP) (Capel, 2015) is a publicly available reference² that contains information about which words, phrases, idioms, and collocations are used at each level of English learning. It is grounded in extensive research using the Cambridge Learner Corpus (Nicholls, 2003), a growing collection of exam scripts written by learners worldwide.

For our experiments, we only considered the British English section of the profile, which includes 15671 entries corresponding to 6747 unique words. The difference arises because some words have multiple meanings or grammatical functions, resulting in several entries comprising multiple op-

¹Other works have targeted word-level vocabulary assessment, but not for English (Gala et al., 2014; Alfter et al., 2016; Alfter and Volodina, 2018).

²englishprofile.org/

tions. Additional details about the degree of polysemy are provided in Table 6 in Appendix A.

Base Word	Guideword	Level	Part of Speech	Topic	Details
if/when push comes to shove	IDIOM	C2	phrase		
push	MOVE SOMEONE/SOME THING	A2	verb	people: actions	
push	PRESS	B1	verb	people: actions	
push	MOVE YOURSELF	B1	verb	people: actions	
push	ENCOURAGE	C1	verb	people: actions	
push	ENCOURAGEMENT	C1	noun		
push	PRESS	B1	noun	people: actions	
push (sb) for sth/to do sth		B2	phrase	communication	
pushy		C2	adjective	people: personality	
push yourself		B2	phrase	people: actions	

Figure 1: Example for the word *push* from the EVP.

Each EVP entry includes a base word, a guideword, its CEFR level(s), manually assigned PoS, topic, and additional details such as learner and dictionary examples in context (see Figure 1). Among all entries, when distinguishing by PoS, 62.24% have more than one CEFR level. However, when considering only unique base words, this proportion drops to 29.21%. In this work, we refer to words that, within a single PoS, have multiple CEFR levels as *ambiguous* words; the rest are considered *non-ambiguous*. For example, the noun *aim* is classified solely as B1, making it non-ambiguous, whereas the verb *aim* spans levels A2, B2, C1, and C2 depending on context, making it ambiguous. About 95% of the section of the EVP we considered in our work has examples taken from L2 learner writing, many of which also contain grammatical errors. For the remaining 5%, we relied on dictionary examples in our experiments.

3.2 L2 learner datasets

3.2.1 OneStopEnglish

OneStopEnglish (Vajjala and Lučić, 2018), is a publicly available corpus³ for readability assessment and text simplification including 189 parallel compositions across three readability levels: Elementary, Intermediate, and Advanced. We extracted 293 triplets of parallel sentences from the corpus, spanning these three readability levels (see Appendix B for an example), and annotated them at the word level with CEFR levels from the EVP, excluding stopwords,⁴ punctuation, and words not

³github.com/nishkalavallabhi/OneStopEnglishCorpus

⁴Stopwords include those from the NLTK list ([nltk.org](https://www.nltk.org)) plus *across*, *among*, and *away*.

featured in the EVP (see Figure 2). These annotations will be made publicly available.

3.2.2 EFCAMDAT

Arguably the largest publicly available L2 learner corpus,⁵ the second release of the EFCAMDAT (Geertzen et al., 2013; Huang et al., 2017) consists of 1,180,310 scripts written by 174,743 L2 learners as assignments for Englishtown, an online English language school. The corpus includes 128 distinct writing tasks covering a range of topics, such as describing the rules of a game, reporting a news story, explaining a homemade remedy for fever, and writing to a pen pal. Each composition is annotated with a proficiency level from 1 to 16, corresponding to CEFR levels A1 to C2.⁶ Learners' first languages are not directly available but can be inferred from their nationalities (approximately 200 in total). For our experiments, we randomly selected 1,000 essays for each CEFR level from the corpus, resulting in a total of 6,000 essays.

3.2.3 ELLIPSE

The English Language Learner Insight, Proficiency, and Skills Evaluation (ELLIPSE) Corpus (Crossley et al., 2023) is a publicly available collection⁷ of approximately 6,500 writing samples from L2 learners of English. Each sample is annotated with both an overall holistic proficiency score and detailed analytic scores covering aspects such as cohesion, syntax, vocabulary, phraseology, grammar, and punctuation conventions. Proficiency scores are on scale from 1 to 5. For our experiments, we extracted 359 essays from the training set and 175 from the test set. Further details about the selected essays can be found in Table 7 in Appendix A.

4 Experiments

4.1 Semantic understanding

Our first experiment serves as a proof of concept and focuses on semantic understanding, with the goal of identifying the intended meaning of a unique word from the EVP when considering the EVP learner examples. Specifically, an LLM is provided with an EVP learner-produced example sentence containing a given word along with

⁵ef-lab.mm11.cam.ac.uk/EFCAMDAT.html

⁶The official EF-CEFR mapping can be found at: myenglishlive.ef.com/help-me-article?articleID=55&Vote=Up

⁷github.com/scrosseye/ELLIPSE-Corpus

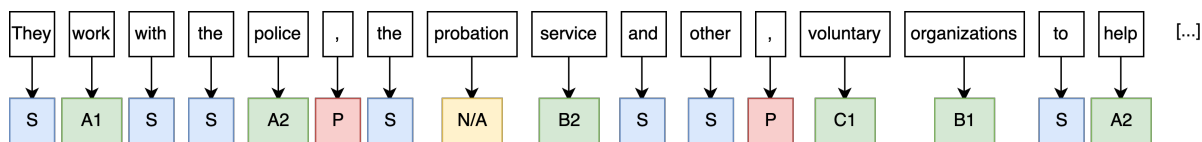


Figure 2: Annotation example on OneStopEnglish. S: stopword; P: punctuation; N/A: word not featured in the EVP.

all possible EVP entries for that given word. The model’s task is to select the most contextually appropriate meaning from the options provided. Note that multi-word expressions are also included by feeding their reference words, e.g., the word *push* for the expression *push (sb) for sth/to do sth* (see Figure 1).

The prompt used for this task and a relevant example are reported in Appendix C (“Prompt for semantic understanding”). For this experiment, we focused on words with 3, 4, 5, and 6 possible meanings. To avoid positional bias (Liusie et al., 2024), all possible permutations were considered for words with 3 options.⁸ For words with 4, 5, and 6 options, however, only ten permutations were considered due to time and resource constraints. For each permutation, we extract the logits for each option, apply a softmax function, then compute the average probability across permutations and select the option with the highest average probability.

For this experiment, we compared the performance of two proprietary LLMs, i.e., GPT-4o, GPT-4o-mini (OpenAI, 2023), and three open-source LLMs, i.e., Llama 3.1 8B, Llama 3.1 70B (4-bit quantised) (Llama Team, 2024), and Qwen 2.5 32B (4-bit quantised) (Yang et al., 2024). These models were selected to ensure a representative range in terms of both model size and the open-source versus proprietary distinction.

Results are evaluated in terms of Accuracy.

4.2 Word-level proficiency prediction

In the second part of our work, we perform CEFR proficiency level prediction at the word level on the annotated sentences extracted from the OneStopEnglish dataset (see Section 3.2 and Figure 2). We implement the approach proposed in Figure 3 in order to extract the proficiency level for each word in a given composition.

⁸In this context, a permutation refers to a unique ordering of the multiple-choice options associated with each question. That is, while the content of the options remains the same, their positions (e.g., labeled A, B, and C) are shuffled across different permutations. This ensures that the model’s predictions are not influenced by the fixed position of any particular option, thereby reducing positional bias.

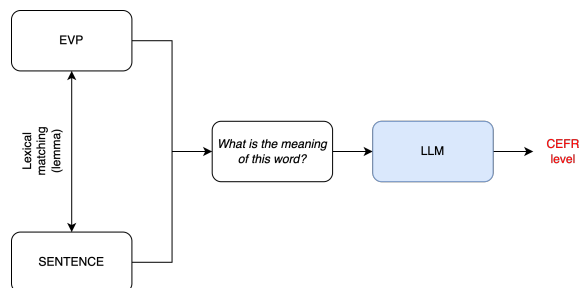


Figure 3: High-level diagram of the approach proposed for word-level CEFR prediction.

Since it is not feasible to feed all EVP entries for each word in a sentence into the LLM at once, we use spaCy⁹ to lemmatise each word and map the resulting lemma to its corresponding entries in the EVP. For example, for the word *work* in the sentence shown in Figure 2, the lemma remains *work*, which we then match to all corresponding *work* entries in the EVP. This allows us to filter out irrelevant EVP words in each LLM run.

[They] work with the police [...]
 They [work] with the police [...]
 They work [with] the police [...]
 They work with [the] police [...]
 They work with the [police] [...]

Figure 4: Highlighting method.

For each sentence we iterate through each word by highlighting it in square brackets, as shown in Figure 4. If there is no match between a given lemmatised word in the sentence and the EVP, we automatically assign *N/A* to the word (see Figure 2). In all the other cases, we add a “None of the other options” choice to the other available options and ask the LLM to choose the most suitable option. Then, we extract the logits for each option, apply a softmax function, and select the option with the highest average probability. Finally, we select the

⁹spacy.io

CEFR level assigned to this option. In this set of experiments, we also feed the manually assigned PoS information and a brief definition associated to each option.

The prompt used for these experiments as well as a relevant example can be found in Appendix C (“Prompt for word-level CEFR prediction”).¹⁰

It is important to note that our approach can effectively handle multi-word expressions. For example, when encountering the phrasal verb *take off* in the sense of “becoming airborne”, the method processes *take* and *off* separately. The word *off* is treated as a stopword and thus excluded from the final assessment. However, *take* is still evaluated in context, allowing the LLM to infer the intended meaning from the surrounding sentence or essay, in addition to the PoS information (i.e., *verb*) and the EVP definition (i.e., *If an aircraft takes off, it leaves the ground and begins to fly.*). This ensures that the CEFR level assigned reflects the actual usage rather than the isolated form. Similarly, for the expression *take advantage of sth*, both *take* and *advantage* are mapped to the same corresponding multi-word expression in the EVP, while the stopword *of* is excluded from the evaluation.

For this part of our work, we try the same LLMs as Section 4.1 except for GPT-4o-mini and Llama 3.1 8B. We compare the performance of these LLMs with a PoS-based system, which only relies on PoS tags extracted using spaCy. Specifically, if a given lemma in the sentence has only one entry in the EVP then its corresponding level is automatically assigned to the word (e.g., *voluntary* in Appendix B has only one EVP entry, hence one CEFR level). Otherwise the relevant PoS tag is matched with the corresponding one in the EVP. If the EVP only specifies one CEFR level for this lemma/PoS pair then this level is automatically assigned to the word (e.g., *criminal* in Appendix B has two entries – noun and adjective – at two different CEFR levels). Where the pairing is assigned to multiple CEFR levels in the EVP, the lowest of these CEFR levels is assigned to the word (e.g., the word *service* in the example in Appendix B has 6 noun entries with 3 different CEFR levels). Additionally, we consider a random baseline that relies solely on lexical matching. In this case, for words with a single entry, such as *voluntary*, after lemmatisation, we simply assign the associated CEFR

level. For words with multiple levels, we randomly pick a level from the available EVP entries.

Results are evaluated in terms of overall Accuracy. Additionally, performance on individual CEFR levels is reported using F_1 score.

4.3 Essay-level proficiency prediction

The third part of our work applies the approach presented in the previous section to additional L2 learner datasets, namely EFCAMDAT and ELLIPSE (see Sections 3.2). Unlike the On-StopEnglish data, these datasets are annotated only at the essay level and lack word-level CEFR labels. We therefore use our approach to predict the CEFR level of each word and leverage this information as features for essay-level proficiency prediction. For this part of the work, we only used Qwen 2.5 32B¹¹ and the PoS-based model to extract vocabulary-related information. In other words, we aim to explore the predictive power of vocabulary-related features extracted through our models in the context of holistic proficiency assessment. Additionally, since the ELLIPSE data include analytic scores targeting specific aspects of proficiency, we further investigate the relationship between these features and scores related to dimensions such as cohesion, syntax, vocabulary, phraseology, grammar, and punctuation conventions.

To do this, for EFCAMDAT, we employ a naive classifier that uses the distribution of predicted CEFR levels within each essay. Specifically, we compute the proportion of words at each predicted CEFR level (i.e., the count of words at each level divided by the total essay length), weight these proportions by their corresponding CEFR levels (i.e., 1 for A1, 2 for A2, 3 for B1, etc.), and sum them to obtain a composite score. We then assess the correlation between this score and the human-assigned holistic score. In addition to the naive classifier, we also employ a simple Support Vector Regression (SVR) model with default parameters (i.e., $\epsilon = 0.1$, $C = 1$, and a radial basis function kernel). The model is trained using the proportions of words at each predicted CEFR level as input features and evaluated using 5-fold cross-validation.

The SVR is also employed for the ELLIPSE dataset. The model is trained on the training split to predict the holistic proficiency scores. Subse-

¹⁰We also tried this task by prompting the LLMs without EVP information, but results were significantly worse.

¹¹The prompt is the same reported in Appendix C (“Prompt for word-level CEFR prediction”) with the only difference that “sentence” is replaced with “essay”.

quently, it is evaluated on the test set for both holistic and analytic scores prediction (see Section 3.2).

Results are evaluated in terms of Pearson’s correlation coefficient (PCC) and Spearman’s rank coefficient (SRC).

Experimental results

4.4 Semantic understanding results

Table 1 reports the results in terms of Accuracy for the semantic understanding task. As can be seen, GPT-4o achieves the best performance, followed by Llama 3.1 70B, with Qwen 2.5 32B performing nearly on par. While model size appears to play a significant role in the performance gap between GPT-4o and the other models, this pattern does not hold when comparing Qwen 2.5 32B to Llama 3.1 70B, despite the latter being more than twice as large. All models show a remarkable performance for this task with the exception of Llama 3.1 8B. As expected, Accuracy decreases as the number of options available for a given word increases.

A reasonable question to ask is whether these LLMs have been exposed to the EVP during training, given that it is publicly available. This consideration, among others, motivated us to extend our experiments to additional L2 learner datasets.

Model	No. of options				avg.
	3	4	5	6	
GPT-4o	89.0	86.2	83.1	79.5	84.4
GPT-4o_{mini}	84.4	78.9	75.4	71.2	77.5
Llama3.1_{70B}	85.1	82.0	78.9	73.1	79.8
Llama3.1_{8B}	77.4	70.0	64.3	64.4	69.0
Qwen2.5_{32B}	85.6	80.8	76.4	75.4	79.6

Table 1: Accuracy (%) of EVP classification results.

4.5 Word-level proficiency prediction results

Model	Ambig.	Non-amb.	All
Random	29.1	88.7	61.6
PoS-based	66.7	93.4	80.7
GPT-4o	75.1	90.5	83.3
Llama 3.1 70B	76.9	91.5	84.6
Qwen 2.5 32B	80.5	92.8	87.0

Table 2: Accuracy (%) results for word-level CEFR prediction (OneStopEnglish).

Table 2 shows the results in terms of Accuracy for the task of word-level CEFR level prediction on the OneStopEnglish data. As mentioned in

Section 3.1, we refer to words that have multiple CEFR levels within a single PoS as *ambiguous* words, whereas the rest are considered *non-ambiguous*. We report the results for ambiguous, non-ambiguous, and all words. As expected, the Random classifier performs extremely poorly when predicting the level of ambiguous words. Incorporating PoS information leads to noticeable improvements, as demonstrated by the PoS-based model. However, the best performance on both ambiguous and overall cases is achieved – in increasing order – by GPT-4o, Llama 3.1 70B, and Qwen 2.5 32B. Remarkably, Qwen, despite being the smallest of the three LLMs, outperforms all others on this task. The improved performance of LLMs is likely due to their ability to leverage semantic information in addition to grammatical and syntactic knowledge.

For non-ambiguous words, as expected, the Random classifier achieves relatively decent performance. In this setting, the PoS-based model yields the highest accuracy, followed closely by Qwen 2.5 32B. One might expect the PoS-based model to achieve perfect accuracy; however, this is not the case, as the data include instances where words are used with meanings that are not covered in the EVP, leading to misclassifications by the PoS-based model. To address such cases, we included a “None of the other options” choice among the LLM’s available options (see Section 4.2) – though this option appears to be over-selected by the model, potentially affecting its accuracy.

Another reason why the LLMs do not outperform the PoS-based model for non-ambiguous words lies in the way options are selected. When multiple PoS entries exist for a given lemma, the PoS-based model uses only the relevant, non-ambiguous entry matching the POS tag. In contrast, all available options – including those with irrelevant PoS tags – are fed into the LLMs. For instance, as illustrated in Section 3.1, if a sentence contains the word *aim* used as a noun (which appears in the EVP only at the B1 level), the PoS-based model considers only this entry. However, the LLM is presented with both the noun and verb entries (at A2, B2, C1, and C2 in the EVP) for *aim*.¹² These findings are further supported by the performance figures in Table 9 in Appendix D, which reports the breakdown by word-level CEFR level in terms of F_1 score.

¹²We deliberately chose not to filter out these entries to avoid assuming perfect accuracy from the spaCy tagger.

Overall, these results suggest that the most effective solution may lie in a hybrid approach, where an LLM is used to handle ambiguous words, while a PoS-based model deals with non-ambiguous ones.

4.6 Essay-level proficiency prediction results

We use the same approach to predict word-level CEFR levels on additional L2 learner data, which are only annotated at the essay level. Starting from EFCAMDAT, Figure 5 shows a cumulative plot of the predicted normalised distribution of words for each word-level CEFR level across essay-level CEFR levels. We intentionally reverse the order of CEFR levels on the x-axis to emphasise that vocabulary usage is indicative of a certain proficiency level or higher, rather than exclusive to that level. For example, B1-level items are also commonly used by more advanced learners.

Focusing on a specific element of the figure, if we observe B1 level essays (represented by the green line), we observe that vocabulary from the C2 and C1 levels is used very little, while B2 level vocabulary appears to some extent – more than in A1 and A2 level essays, but less than in B2, C1, and C2 level essays. Vocabulary from B1 and lower levels is used even more frequently.

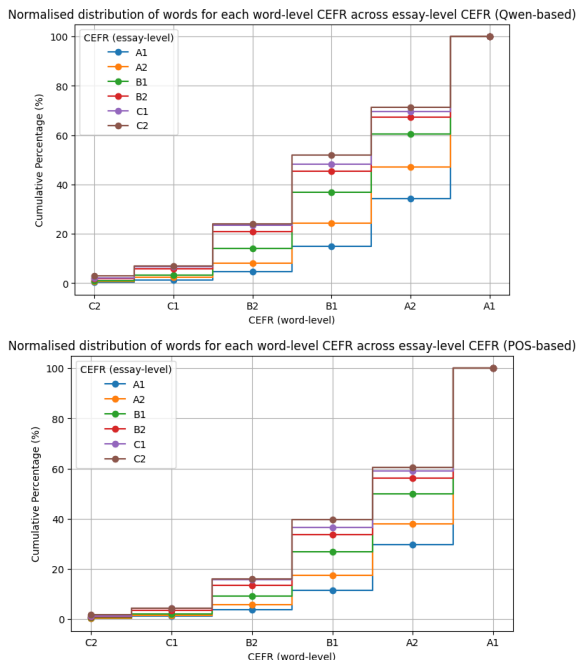


Figure 5: Predicted normalised distribution of words for each word-level CEFR level across essay-level CEFR levels. Qwen 2.5 32B vs PoS-based (EFCAMDAT).

When considering the overall trends, words from A2 to B2 show a steady increase in usage as essay-

level proficiency progresses. Interestingly, higher-level words (i.e., C1 and C2) are not as frequently used even in essays written at higher proficiency levels. Nonetheless, a moderate distinction across essay-level proficiency bands can still be observed. To this end, we present Figure 6, which displays the empirical cumulative distribution function (eCDF) of the AUC (Area Under the Curve) values computed from Figure 5. In both figures, we observe larger gaps – indicating better differentiation across levels – when using Qwen compared to the PoS-based model.

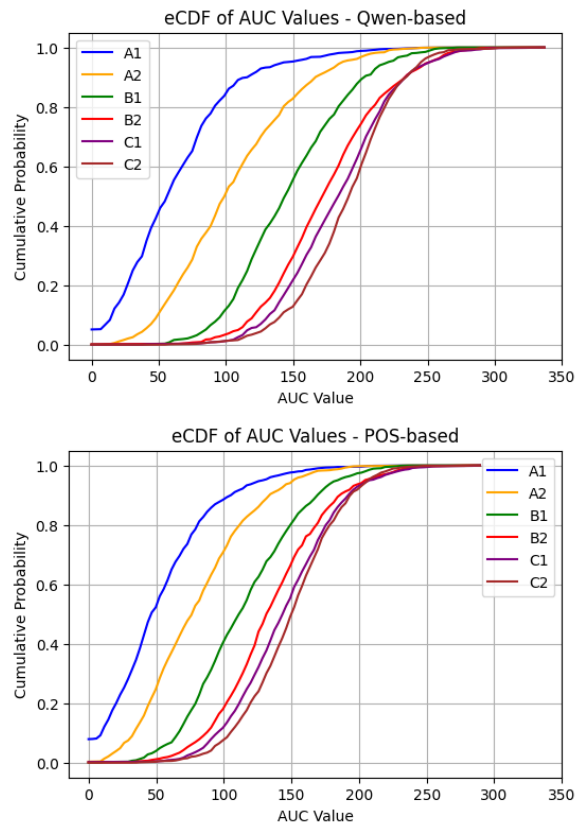


Figure 6: eCDF of AUC values. Qwen 2.5 32B vs PoS-based (EFCAMDAT).

These findings are further supported by the essay assessment results reported in Table 3. When used as features for predicting holistic scores, the vocabulary information extracted with Qwen consistently outperforms that derived from the PoS-based model both in the naive classifier and by the SVR.¹³

Finally, we report the results of our experiments on ELLIPSE using Qwen in combination with an SVR trained on holistic scores (see Section 4.3). As shown in Table 4, despite the high inter-correlation

¹³These experiments aim to demonstrate the predictive power of the extracted features, not to achieve state-of-the-art essay scoring.

Features	Classifier	PCC	SRC
PoS	Naive	0.580	0.603
Qwen		0.636	0.656
PoS	SVR	0.734	0.713
Qwen		0.771	0.749

Table 3: Results for essay-level holistic proficiency prediction (EFCAMDAT).

among all analytic scores (see Table 8 in Appendix A), the SVR (trained on holistic scores) predictions correlate most strongly with the Vocabulary scores, followed by Phraseology, suggesting our features are effectively targeting lexical aspects of language.

	PCC	SRC
Overall	0.650	0.624
Vocabulary	0.637	0.627
Phraseology	0.630	0.614
Grammar	0.577	0.556
Syntax	0.584	0.547
Cohesion	0.595	0.569
Conventions	0.613	0.578

Table 4: Results for essay-level holistic and analytic proficiency prediction using Qwen-SVR (ELLIPSE).

4.7 Word-level analysis

Finally, to evaluate the consistency of the EVP, we reverse the approach used thus far. Specifically, we select the two most common words with multiple meanings in the EFCAMDAT data (i.e., *work* and *like*) and examine their word-level CEFR level distribution (predicted with Qwen) across essay-level CEFR levels, as shown in Figures 7 and 8. The captions report the number of essays containing these words for each essay-level CEFR level. Figures 7

	<i>work</i>	<i>like</i>
≥B2	88.6	-
≥B1	90.6	89.6
≥A2	96.8	99.4

Table 5: EVP consistency in terms of Accuracy (%) for words *work* and *like* (EFCAMDAT).

and 8 are summarised in Table 5, where we assess the consistency of the EVP for these two words in terms of Accuracy. As discussed in Section 4.6, vocabulary use typically reflects a certain proficiency level or higher, rather than being exclusive to a specific level. Accordingly, we compute Accuracy

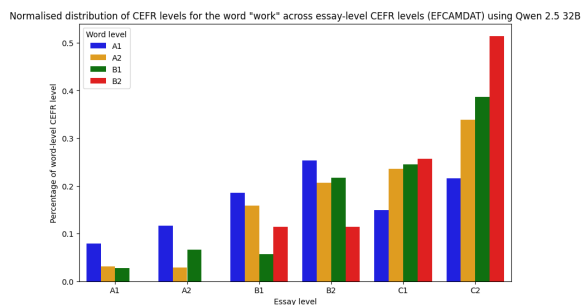


Figure 7: Distribution for word *work* (A1: 191; A2: 278; B1: 482; B2: 667; C1: 457; C2: 667).

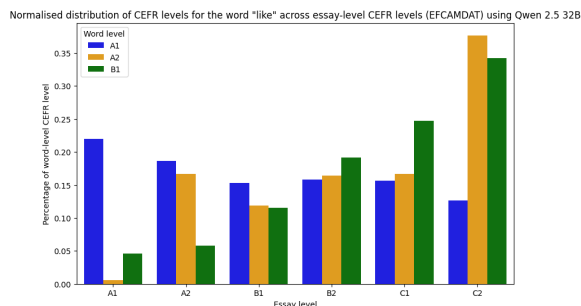


Figure 8: Distribution for word *like* (A1: 401; A2: 411; B1: 380; B2: 459; C1: 497; C2: 591).

by checking whether a word appears in essays at its assigned level or above. It is also important to note that we are comparing word-level *vocabulary* levels with essay-level *holistic* scores, where vocabulary represents only one component of the overall assessment. The results show high accuracy and suggest a strong degree of consistency in the CEFR classification provided by the EVP.

5 Conclusions and future work

In this work, we introduced a novel approach to in-context, word-level L2 vocabulary assessment by leveraging LLMs in combination with the English Vocabulary Profile. We compared the performance of several open-source and proprietary LLMs to a PoS-based model and showed that the former are particularly effective for this task, especially in handling polysemy and lexical ambiguity.

We plan to integrate this approach into an automatic essay grading system, where it could enrich holistic scoring with fine-grained feedback on vocabulary use, for example, by identifying lexical gaps relative to the learner’s proficiency or highlighting advanced vocabulary usage as a strength. Additionally, we plan to extend our experiments to spoken transcriptions in order to further evaluate the robustness of LLMs and assess their effectiveness across different modalities.

Limitations

One limitation of this study is the lack of a systematic investigation into how learner errors affect our approach. In particular, lemmatisation and matching with EVP entries may be hindered by such errors.¹⁴ While spelling mistakes can be addressed with a spellchecker, grammatical or lexical errors pose a greater challenge. In this case, it would be interesting to test our approach on pairs of original and grammatically corrected (manually and/or automatically) sentences or essays and analyse shifts in the LLM's probability distributions in the presence of learner errors.

Another limitation of our approach lies in the way word-level proficiency labels are assigned. In our experiments, each word in the OneStopEnglish data was annotated with its CEFR level, not its specific sense. As a result, polysemous words with multiple meanings but identical CEFR levels could be matched to an incorrect sense, even though the level remains technically accurate. However, it is important to note that the LLMs are prompted to identify the intended meaning of each word based on its context. The CEFR level is then assigned post hoc by mapping that selected meaning to the corresponding entry in the English Vocabulary Profile, hence to a CEFR level.

Finally, due to space constraints, we did not conduct a focused analysis on multi-word expressions such as idioms and phrasal verbs. Nonetheless, given the strong overall results and the fact that multi-word expressions are included in our data, it is reasonable to assume that our approach also performs well on these cases.

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.

¹⁴In our first experiment specifically, learner example sentences from the EVP may contain errors, but not involving the target word. Additionally, this experiment does not require lemmatisation.

References

- 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, Yuri Bizzoni, Anders Agebjörn, Elena Volodina, and Ildikó Pilán. 2016. [From distributions to labels: A lexical proficiency analysis using learner corpora](#). In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, pages 1–7, Umeå, Sweden. LiU Electronic Press.
- 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.
- Richard C. Anderson and Peter Freebody. 1981. Vocabulary knowledge. In J. P. Guthrie, editor, *Comprehension and teaching: Research reviews*, pages 77–117. International Reading Association, Newark.
- Richard C. Anderson and Peter Freebody. 1983. Reading comprehension and the assessment and acquisition of word knowledge. In B. Hutson, editor, *Advances in reading / language research*, volume 2. JAI Press, Greenwich.
- Yuki Arase, Satoru Uchida, and Tomoyuki Kajiwara. 2022. [CEFR-based sentence difficulty annotation and assessment](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6206–6219, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Melissa M. Baese-Berk, Shiloh Drake, Kurtis Foster, Dae-yong Lee, Cecelia Staggs, and Jonathan M. Wright. 2021. Lexical diversity, lexical sophistication, and predictability for speech in multiple listening conditions. *Frontiers in psychology*, 12:661415.
- Stefano Bannò, 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.
- Stephen Bax. 2012. [Text inspector](#). *Online text analysis tool*.
- Annette Capel. 2015. The English Vocabulary Profile. In J. Harrison and F. Barker, editors, *English Profile in Practice*. Cambridge University Press, Cambridge.

- Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press, Cambridge.
- Council of Europe. 2020. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment - Companion volume*. Council of Europe, Strasbourg.
- Scott Crossley, Yu Tian, Perpetual Baffour, Alex Franklin, Youngmeen Kim, Wesley Morris, Meg Benner, Aigner Picou, and Ulrich Boser. 2023. The English Language Learner Insight, Proficiency and Skills Evaluation (ELLIPSE) Corpus. *International Journal of Learner Corpus Research*, 9(2):248–269.
- Scott A Crossley and Danielle S McNamara. 2012. Predicting second language writing proficiency: The roles of cohesion and linguistic sophistication. *Journal of Research in Reading*, 35(2):115–135.
- 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).
- Núria Gala, Thomas François, Delphine Bernhard, and Cédric Fairon. 2014. Un modèle pour prédire la complexité lexicale et graduer les mots. In *TALN'2014*, pages 91–102.
- Aina Garí Soler and Marianna Apidianaki. 2021. Let's play monopoly: BERT can reveal words' polysemy level and partitionability into senses. *Transactions of the Association for Computational Linguistics*, 9:825–844.
- Atta Gebril and Lia Plakans. 2016. [Source-based tasks in academic writing assessment: Lexical diversity, textual borrowing and proficiency](#). *Journal of English for Academic Purposes*, 24:78–88.
- Jeroen Geertzen, Theodora Alexopoulou, and Anna Korhonen. 2013. [Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database \(EFCAMDAT\)](#). In *Proceedings of the 31st Second Language Research Forum*, pages 240–254, Somerville. Cascadilla Proceedings Project.
- Liz Hamp-Lyons. 1995. [Rating nonnative writing: The trouble with holistic scoring](#). *TESOL Quarterly*, 29(4):759–762.
- Luke Harding. 2014. [Communicative language testing: Current issues and future research](#). *Language Assessment Quarterly*, 11(2):186–197.
- Yan Huang, Jeroen Geertzen, Rachel Baker, Anna Korhonen, and Theodora Alexopoulou. 2017. The EF Cambridge Open Language Database (EFCAMDAT): Information for users.
- Shi Huawei and Vahid Aryadoust. 2023. A systematic review of automated writing evaluation systems. *Education and Information Technologies*, 28(1):771–795.
- Kristopher Kyle, Scott Crossley, and Cynthia Berger. 2018. The tool for the automatic analysis of lexical sophistication (TAALES): version 2.0. *Behavior research methods*, 50(3):1030–1046.
- Kristopher Kyle and Scott A Crossley. 2015. Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly*, 49(4):757–786.
- Batia Laufer and Paul Nation. 1995. Vocabulary size and use: Lexical richness in L2 written production. *Applied Linguistics*, 16(3):307–322.
- Paul Lennon. 2008. Contrastive analysis, error analysis, interlanguage. In S. Gramley and V. Gramley, editors, *Bielefeld Introduction to Applied Linguistics. A Course Book. Bielefeld*, pages 51–60. Aisthesis, Bielefeld.
- Agnieszka Leńko-Szymańska. 2015. The English Vocabulary Profile as a benchmark for assigning levels to learner corpus data. In M. Callies and S. Götz, editors, *Learner Corpora in Language Testing and Assessment*. John Benjamins, Amsterdam; Philadelphia.
- 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.
- Llama Team. 2024. [The Llama 3 herd of models](#).
- Xiaofei Lu. 2012. The relationship of lexical richness to the quality of ESL learners' oral narratives. *The Modern Language Journal*, 96(2):190–208.
- Paul Meara and Barbara Buxton. 1987. An alternative to multiple choice vocabulary tests. *Language Testing*, 4(2):142–154.
- Paul Nation. 1983. Testing and teaching vocabulary. *Guidelines*, 5(1):12–25.
- Diane Nicholls. 2003. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 Conference*, pages 572–581.
- OpenAI. 2023. [GPT-4 Technical Report](#). Preprint, arXiv:2303.08774.
- John Read. 2000. *Assessing vocabulary*. Cambridge University Press, Cambridge.
- John Read. 2013. Second language vocabulary assessment. *Language Teaching*, 46(1):41–52.

- Norbert Schmitt, Diane Schmitt, and Caroline Clapham. 2001. Developing and exploring the behaviour of two new versions of the Vocabulary Levels Test. *Language Testing*, 18(1):55–88.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for computational Linguistics*, 8:247–263.
- Yukio Tono. 2013. The CEFR-J handbook: A resource book for using CAN-DO descriptors for English language teaching. *Tokyo: Taishukan*.
- Jeanine Treffers-Daller, Patrick Parslow, and Shirley Williams. 2018. Back to basics: How measures of lexical diversity can help discriminate between cefr levels. *Applied Linguistics*, 39(3):302–327.
- Satoru Uchida and Masashi Negishi. 2018. Assigning CEFR-J levels to English texts based on textual features. In *Proceedings of Asia Pacific Corpus Linguistics Conference*, volume 4, pages 463–467.
- Sowmya Vajjala and Ivana Lučić. 2018. [OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification](#). pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.
- Anne Vermeer. 2001. Breadth and depth of vocabulary in relation to L1/L2 acquisition and frequency of input. *Applied Psycholinguistics*, 22(2):217–234.
- Sara Cushing Weigle. 2002. *Assessing Writing*. Cambridge Language Assessment. Cambridge University Press.
- Kelly Woods, Brett Hashimoto, and Earl K. Brown. 2023. [A multi-measure approach for lexical diversity in writing assessments: Considerations in measurement and timing](#). *Assessing Writing*, 55:100688.
- 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, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.
- Su-Youn Yoon, Suma Bhat, and Klaus Zechner. 2012. Vocabulary profile as a measure of vocabulary sophistication. In *Proceedings of the seventh workshop on building educational applications using NLP*, pages 180–189.
- Guoxing Yu. 2010. Lexical diversity in writing and speaking task performances. *Applied Linguistics*, 31(2):236–259.

A Appendix A: Other stats

Table 6 shows the degree of polysemy in the EVP.

Table 7 shows the scores distribution in the selected ELLIPSE essays. Note that, in the original

No. of options	count (%)
1	56.96
2	21.13
3	8.82
4	3.99
5	2.59
6	1.53
> 6	4.98

Table 6: Degree of polysemy in the EVP.

corpus, higher (i.e., 4.5 and 5) and lower (i.e., 1 and 1.5) proficiency levels are underrepresented compared to intermediate levels.

Overall Score	# Essays	
	Train	Test
5	28	12
4.5	13	7
4	87	43
3.5	39	23
3	61	27
2.5	62	31
2	38	19
1.5	22	11
1	9	2

Table 7: Distribution of ELLIPSE essays by overall score in the considered training and test sets.

Table 8 shows the Correlation between ground truth analytic scores and ground truth overall scores in terms of SRC for the ELLIPSE test split considered in this work.

B Appendix B: OneStopEnglish examples

The following is an example of three parallel sentences across the three proficiency levels drawn from OneStopEnglish:

Elementary: *They work with the police, the probation service and other, voluntary organizations to help members of the violent criminal gangs of London.*

Intermediate: *They work with the police, the probation service and other, voluntary organizations to help people who feel trapped and frightened in the violent criminal gangs of London.*

Advanced: *They work with the police, the probation service and other, voluntary organizations to help those who feel trapped and frightened in the*

	Overall	Vocabulary	Phraseology	Cohesion	Grammar	Syntax	Conventions
Overall	1.000	0.869	0.897	0.880	0.878	0.911	0.873
Vocabulary	0.869	1.000	0.863	0.776	0.776	0.798	0.748
Phraseology	0.897	0.863	1.000	0.828	0.835	0.847	0.811
Cohesion	0.880	0.776	0.828	1.000	0.781	0.825	0.816
Grammar	0.878	0.776	0.835	0.781	1.000	0.840	0.794
Syntax	0.911	0.798	0.847	0.825	0.840	1.000	0.816
Conventions	0.873	0.748	0.811	0.816	0.794	0.816	1.000

Table 8: Correlation between ground truth analytic scores and ground truth overall scores in terms of SRC.

violent criminal gangs that operate across London.

C Appendix C: Prompts

Prompt for semantic understanding

Read this sentence: **[EVP SENTENCE]**
Choose the correct meaning of **[WORD]** by selecting the most suitable among the following options A, B, C, D, E, or F. No other answer is allowed. Only output the respective option letter without any additional comments, notes, or explanations.

A) **[DEFINITION A]**

B) **[DEFINITION B]**

C) **[DEFINITION C]**

D) **[DEFINITION D]**

E) **[DEFINITION E]**

F) **[DEFINITION F]**

Example

Read this sentence: **“It was tough on the worn out employees.”** Choose the correct meaning of **“tough”** by selecting the most suitable among the following options A, B, C, D, E, or F. No other answer is allowed. Only output the respective option letter without any additional comments, notes, or explanations.

A) not easy to break or damage

B) describes food that is difficult to cut or eat

C) Tough people are mentally strong and not afraid of difficult situations.

D) difficult

E) Tough rules are severe.

F) unfair or unlucky

Prompt for word-level CEFR prediction

Read this L2 learner sentence: **[SENTENCE]**

Choose the correct meaning of **[WORD]** (in square brackets) by selecting the most suitable among the following options. Also consider the additional information and the PoS of each option. No other answer is allowed. Only output the respective option number without any additional comments, notes, or explanations.

1. **[DEFINITION 1]** - Additional information: **[INFO]** (PoS)

2. **[DEFINITION 2]** - Additional information: **[INFO]** (PoS)

3. **[DEFINITION 3]** - Additional information: **[INFO]** (PoS)

[...]

n. None of the other options.

where the additional information consists of a brief definition of the word. In round brackets, we also feed the information related to the manually assigned PoS as described in the EVP. See the example below for further information.

The prompt for the experiments conducted on essay-level proficiency prediction is the same. The only difference is that “sentence” is replaced with “essay”.

Example

Read this L2 learner sentence: **They [work] with the police , the probation service and other , voluntary organizations to help those who feel trapped and frightened in the violent criminal gangs that operate across London .**

Choose the correct meaning of **“work”** (in square brackets) by selecting the most suitable among the following options.

Also consider the additional information and the part of speech of each option. No other answer is allowed. Only output the respective option number without any additional comments, notes, or explanations.

1) the place where you go to do your job - Additional information: work (PLACE) (Part of speech: noun)

2) something you do as a job to earn money - Additional information: work (JOB) (Part of speech: noun)

3) to do a job, especially the job you do to earn money - Additional information: work (DO JOB) (Part of speech: verb)

4) the activities that you have to do at school, for your job, etc. - Additional information: work (ACTIVITY) (Part of speech: noun)

5) If a machine or piece of equipment works, it is not broken. - Additional information: work (OPERATE) (Part of speech: verb)

6) when you use physical or mental effort to do something - Additional information: work (EFFORT) (Part of speech: noun)

7) If something works, it is effective or successful. - Additional information: work (SUCCEED) (Part of speech: verb)

8) to exercise in order to improve the strength or appearance of your body - Additional information: work out (EXERCISE) (Part of speech: verb)

9) a painting, book, piece of music, etc. - Additional information: work (CREATION) (Part of speech: noun)

10) to try hard to achieve something - Additional information: work at sth (Part of speech: verb)

11) to spend time repairing or improving something - Additional information: work on sth (Part of speech: verb)

12) to do a calculation to get an answer to a mathematical question - Additional information: work sth out or work out sth (Part of speech: verb)

13) If a problem or a complicated situation works out, it ends in a success-

ful way. - Additional information: work out (BECOME BETTER) (Part of speech: verb)

14) to know how to use a machine or piece of equipment - Additional information: can work sth; know how to work sth (Part of speech: verb)

15) to understand something or to find the answer to something by thinking about it - Additional information: work sth out or work out sth (UNDERSTAND) (Part of speech: verb)

16) None of the other options

D Appendix D: Other results

Table 9 reports the breakdown by word-level CEFR level in terms of F_1 score. For ambiguous cases, the PoS-based model performs reasonably well only at the A1 level. However, this result is partly influenced by the rule we set, i.e., assigning the lowest available CEFR level when multiple options are present (see Section 4.2), which inherently favours lower-level classifications. For both the N/A and C2 levels, the model yields an F_1 score of 0, highlighting its complete inability to handle these cases. For non-ambiguous cases, the PoS-based achieves the best performance on all the CEFR levels for the reasons explained in Section 4.5, with the exception of the C2 level. For this specific CEFR level, the PoS-based model obtains a high Recall (99.28%) but a low Precision (73.02%), while Qwen shows more balanced results, with a Recall of 90.65% and a Precision of 89.36%.

	Ambiguous		Non-ambiguous		All	
	Qwen 2.5 32B	PoS	Qwen 2.5 32B	PoS	Qwen 2.5 32B	PoS
N/A	50.0	0.0	94.3	94.4	92.0	91.6
A1	84.4	83.1	91.3	94.0	86.5	85.8
A2	82.1	67.4	89.8	91.8	85.1	75.6
B1	80.6	56.7	94.3	95.1	85.7	72.9
B2	78.5	49.3	90.5	90.8	84.2	72.0
C1	75.5	20.0	91.9	92.3	82.9	61.5
C2	67.2	0.0	90.0	84.2	78.9	59.7

Table 9: Breakdown of classification results by word-level CEFR level in terms of F_1 (OneStopEnglish).

Advancing Question Generation with Joint Narrative and Difficulty Control

Bernardo Leite and Henrique Lopes Cardoso
LIACC, Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
{bernardo.leite, hlc}@fe.up.pt

Abstract

Question Generation (QG), the task of automatically generating questions from a source input, has seen significant progress in recent years. Difficulty-controllable QG (DCQG) enables control over the difficulty level of generated questions while considering the learner’s ability. Additionally, narrative-controllable QG (NCQG) allows control over the narrative aspects embedded in the questions. However, research in QG lacks a focus on combining these two types of control, which is important for generating questions tailored to educational purposes. To address this gap, we propose a strategy for Joint Narrative and Difficulty Control, enabling *simultaneous* control over these two attributes in the generation of reading comprehension questions. Our evaluation provides preliminary evidence that this approach is feasible, though it is not effective across all instances. Our findings highlight the conditions under which the strategy performs well and discuss the trade-offs associated with its application.

1 Introduction

Question Generation (QG) focuses on the automated generation of coherent and meaningful questions targeting a data source, including unstructured text or knowledge bases (Rus et al., 2008). Controllable QG plays a crucial role in education (Kurdi et al., 2020), as it facilitates the generation of personalized questions that address the unique needs and learning goals of students. Recent work on QG utilized techniques such as fine-tuning (Zhang et al., 2021; Ushio et al., 2022) and few-shot prompting (Wang et al., 2022b; Chen et al., 2024) to generate questions based on a source text and, optionally, a target answer. In controllable QG, this process is augmented by incorporating controllability labels into the input or prompt to guide the generation process. Specifically, research on Narrative-Controlled

Passage: Once there were a hare and a turtle. The hare was proud of his speed and challenged the turtle to a race. Although the turtle was slow, he accepted. The hare quickly left the turtle behind but decided to rest and fell asleep. Meanwhile, the turtle kept going steadily and eventually reached the finish line first, winning the race.

Narrative: “character” **Difficulty:** “easy”
Generated QA Pair: Who challenged the turtle to a race? The hare.

Narrative: “outcome” **Difficulty:** “medium”
Generated QA Pair: What happened after the hare left the turtle behind? Decided to rest and fell asleep.

Narrative: “outcome” **Difficulty:** “hard”
Generated QA Pair: What happened because the turtle kept going steadily? The turtle won the race.

Figure 1: Illustrative example of controlled question-answer generation with varying difficulty levels and narrative attributes.

Question Generation (NCQG) focuses on controlling the **content** of generated questions, guided by underlying narrative elements (e.g., causal relationship) (Zhao et al., 2022; Leite and Lopes Cardoso, 2023; Li and Zhang, 2024). In turn, Difficulty-Controllable Question Generation (DCQG) emphasizes controlling the expected difficulty in answering the questions (Gao et al., 2019; Kumar et al., 2019; Cheng et al., 2021; Bi et al., 2021). Some studies have considered the relationship between question **difficulty** and the **learner’s ability** (Uto et al., 2023; Tomikawa and Uto, 2024).

However, research in controllable QG lacks the combination of these two types of control, which is especially important to facilitate human control (Wang et al., 2022a) in the ever-increasing usage of generative models in this field. Therefore, this research proposes a strategy that explores the feasibility of joining narrative and difficulty control to generate reading comprehension question-answer

(QA) pairs from children-targeted narrative stories. Figure 1 shows an example of the strategy. Formally, we investigate the following research question (RQ): *How effectively can we control the generation of question-answer pairs conditioned on both narrative and difficulty attributes using a modest¹ scale model?*

For our experiments, we use a well-known dataset — FairyTaleQA (Xu et al., 2022) — in which each question is already annotated with one of seven narrative labels. Our method involves two main steps: (1) using simulated-learner QA systems to answer questions from FairyTaleQA, thereby estimating the difficulty labels via Item Response Theory, and (2) applying a joint narrative and difficulty control model, utilizing human-annotated narrative labels and the estimated difficulty labels for each question.

The proposed method is evaluated to determine whether both NCQG and DCQG have been successfully applied to the generated questions. For NCQG, we compare the similarity between human-authored and generated questions. For DCQG, we assess the performance of simulated-learner QA systems on questions generated with distinct difficulty levels. Although the results demonstrate the effectiveness of the strategy, NCQG shows consistent success, whereas DCQG exhibits moderate success, with performance varying across specific narrative attributes and difficulty levels. Our goal is to highlight the conditions under which the strategy performs with high or low efficacy, providing insights for researchers pursuing similar research lines. In summary, our contributions are:

- We propose a joint strategy for controlling the generation of question-answer pairs conditioned on narrative and difficulty attributes.
- We report on the linguistic features influenced by control and conduct an error analysis of the generated QA pairs, providing insights into the performance and limitations of the method.

2 Background and Related Work

2.1 Controllable Question Generation (CQG)

As stated by Li and Zhang (2024), prior research on CQG has explored two main perspectives: content (or type) and difficulty.

Content control relates to the linguistic elements incorporated into the generated questions. For instance, Ghanem et al. (2022) proposed controlling specific reading comprehension skills, such as figurative language and vocabulary. Additionally, Zhao et al. (2022) focused on controlling narrative elements, while Leite and Lopes Cardoso (2023) extended this approach by controlling explicitness attributes. Elkins et al. (2023) propose to control Bloom’s question taxonomy (Krathwohl, 2002).

Difficulty control is related to the challenge of answering the generated questions, a concept that is often subjective (i.e., difficulty can vary depending on the respondent). In this regard, Gao et al. (2019) assigned difficulty labels (easy or hard) to questions based on whether QA systems could answer them correctly and used these labels as inputs to control the generation process. Kumar et al. (2019) proposed estimating difficulty based on named entity popularity, while Bi et al. (2021) tackle the challenge of high diversity in QG. Furthermore, Cheng et al. (2021) controlled question difficulty by considering the number of inference steps required to arrive at an answer.

One limitation of previous approaches is (1) the lack of emphasis on the relationship between question difficulty and learner ability. Addressing this problem, Uto et al. (2023) proposed to use Item Response Theory (IRT) (Lord, 2012), a mathematical framework in test theory, to quantify question difficulty and directly relate it to learner ability. Another limitation is (2) the lack of integration of multiple attributes. While Li and Zhang (2024) combine both narrative and difficulty attributes, they define *difficulty* in terms of answer explicitness and the number of sentences needed to answer the questions. The novelty of this study lies in integrating content control, through narrative elements, with difficulty control *informed by simulated learners’ ability*, thus building on the foundations laid by previous research.

2.2 Item Response Theory (IRT)

IRT (Lord, 2012) is a statistical framework used to study the interaction between test-takers (ability or proficiency) and their performance on test items. A key aspect of IRT is to model the relationship between question difficulty and learner ability, offering insights into how well a question differentiates between individuals with varying levels of skill. This relationship allows for an estimation of

¹<1 billion of parameters.

the likelihood that a learner with a specific ability level can correctly answer a given question, making it particularly useful for adaptive testing and understanding question complexity. A commonly used model in IRT is the **Rasch model**, which assumes that the probability of a correct response depends on the relation between learner ability (θ) and the item’s difficulty (b):

$$P(X_{ij} = 1 \mid \theta_i, b_j) = \frac{e^{\theta_i - b_j}}{1 + e^{\theta_i - b_j}}, \quad (1)$$

where θ_i is the learner ability of individual i , b_j is the difficulty of item j , and $P(X_{ij} = 1 \mid \theta_i, b_j)$ is the probability that individual i correctly answers item j . In our study, we use IRT to estimate both question difficulty (b) and learner ability (θ) parameters.

2.3 FairyTaleQA: Purpose and Value

We use the FairyTaleQA dataset (Xu et al., 2022) because its stories and corresponding question-answer pairs align with the goal of addressing *narrative comprehension*. According to Xu et al. (2022), narrative comprehension represents a high-level cognitive skill closely linked to overall reading proficiency (Lynch et al., 2008). A key feature of FairyTaleQA is the expert annotations on each question, which are grounded in evidence-based frameworks (Paris and Paris, 2003; Alonzo et al., 2009). The annotated narrative elements targeted for control are:

- **Character:** Addresses identities or traits of story characters (e.g., “Who...?”);
- **Setting:** Focusing on the time and place of events, often starting with “Where...?” or “When...?”;
- **Action:** Related to the actions of characters;
- **Feeling:** Exploring emotional states or reactions (e.g., “How did/does X feel?”);
- **Causal relationship:** Addressing cause-and-effect (e.g., “Why...?” or “What caused/made X?”);
- **Outcome resolution:** Focusing on the outcomes of events (e.g., “What happened/happens after X?”);
- **Prediction:** Questions about future or unknown events based on textual evidence.

While there are other popular educational QA datasets (following the open-ended *wh*-questions format), such as NarrativeQA (Kočíský et al., 2018) and StoryQA (Zhao et al., 2023), they are not annotated with specific reading comprehension skills. This further motivated our decision to use FairyTaleQA in this study.

3 Method

This section outlines the methodology of this research, which includes augmenting FairyTaleQA with IRT-based difficulty labels and developing a question-answer pair generation model with joint narrative and difficulty control. Figure 2 provides an overview of the steps discussed in this section.

3.1 Augmenting FairyTaleQA With IRT-Based Question Difficulty Labels

Let D be our dataset consisting of instances represented as quartets:

$$D_i = (t, q, a, n), \quad (2)$$

where t is a text, q is the question, a is the answer about the text, and n is the narrative element associated with the question-answer pair (q, a). The aim is to create a fifth element d , resulting in a new instance augmented:

$$D_{i\text{-augmented}} = (t, q, a, n, d), \quad (3)$$

where d is the estimated difficulty value associated with the question-answer pair (q, a). To create these augmented instances, we used the method proposed by Uto et al. (2023) and Tomikawa et al. (2024):

1. **Collecting response data for each question-answer pair:** We collected answers to the questions from multiple respondents. Given the unavailability of real students, we utilized simulated-learner QA systems, which are models capable of automatically extracting answers to the posed questions. As explained in Section 4.2, the QA models were deliberately chosen to represent different levels of performance to simulate varying ability levels.
2. **Estimating Question Difficulty with IRT:** Using the answers collected from the simulated-learner QA systems, we estimated the difficulty of each question using IRT, specifically employing the Rasch model as described in Section 2.2.

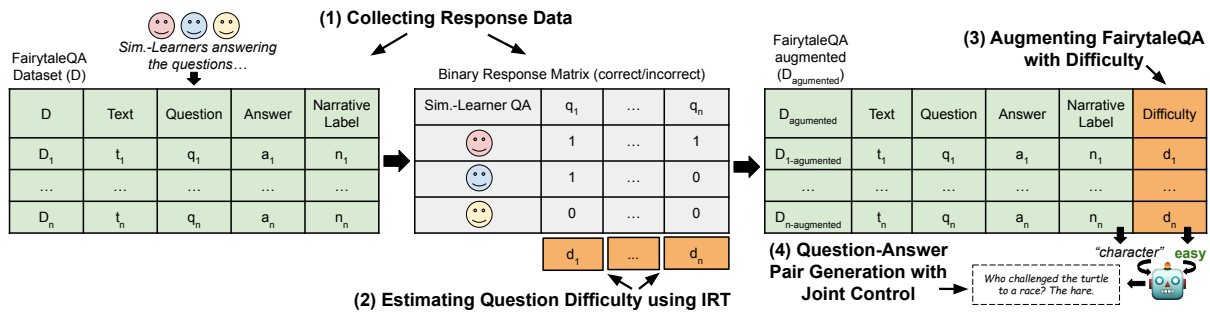


Figure 2: Overall methodology for joint narrative and difficulty control.

3. **Augmenting FairytaleQA with difficulty estimates:** Based on the estimated difficulty values, we augment each instance of the dataset with d , resulting in $D_{i\text{-augmented}} = (t, q, a, n, d)$.

3.2 Question-Answer Pair Generation with Joint Narrative and Difficulty Control

The controllable process can be represented as follows: given an instruction prompt p , the aim is to use a model M to generate a question-answer pair $(q_{\text{new}}, a_{\text{new}})$. This can be formulated as:

$$(q_{\text{new}}, a_{\text{new}}) = M(p), \quad (4)$$

where prompt p incorporates the desired narrative label n , difficulty value d , and target text t . The prompt follows this template:

“Generate a $\langle d \rangle$ question-answer pair about narrative label $\langle n \rangle$ considering the following text: $\langle t \rangle$ ”

M is an encoder-decoder model that is fine-tuned using $D_{i\text{-augmented}} = (t, q, a, n, d)$ instances. The encoder receives prompt p and encodes it into a fixed-length representation known as a context vector. The decoder takes the context vector and generates the output text $(q_{\text{new}}, a_{\text{new}})$, using special tokens $\langle \text{QU} \rangle$ and $\langle \text{AN} \rangle$ that serve to differentiate between q_{new} and a_{new} . The idea is to guide the model in generating a question-answer pair of the intended difficulty d and narrative element n .

4 Experimental Setup

4.1 Preparing the FairytaleQA Dataset

We use FairytaleQA (Xu et al., 2022), which comprises 10,580 question-answer pairs manually created by educational experts based on 278 narrative stories. Each story contains approximately 15 section texts, and each section (about 149 tokens) contains approximately 3 question-answer pairs. From

the original dataset, we have prepared different data setups² for generating a QA pair:

- **Text** \rightarrow **QA**: This setup only contains the text as input, so it serves as a baseline to compare with the subsequent setups, which consider control attributes.
- **Nar + Text** \rightarrow **QA**: This setup considers *narrative* as a control attribute in the input.
- **Dif + Text** \rightarrow **QA**: This setup considers *difficulty* as a control attribute in the input.
- **Nar + Dif + Text** \rightarrow **QA**: This setup considers both the narrative and difficulty attributes.

4.2 Creating Simulated-Learner QA Systems

To create the simulated-learner QA systems, we trained five QA models. The choice of five was made empirically: it provided sufficient granularity for analysis while avoiding ties that could arise with fewer levels (e.g., four). The selected encoder models are DeBERTaV3 (He et al., 2021), RoBERTa (Liu, 2019), BERT (Devlin et al., 2019) and DistilBERT (Sanh, 2019). We also use one decoder: GPT-2 (Radford et al., 2019). They were fine-tuned on separate general-purpose question answering data (the SQuAD v1.1 dataset (Rajpurkar et al., 2016)). The models were deliberately chosen for their varying performance levels, thereby simulating different levels of learner skill. Table 1 shows the performance of each QA system on the SQuAD v1.1 evaluation set, using the n -gram similarity metric ROUGE_L-F1 (Lin, 2004) (QA answer vs. SQuAD ground-truth answer).

²The arrow separates the input (left) and output (right) information. On the left part, the + symbol illustrates whether the method incorporates control attributes.

Table 1: Simulated-Learner QA systems performance on SQuAD v1.1 evaluation set.

Sim.-Learner QA	ROUGE _L -F1 (0-1)
DeBERTaV3 (large)	0.87
RoBERTa (base)	0.82
BERT (base)	0.75
DistilBERT (base)	0.69
GPT-2	0.46

4.3 Answering FairytaleQA Questions with QA Systems

For each question in the train and validation sets of the FairytaleQA dataset, all five simulated-learner QA systems generated their own answers. Each QA answer is then compared to the corresponding ground-truth answer to determine correctness. We considered an answer correct if it achieved either an exact match score of 1 or a ROUGE_L-F1 score of at least 0.5. The QA answers are organized into a binary response matrix — Figure 2 shows an example of such a matrix. Each row corresponds to a simulated-learner QA system and each column corresponds to a question ID. Each cell contains a 0 or 1, indicating incorrect or correct answers, respectively. This matrix serves as input data for the subsequent question difficulty estimation using IRT.

4.4 Estimating Question Difficulty with IRT

Based on the collected correct and incorrect answers for each question — organized into a binary response matrix — we estimated question difficulty using the Rasch Model (recall Section 2.2). Specifically, using the binary correctness data produced by the simulated-learner QA systems, the estimation is performed using the Expectation-Maximization (EM) algorithm (Embretson and Reise, 2000). This yielded difficulty values that were subsequently normalized to a 0-1 scale (0, 0.28, 0.50, 0.72, and 1), where higher values represent more difficult questions. The numerical values were converted into corresponding categorical labels – *easy*, *medium*, *moderate*, *hard*, and *extreme* – to be used in textual prompts. The distribution of the estimated difficulty values by narrative label in the data is presented in Table 2. Some attributes (e.g., *feeling* and *prediction*) have limited representation in the dataset.

Additionally, using the Maximum a Posteriori

Nar.	Easy	Med.	Mod.	Hard	Extr.
Action	773	362	375	435	749
Causal	316	200	245	316	1291
Char.	497	133	101	116	115
Feeling	55	79	62	89	539
Out.	126	114	138	165	268
Pred.	22	21	23	50	250
Setting	276	70	60	54	63
Action	76	40	65	60	92
Causal	35	27	31	50	151
Char.	50	17	14	9	17
Feeling	0	9	9	5	71
Out.	11	13	19	15	39
Pred.	1	3	6	7	38
Setting	29	4	5	4	3

Table 2: Difficulty values by Nar. (train and val set).

(MAP) algorithm (Embretson and Reise, 2000), we estimated the ability (θ) values for each QA system. These values are reported in Table 3, with higher values representing higher abilities. These values align, as expected, with the systems’ original performance levels shown in Table 1.

Sim.-Learner QA	Ability (θ)
DeBERTaV3 (large)	0.43
RoBERTa (base)	0
BERT (base)	-0.66
DistilBERT (base)	-1.25
GPT-2	-1.60

Table 3: Simulated-learner estimated ability values (θ) after answering questions from the FairytaleQA dataset.

We use *mirt*³ tool for IRT, including all estimations.

4.5 Creating a Question-Answer Pair Generation Model

We use the FLan-T5 (Chung et al., 2024) encoder-decoder model for the controllable task. This model builds upon the original T5 (Raffel et al., 2020), which has been fine-tuned with task-specific instructions using prefixes, making it well-suited for our methodology. Additionally, FLan-T5

³<https://cran.r-project.org/web/packages/mirt/index.html>

demonstrates remarkable performance in text generation tasks, particularly in QG (Chen et al., 2024; Li and Zhang, 2024). We employ the `flan-t5-large` version, which is publicly available via Hugging Face⁴. Training is conducted for up to 10 epochs, with early stopping implemented using a patience of 2 epochs. During inference, we apply Top-k sampling with $k = 50$, $p = 0.9$ and $temp = 1.2$ to encourage diversity (values obtained experimentally). We initially explored beam search, a widely used technique in QG; however, we observed that it frequently produced repetitive questions when tasked with generating questions for the same narrative element across different difficulty levels.

4.6 Generating QA Pairs for Evaluation

We fine-tune the `Flan-T5` model on the training set of `FairyTaleQA`. We obtain 4 models, as the model has been trained on each of the 4 data setups described in Section 4.1. For the 2 setups where difficulty labels are used, we apply the resulting models (inference) to the corresponding test set and generate 5 QA pairs for each text’s section — one QA pair for each difficulty label. Since the `FairyTaleQA` test set contains 394 section texts, we obtain a total of 1,970 generated QA pairs. Additionally, each text includes human-authored QA pairs associated with different narrative labels. This approach ensures that the generated QA pairs are balanced across distinct difficulty levels and narrative elements for further evaluation.

5 Evaluation

5.1 Evaluation Procedure

For NCQG, our evaluation protocol follows prior studies (Zhao et al., 2022; Leite and Lopes Cardoso, 2023, 2024) that focused on controlled generation using narrative labels. For DCQG, the evaluation protocol is based on recent works (Uto et al., 2023; Tomikawa et al., 2024; Tomikawa and Uto, 2024) that emphasize the use of simulated-learner QA systems across generated questions with distinct difficulty levels.

Narrative Control: To assess narrative control, we use a standard approach in QG: comparing generated questions directly with human-authored ground-truth questions. Hypothesis 1 (H1) is that *incorporating narrative attributes will result in generated questions that are more similar to the*

⁴<https://huggingface.co/google/flan-t5-large>

ground-truth, as previously shown by Leite and Lopes Cardoso (2024). To quantify the similarity, we employ the n -gram similarity metric `ROUGEL-F1` (Lin, 2004), as originally adopted by the `FairyTaleQA` authors. For a better perception of the idea, consider the human-authored ground-truth question: “What did Matte and Maie do on Saturdays?” (annotated with the *action* narrative element) and the generated question targeting the same narrative element: “What did Maie and Matte do to provide for themselves?”. These questions yield a high `ROUGEL-F1` score because they are similar in terms of the narrative-related vocabulary they share, thus indicating successful narrative control.

Difficulty Control: For difficulty control, the evaluation focuses on analyzing the performance of simulated-learner QA systems when answering questions generated at varying difficulty levels. Hypothesis 2 (H2) posits that *simulated-learner QA systems will perform better on easier questions and worse on more difficult ones, relative to their ability levels*.

5.2 Results

Narrative Control: Table 4 presents the results from the narrative control perspective, measured using `ROUGEL-F1` n -gram similarity between the generated questions and the human-authored ground-truth questions. We observe an improvement in the similarity to ground-truth questions when narrative control attributes are incorporated. This trend is consistently observed across all seven narrative labels. Furthermore, these findings align with the results reported in prior studies on narrative control (Leite and Lopes Cardoso, 2023, 2024). Of novelty, when narrative and difficulty labels are fused, we observe a similar improvement trend, comparable to the incorporation of narrative attributes alone. These results support Hypothesis 1 (H1), indicating that our method effectively controls the narrative elements underlying the generated questions. Appendix A shows further support by reporting semantic similarity results.

Difficulty Control: Figure 3 presents the results for difficulty control only, showing the percentage of correct responses from the simulated-learner QA systems across all difficulty levels. The percentage of correct answers decreases as the difficulty level increases for all simulated learners⁵. Additionally,

⁵All percentages are relatively low (<60). This is because the QA models were not trained on the `FairyTaleQA` dataset but were instead trained on `SQuAD`. This intentional choice

Data Setup	Char.	Setting	Action	Feeling	Causal	Out.	Pred.
Text \rightarrow QA	.227	.269	.287	.281	.271	.227	.251
Nar + Text \rightarrow QA	.304	.537	.427	.527	.412	.458	.348
Nar + Dif + Text \rightarrow QA	.305	.530	.412	.529	.405	.425	.365

Table 4: **Narrative Control:** Similarity (ROUGE_L-F1) between generated and ground-truth questions on the test set by narrative element. **Text \rightarrow QA** is used as a baseline to assess whether narrative control helps the generated questions approximate the ground-truth questions.

learners with higher abilities achieve higher percentages of correct answers, while those with lower abilities achieve lower percentages. These findings are consistent with previous works (Uto et al., 2023; Tomikawa et al., 2024) and support Hypothesis 2 (H2), demonstrating that the method controls the difficulty levels of the generated questions.

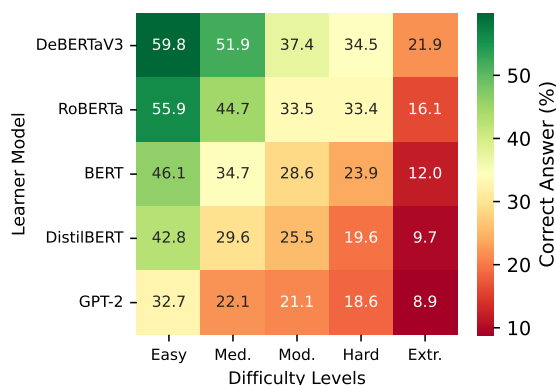


Figure 3: Percentage (%) of correct answers by difficulty level when only difficulty control labels are used (**Dif + Text \rightarrow QA**).

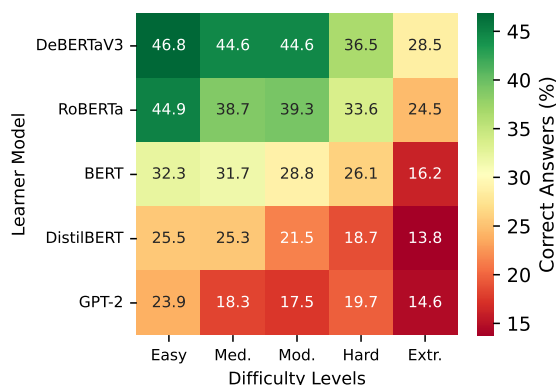


Figure 4: Percentage (%) of correct answers by difficulty level when both difficulty and narrative control labels are used (**Nar + Dif + Text \rightarrow QA**).

ensures that the models’ knowledge remains unbiased with respect to FairyTaleQA content.

Joint Narrative and Difficulty Control: Figure 4 presents the results for difficulty control when difficulty and narrative attributes are fused. In most cases, the percentage of correct answers decreases as the difficulty level increases across all simulated learners. These findings demonstrate that even when conditioning the generation process on both narrative content and difficulty, it remains possible to perform difficulty control. However, some inconsistencies are observed: for DeBERTaV3, there is no distinction between medium and moderate difficulty levels; for RoBERTa, the percentage of correct answers increases between medium and moderate levels; and for GPT-2, a similar trend occurs between moderate and hard levels. For an overall graphical comparison of difficulty control using only difficulty versus combining difficulty and narrative attributes, see Appendix B.

Figure 5 shows the overall accuracy for each narrative label, with trends suggesting difficulty control particularly between easy, hard, and extreme levels. However, control becomes inconsistent at intermediate levels. Among the attributes, *causal* and *outcome* demonstrate the most consistent control across difficulty levels, while *prediction* and *feeling* exhibit the least success. This inconsistency can be related to the limited representation of these attributes in the FairyTaleQA dataset (recall Table 2), which prevents the model from learning to generate questions across different difficulty levels. Additionally, questions tied to these attributes are inherently more challenging, as reflected in the lower global performance of simulated-learner QA systems. For attributes such as *character*, *prediction*, *action*, and *setting*, the confusion is particularly evident between medium and moderate levels. To address this, we experimented with an alternative model trained on a lower granularity of difficulty levels, combining medium, moderate, and hard into a single medium level. In Figure 6, we show the result of this experiment, which demonstrates more consistent control across all levels.

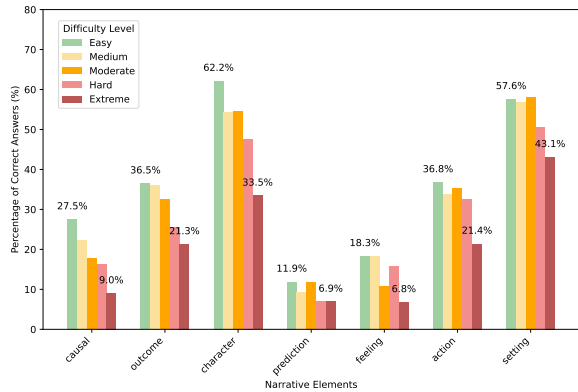


Figure 5: Percentage (%) of correct answers per narrative element and difficulty level (5 levels).

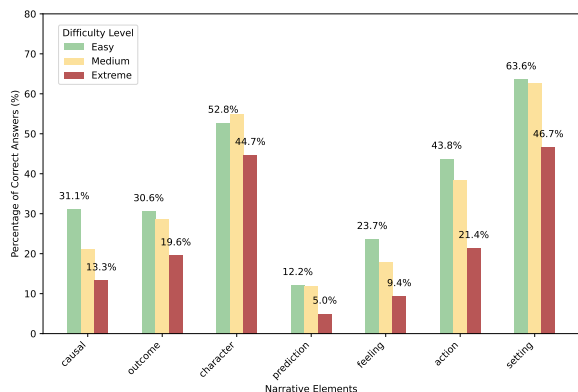


Figure 6: Percentage (%) of correct answers per narrative element and difficulty level (3 levels).

However, the *character* and *prediction* attributes continue to reveal some difficulty in distinguishing levels. These results support Hypothesis 2 (H2), confirming that the joint method enables difficulty control, although with less consistency than when controlling for difficulty alone. In Section 6, we outline potential explanations for these results.

Linguistic Features Influenced By Control: To better understand the linguistic features influenced by the controllability method, we analyze the linguistic properties of the generated QA pairs across different difficulty levels and narrative attributes. Prior work on difficulty-only controlled generation (Tomikawa et al., 2024) identifies two key factors that distinguish difficulty levels: (1) the average number of words in the generated answers, and (2) the distribution of initial interrogative terms in the generated questions. While we also explore these features (see Appendix C), we emphasize here a novel aspect that we also found experimentally to be relevant: (3) the degree of lexical novelty in the generated QA pairs relative to the source

narrative text. To quantify this, we use the PINC (Paraphrase In N-gram Changes) metric (Chen and Dolan, 2011), which computes the percentage of n -grams present in the generated QA pairs but not in the source text. Higher PINC scores indicate greater lexical novelty and diversity. The results in Table 5 show that the diversity of the QA pairs increases with higher difficulty levels. This trend is observed both when difficulty labels are used independently and when combined with narrative labels. Therefore, we conclude that the linguistic diversity between the generated QA pairs and the source text is a feature influenced by difficulty control, regardless of whether difficulty labels are used alone or in conjunction with narrative labels.

Data Setup		Easy	Med.	Extr.
Dif + Text	Q	55.60	60.23	63.94
	→ QA	A 9.88	23.17	48.69
Nar + Dif + Text	Q	57.34	60.72	65.57
	→ QA	A 22.02	26.00	41.14

Table 5: PINC values (%) considering 3-gram novelty for generated questions (Q) and answers (A).

Error Analysis of Generated QA Pairs: We randomly selected 100 generated QA pairs and analyzed potential problems. Two main issues were identified, as shown in Figure 7: (1) *hallucinated content* (14%) and (2) *nonsensical QA pairs* (10%).

In the first example (1), the generated QA pair introduces made-up terms that do not exist in the source text. This is the most frequent problem encountered, likely due to the nature of the inference method (Top-k sampling). In example 2a, the problem is that the generated QA pair is misleading. In Example 2b, the question seeks a motivation not in the text, and the vague answer just reuses the question’s wording without adding meaning.

6 Discussion

We revisit our research question (RQ): *How effectively can we control the generation of question-answer pairs conditioned on both narrative and difficulty attributes using a modest scale model?*

Our findings indicate that the model effectively controls narrative elements, whether alone or combined with difficulty attributes. It also shows effective difficulty control when conditioned solely on difficulty attributes. The joint model, which combines narrative and difficulty attributes, generally achieves consistent control for at least three levels

<p>Example 1 (hallucinated content):</p> <p>Text: And then he cried for the third time: “Old Hoggiant!” And there appeared a fellow of truly horrible size, with a scythe as large as a ship’s mast.</p> <p>Gen. Q: Who appeared to the hunter at the swan?</p> <p>Gen. A: A fellow of truly horrible size, with a scythe as large as a ship’s mast.</p> <p>Explanation: There is no mention of any hunter or swan in the source text.</p>
<p>Example 2a (nonsensical QA pairs):</p> <p>Text: When the old man had finished his dance, the big demon said: “Thank you very much for your amusing dance. Now give us the pleasure of drinking a cup of wine with us,” and with these words he handed him his largest wine-cup.</p> <p>Gen. Q: What did the old man do after he finished dancing?</p> <p>Gen. A: Danced with the big demon.</p> <p>Explanation: The old man had finished dancing and did not dance with anyone again.</p>
<p>Example 2b (nonsensical QA pairs):</p> <p>Text: But the rich neighbor wanted the meadow for himself alone, and told the poor one that he would drive him out of house and home if he did not come to an agreement with him that whichever one of them mowed the largest stretch of the meadowland in a single day, should receive the entire meadow.</p> <p>Gen. Q: Why did the neighbor want to own the meadow?</p> <p>Gen. A: He wanted to have the meadow.</p> <p>Explanation: The question contains the answer.</p>

Figure 7: Examples of problematic generated question-answer pairs (error analysis).

(easy, hard, and extreme). However, inconsistencies arise in the intermediate levels (medium and moderate). We also observed that certain attributes are more conducive to effective control, while others, like *prediction* and *feeling*, are less effective. Notably, reducing the granularity of difficulty levels improves the overall control. We now delve into two main factors that underlie our findings.

First, *generating QA pairs while simultaneously controlling both difficulty and narrative attributes is an inherently challenging task*. When the narrative element is fixed, the space of plausible questions becomes more constrained. This makes it harder to vary difficulty meaningfully, as the questions tend to focus on similar content. For instance, in Figure 1, the last two questions share the same narrative element but differ in difficulty. This overlap in content makes it harder to generate questions with clearly distinct difficulty levels.

Second, *some narrative attributes naturally lead to easier questions*. For instance, the *character* attribute often involves straightforward “Who” questions, making it harder to create questions with distinct difficulty levels. In contrast, questions following the *prediction* attribute are demanding, adding

complexity to the learning process of generating well-differentiated questions.

Transferability to other domains: While our current work focuses on narrative comprehension, the principles of controllable QG are not domain-specific. For instance, it would be feasible to control generation based on other reading comprehension skills, as explored by Ghanem et al. (2022). Progress in this direction depends on the availability of datasets annotated with these dimensions, which are scarce.

Relevance to education: We believe our findings hold promise for educational applications, particularly in personalized QG. Recent work has explored adapting QG to student ability (Tomikawa et al., 2024). We argue that incorporating narrative control adds another valuable layer to personalization, enabling more targeted and contextually rich QG.

7 Conclusions

This work investigates a strategy for controlling both narrative and difficulty attributes in generated QA pairs. The results offer a preliminary yet promising demonstration of the potential of QG models and the proposed control strategy. Future efforts could leverage larger datasets with a more balanced distribution of questions across categories to improve the model’s control capabilities. Additionally, examining the impact of different inference methods on generation would be valuable, especially to address the issue of repetitive outputs observed with beam search. Finally, future research could explore few-shot prompting techniques, providing minimal examples to assess the model’s control ability without extensive training.

Limitations

While our approach provides promising insights into controllable QG, some limitations should be acknowledged.

First, *the limited representation of question categories across narrative attributes and difficulty levels hinders the model’s ability to learn effectively*. FairytaleQA consists of approximately 10k instances. Associating questions with multiple narrative elements and difficulty levels significantly reduces the number of examples per category, limiting the model’s ability to learn effectively. For instance, as shown previously in Table 2, *prediction* and *feeling* questions are poorly represented.

Second, *top-k sampling enables control over narrative elements and question difficulty but can lead to undesired hallucinations*. Initially, we experimented with beam search — a more commonly used technique for QG — but found it often generated repetitive questions when addressing the same narrative element across varying difficulty levels. Moreover, our findings indicate that the choice of inference method significantly impacts control. For instance, as shown in Section 5.2, the diversity of the generated QA pairs increases at higher difficulty levels. However, this diversity can also produce unintended side effects, such as the hallucinations noted with error analysis. While hallucinated QA pairs may affect evaluation by inflating perceived difficulty, we believe that reporting such cases was important to reveal potential failure modes of controllable QG systems. Although they may add some noise, these observations help contextualize the results and guide future improvements in model robustness.

Third, the *evaluation relies on simulated learner responses rather than real student data*. While this approach offers scalability and approximations of question difficulty, it may not fully reflect how actual students would respond. Nonetheless, it provides a valuable proxy for assessing the model’s behavior, and we believe it still offers meaningful insight into the controllability of QG systems. Future work should explore incorporating real student data to further validate these findings.

Ethics Statement

This research involves the automatic generation of QA pairs from narrative texts, incorporating control attributes such as difficulty level and narrative elements. The dataset used, FairytaleQA, consists of human-authored QA pairs from publicly available fairy tales. No personally identifiable or sensitive information is included, ensuring compliance with ethical guidelines for data usage. The generated QA pairs were evaluated using both automatic metrics and manual inspection to identify potential errors, such as hallucinated content and nonsensical questions. We acknowledge that these models may introduce unintended errors or biases. While this paper does not focus on error mitigation, future work could explore extended human-in-the-loop validation to enhance the reliability of generated QA pairs, particularly in deployment scenarios.

Acknowledgments

The authors would like to thank Professor Masaki Uto and Yuto Tomikawa for their helpful clarifications and discussions related to prior work. This work was financially supported by UID/00027 – the Artificial Intelligence and Computer Science Laboratory (LIACC), funded by Fundação para a Ciência e a Tecnologia, I.P./ MCTES through national funds. Bernardo Leite is supported by a PhD studentship (with reference 2021.05432.BD), funded by FCT.

References

- Julie Alonzo, Deni Basaraba, Gerald Tindal, and Ronald S Carriveau. 2009. They read, but how well do they understand? an empirical look at the nuances of measuring reading comprehension. *Assessment for Effective Intervention*, 35(1):34–44.
- Sheng Bi, Xiya Cheng, Yuan-Fang Li, Lizhen Qu, Shirong Shen, Guilin Qi, Lu Pan, and Yinlin Jiang. 2021. Simple or complex? complexity-controllable question generation with soft templates and deep mixture of experts model. *arXiv preprint arXiv:2110.06560*.
- David Chen and William Dolan. 2011. [Collecting highly parallel data for paraphrase evaluation](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA. Association for Computational Linguistics.
- Jiaju Chen, Yuxuan Lu, Shao Zhang, Bingsheng Yao, Yuanzhe Dong, Ying Xu, Yunyao Li, Qianwen Wang, Dakuo Wang, and Yuling Sun. 2024. [StorySparkQA: Expert-annotated QA pairs with real-world knowledge for children’s story-based learning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17351–17370, Miami, Florida, USA. Association for Computational Linguistics.
- Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. [Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting](#). In *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 5968–5978, Online. Association for 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.

- 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.
- Sabina Elkins, Ekaterina Kochmar, Iulian Serban, and Jackie C. K. Cheung. 2023. How useful are educational questions generated by large language models? In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pages 536–542, Cham. Springer Nature Switzerland.
- SE Embretson and SP Reise. 2000. Item response theory for psychologists. *Lawrence Earlbaum Associates, Mahwah, NJ*.
- Yifan Gao, Lidong Bing, Wang Chen, Michael Lyu, and Irwin King. 2019. [Difficulty controllable generation of reading comprehension questions](#). In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4968–4974. International Joint Conferences on Artificial Intelligence Organization.
- Bilal Ghanem, Lauren Lutz Coleman, Julia Rivard Dexter, Spencer von der Ohe, and Alona Fyshe. 2022. [Question generation for reading comprehension assessment by modeling how and what to ask](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2131–2146, Dublin, Ireland. Association for Computational Linguistics.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). *arXiv preprint arXiv:2111.09543*.
- Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. [The NarrativeQA reading comprehension challenge](#). *Transactions of the Association for Computational Linguistics*, 6:317–328.
- David R. Krathwohl. 2002. [A revision of bloom’s taxonomy: An overview](#). *Theory Into Practice*, 41(4):212–218.
- Vishwajeet Kumar, Yuncheng Hua, Ganesh Ramakrishnan, Guilin Qi, Lianli Gao, and Yuan-Fang Li. 2019. [Difficulty-controllable multi-hop question generation from knowledge graphs](#). In *The Semantic Web – ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I*, page 382–398, Berlin, Heidelberg. Springer-Verlag.
- Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204.
- Bernardo Leite and Henrique Lopes Cardoso. 2023. Towards enriched controllability for educational question generation. In *Artificial Intelligence in Education*, pages 786–791, Cham. Springer Nature Switzerland.
- Bernardo Leite and Henrique Lopes Cardoso. 2024. [On few-shot prompting for controllable question-answer generation in narrative comprehension](#). In *Proceedings of the 16th International Conference on Computer Supported Education - Volume 2: CSEDU*, pages 63–74. INSTICC, SciTePress.
- Kunze Li and Yu Zhang. 2024. [Planning first, question second: An LLM-guided method for controllable question generation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4715–4729, Bangkok, Thailand. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A Package for Automatic Evaluation of Summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. ACL.
- Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.
- Frederic M Lord. 2012. *Applications of item response theory to practical testing problems*. Routledge.
- Julie S Lynch, Paul Van Den Broek, Kathleen E Kremer, Panayiota Kendeou, Mary Jane White, and Elizabeth P Lorch. 2008. The development of narrative comprehension and its relation to other early reading skills. *Reading Psychology*, 29(4):327–365.
- Alison H Paris and Scott G Paris. 2003. Assessing narrative comprehension in young children. *Reading Research Quarterly*, 38(1):36–76.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

- Vasile Rus, Zhiqiang Cai, and Art Graesser. 2008. Question generation: Example of a multi-year evaluation campaign. *Proc WS on the Question Generation Shared Task and Evaluation Challenge*.
- V Sanh. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Yuto Tomikawa, Ayaka Suzuki, and Masaki Uto. 2024. Adaptive question–answer generation with difficulty control using item response theory and pretrained transformer models. *IEEE Transactions on Learning Technologies*, 17:2240–2252.
- Yuto Tomikawa and Masaki Uto. 2024. Difficulty-controllable multiple-choice question generation for reading comprehension using item response theory. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pages 312–320, Cham. Springer Nature Switzerland.
- Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2022. Generative language models for paragraph-level question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 670–688, Abu Dhabi, United Arab Emirates. 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.
- Xu Wang, Simin Fan, Jessica Houghton, and Lu Wang. 2022a. Towards process-oriented, modular, and versatile question generation that meets educational needs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 291–302, Seattle, United States. Association for Computational Linguistics.
- Zichao Wang, Jakob Valdez, Debshila Basu Mallick, and Richard G. Baraniuk. 2022b. Towards human-like educational question generation with large language models. In *Artificial Intelligence in Education*, pages 153–166, Cham. Springer International Publishing.
- Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. 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.*, 40(1).
- Sanqiang Zhao, Seokhwan Kim, Yang Liu, Robinson Piramuthu, and Dilek Hakkani-Tür. 2023. Storyqa: Story grounded question answering dataset.
- Zhenjie Zhao, Yufang Hou, Dakuo Wang, Mo Yu, Chengzhong Liu, and Xiaojuan Ma. 2022. Educational question generation of children storybooks via question type distribution learning and event-centric summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5073–5085, Dublin, Ireland.

A Narrative Control: Semantic Similarity

Table 6 presents the results from the narrative control perspective, measured using BLEURT (Sellam et al., 2020). The goal is to show an improvement in semantic similarity to ground-truth questions when narrative control attributes are incorporated. As observed with ROUGE_L-F1 similarity (recall Section 5.2), this trend is observed across all seven narrative labels. When narrative and difficulty labels are fused, we observe a similar improvement trend, comparable to the incorporation of narrative attributes alone. These results further support Hypothesis 1 (H1) — *incorporating narrative attributes will result in generated questions that are more similar to the ground-truth* — indicating that our method controls the narrative elements underlying the generated questions.

B Difficulty-Only vs. Difficulty+Narrative Control

To compare difficulty control when operating solely on difficulty versus combining difficulty and narrative attributes, Figure 8 provides an overview of the performance at each level for both setups. Both setups show the expected trend: the percentage of correct answers decreases as difficulty increases. However, a linear approximation of the observed data points reveals that the decrease is less pronounced when both attributes are combined, though it remains consistent overall.

Data Setup	Char.	Setting	Action	Feeling	Causal	Out.	Pred.
Text → QA	.332	.332	.353	.370	.360	.346	.358
Nar + Text → QA	.379	.504	.422	.491	.418	.444	.409
Nar + Dif + Text → QA	.378	.482	.413	.499	.417	.422	.401

Table 6: **Narrative Control**: Semantic similarity (BLEURT) between generated and ground-truth questions on the test set by narrative element. **Text** → **QA** is used as a baseline to assess whether narrative control helps the generated questions approximate the ground-truth questions.

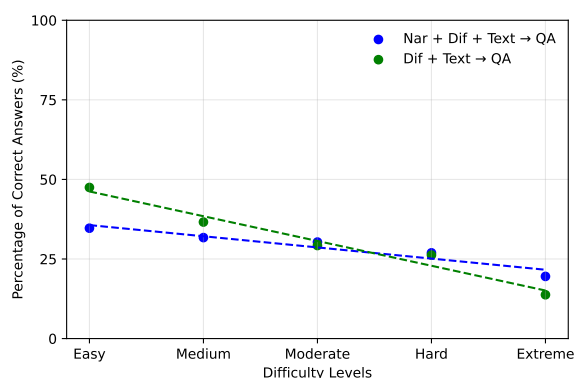


Figure 8: Percentage of Correct Answers by Dif. Level.

C Additional Linguistic Features Influenced By Control

Table 7 presents the average number of words in the generated question-answer pairs. For generated answers, when only difficulty labels are incorporated, no significant trend is observed. For generated questions, an upward trend is noted, though it is not significant. When narrative and difficulty labels are combined, no trend is observed. Based on these findings, we conclude that the average length of generated question-answer pairs is not influenced by difficulty or narrative control labels in our experiments.

Data Setup		Easy	Med.	Extr.
Dif + Text	Q	10.80	11.83	12.49
	A	7.19	8.95	8.88
Nar + Dif + Text	Q	11.81	11.62	11.70
	A	7.42	7.96	7.61

Table 7: Average number of words for generated questions (Q) and answers (A).

Figure 9 illustrates the proportion of initial interrogative terms in the generated questions. When only difficulty labels are used (top chart), higher difficulty levels show an increase in terms like “why” and “how” and a decrease in terms like

“what” “who” and “where”. This aligns with expectations, as “why” and “how” are often linked to questions requiring higher cognitive effort, as described in Bloom’s taxonomy (Krathwohl, 2002). When both narrative and difficulty labels are fused (lower chart), the proportion of all interrogative terms is more consistent across difficulty levels. This outcome is expected since this setup aims to control difficulty levels while also demanding for certain narrative elements. In this case, narrative labels are the primary influence for the choice of interrogative terms (e.g., “who” for character-related questions), rather than difficulty labels.

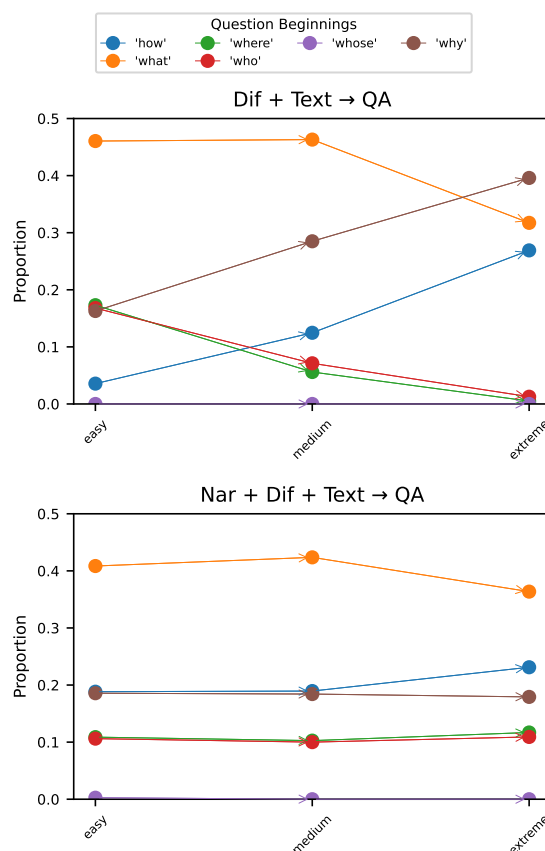


Figure 9: Proportion of initial interrogative terms in the generated questions (arrowed lines indicate increase/decrease trends).

Down the Cascades of Omethi: Hierarchical Automatic Scoring in Large-Scale Assessments

Fabian Zehner^{1,2}, Hyo Jeong Shin³, Emily Kerzabi⁴, Andrea Horbach⁵,
Sebastian Gombert¹, Frank Goldhammer^{1,2}, Torsten Zesch⁶, Nico Andersen¹

¹DIPF, Germany, {f.zehner, n.andersen}@dipf.de

²Centre for International Student Assessment (ZIB), Germany

³Sogang University, South Korea; ⁴Educational Testing Service, USA

⁵IPN, Germany; ⁶FernUniversität in Hagen, Germany

Abstract

We present the framework Omethi, which is aimed at scoring short text responses in a semi-automatic fashion, particularly fit to international large-scale assessments. We evaluate its effectiveness for the massively multilingual PISA tests. Responses are passed through a conditional flow of hierarchically combined scoring components to assign a score. Once a score is assigned, hierarchically lower components are discarded. Models implemented in this study ranged from lexical matching of normalized texts—with excellent accuracy but weak generalizability—to fine-tuned large language models—with lower accuracy but high generalizability. If not scored by any automatic component, responses are passed on to manual scoring. The paper is the first to provide an evaluation of automatic scoring on multilingual PISA data in eleven languages (including Arabic, Finnish, Hebrew, and Kazakh) from three domains ($n = 3.8$ million responses). On average, results show a manual effort reduction of 71 percent alongside an agreement of $\kappa = .957$, when including manual scoring, and $\kappa = .804$ for only the automatically scored responses. The evaluation underscores the framework's effective adaptivity and operational feasibility with its shares of used components varying substantially across domains and languages while maintaining homogeneously high accuracy.

1 Introduction

A river adapts its flow to diverse exterior conditions, by meandering, or alternating its velocity and depth, to reach its target inevitably and naturally. In this paper, we propose the hierarchical, response-adaptive framework *Omethi* for automatically scoring short text responses from assessments. The proposed framework is named after the Omethi River, for it is similarly responsive by combining modern and baseline scoring methodology adaptively at the response level, while contending

with diverse languages and multiple assessment domains in an operational setting and distinct quality requirements. Large-scale assessments, especially international ones (e.g., PISA, the *Programme for International Student Assessment*; OECD, 2023), pose diverse conditions to automatic scoring (Zesch et al., 2023), similar to the varied surroundings a river is exposed to. In turn, automatic scoring encompasses a range of approaches with particular strengths and weaknesses (see Galhardi and Brancher, 2018; Gao et al., 2024).

Accordingly, the paper provides three major contributions. First, we present a novel hierarchical composition of models for automatically scoring short text responses, particularly fit to the complex settings present in large-scale assessments.

Second, for a first implementation of the framework, we propose a hierarchical collection of models, including a new rigorous method with weak generalizability, called *Fuzzy Lexical Matching* (FLM), alongside fine-tuned XLM-RoBERTa (XLM-R; Conneau et al., 2020) and support vector machine classifiers (SVM; Cortes and Vapnik, 1995). Human raters, integral to assessment operations, serve as the final component in the sequence of scoring methods presented here, turning the implemented pipeline into a semi-automatic system.

Third, this is the first paper to evaluate automatic scoring on massively multilingual data from PISA tests including all three major domains (i.e., *reading*, *mathematics*, and *science*; OECD, 2024). With the complete dataset containing 59 test languages from 86 countries and regions in total, we sampled a subset of 11 test languages for the present evaluation, resulting in about 3.8 million text responses to 160 items from 3 assessment domains and more than 270,000 students. To represent diverse language families and writing systems, the selected test languages included Arabic, Finnish, Hebrew, Kazakh, and Korean, among others.

The empirical evaluation was guided by two

overarching research questions. (I) Overall and for each subcomponent, how effective is the model at generating accurate scores and reducing manual effort? (II) How robust are scoring accuracy and reduced manual effort across subsamples with different test languages?

2 Background

2.1 Relevance for Operational Assessments

International educational large-scale assessments, such as PISA, are characterized by their large scope in addressing diverse student characteristics from different cultures using complex methodology. This can pose significant challenges for automatic scoring (Yan et al., 2020; Zesch et al., 2023). The resulting diversity manifests in response texts and corresponding scoring, stemming from many factors, including the world-wide participation (i.e., over ninety countries and economies in PISA 2025; OECD, 2025). The tests are administered in a large number of test languages (almost sixty test languages from 2018 to 2022), with high-resource languages, such as Indonesian, just as low-resource languages, such as Kazakh or Catalan. Moreover, the tests assess three major literacy domains, using a large number of items and various item types with complex coding guides for constructed-response formats. Additionally, the low-stakes nature at the individual level often results in lower test engagement (Schlosser et al., 2019) and, thus, more informal, fragmented, and less integrated (Chafe, 1982) text responses. Continuous changes in assessment design—such as the transition from paper- to computer-based testing and the adoption of adaptive testing—introduce additional variability over time; for example, by reducing the number of responses per item (OECD, 2024) or by impacting the length and quality of text responses (Zehner et al., 2019, 2020). On top of this, not only sample sizes vary largely per test language (e.g., from $n = 269$ to $n = 22,163$ responses per item in the present paper’s reported dataset), which poses challenges for training, but also a reduced rigor in human coding can lead to more label noise in subsamples. At the same time, large-scale assessments pose incontestable quality requirements (see OECD, 2025), including high-quality coding and accountability (i.e., explainability), due to their high stakes at the state level. Shin et al. (2019) demonstrated that automatic scoring can align closely with human experts in identifying rater severity, and less so

regarding centrality and accuracy, highlighting further challenges in introducing automatic systems in operational procedures. Noteworthy, large-scale assessments usually administer a subset of items repeatedly over time, making them an attractive field of application for supervised learning.

Thus far, automatic scoring has seen limited research and operational use in international large-scale assessments. Early efforts include the introduction of PISA’s Machine-Supported Coding System (Yamamoto et al., 2018), a precursor to FLM, and a baseline evaluation for German (Zehner et al., 2016). Recent research funded by international bodies, such as on IEA’s ePIRLS data (*International Association for the Evaluation of Educational Achievement*; Shin et al., 2024), and a competition on data from the National NAEP (*National Assessment of Educational Progress*; Whitmer et al., 2023) signal growing interest in automating scoring, notoriously centering around national U.S. assessments (Yan et al., 2020).

2.2 Diverse Models to Address Text Diversity

All these extraneous factors manifest in varying degrees of linguistic variance in text responses (Zesch et al., 2023; Horbach and Zesch, 2019) across cohorts, subpopulations (i.e., languages), domains, items, and their context. Single automatic scoring approaches can fall short of adequately addressing this diversity. For instance, while lexical matching methods offer excellent accuracy for known responses, they lack generalizability to unseen linguistic expressions. Moreover, supervised classifiers are often hampered as they assign a label regardless of relatively low probabilities (i.e., confidence) for certain instances (Li et al., 2023).

Recognizing these limitations, the here presented first collection of implemented components in an Omethi framework retain human raters as the final recourse when automatic models fail to score responses with sufficient confidence, rendering it a semi-automatic system. By hierarchically composing multiple scoring approaches and discarding lower-level components once a score is confidently assigned, Omethi navigates the complexities of international large-scale assessments while maintaining the high-quality standards required for them.

2.3 Ensembles for (Semi-)Automatic Scoring

Ensembles for automatic and semi-automatic scoring come in two fashions: algorithmic ensembles that inherently comprise multiple models (e.g., ran-

dom forests) or combinations of relatively loosely coupled models (e.g., stacking). Omethi belongs to the latter and diverges from traditional systems by combining multiple components, including supervised classifiers, in a conceptually governed, top-down manner rather than relying on data-driven, bottom-up learning. Unlike the common paradigm of identifying a single optimal model for a dataset, task, or domain, Omethi deliberately alternates models at the response level based on explicit criteria. This approach contrasts with standard ensembles, such as those in Goenka et al. (2020) and Ormerod (2022), where model selection is carried out uniformly (e.g., majority voting or averaging) or with ensembles designed to capture diverse response characteristics (e.g., Mohler et al., 2011; Sahu and Bhowmick, 2020; Sakaguchi et al., 2015; Zhang et al., 2022). For instance, Heilman and Madnani (2013) stacked models for item-specific n -gram features and text similarity, while Roy et al. (2016) employed transfer learning between general and question-specific classifiers.

If humans are still involved during inference, the scoring is considered semi-automatic. For systems deferring responses to humans, appropriate confidence thresholds of the automatic component need to be identified; referred to as *deferral policy* in (Li et al., 2023), which we rephrase here as the *eligibility policy* for assigning a score. This has been investigated for semi-automatic systems, such as in Andersen et al. (2023) and Horbach et al. (2014), which combined unsupervised clustering with human scoring. Horbach and Pinkal (2018) more directly integrated humans and machines via semi-supervised clustering. In the context of label probabilities as a confidence criterion, results on identifying optimal confidence thresholds have been mixed. Suen et al. (2023) successfully set thresholds based on a minimum required F_1 score, while Bexte et al. (2024) observed substantial item- and data-wise variation in confidence distributions with this, failing to identify viable thresholds for certain items at all. Funayama et al. (2022) similarly used confidence scores to revert to human raters, and Li et al. (2025) proposed a constant threshold of $\delta = .25$, basically halving the range of values from random chance to perfect agreement.

3 Omethi Framework

Unlike traditional ensemble methods, Omethi orchestrates scoring components hierarchically based

on their conceptual priority, compiling a logical decision flow that is informed by each component’s inherent characteristics. If a component is eligible by satisfying its specific conditions (i.e., its eligibility policy), it assigns the final score and the response bypasses subsequent components.

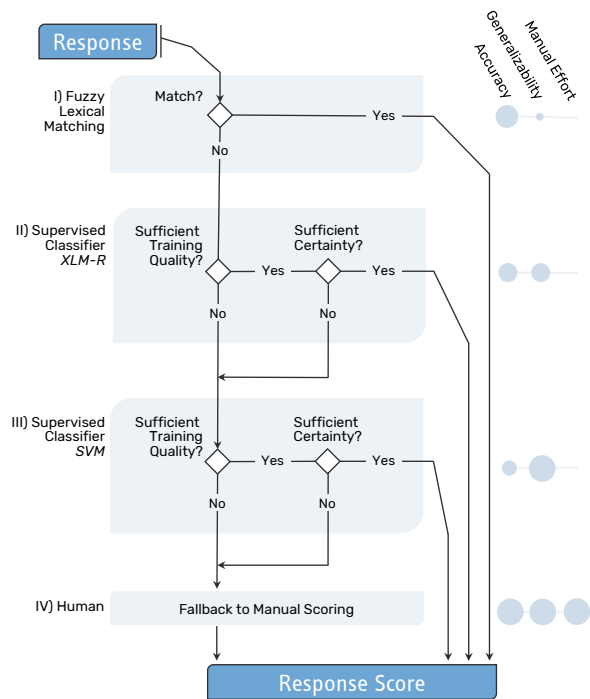


Figure 1: Response flow through the implemented Omethi pipeline

In this paper, we present an initial implementation consisting of four components, described in the following (see Figure 1). The rationale underlying the implementation was to allocate components with the highest accuracy prior to those exhibiting higher generalizability, while also aiming at minimizing human effort resulting from responses deferred to the final component.

During training, each scoring component was built separately using data from a given subsample; that is, for (i) a specific item and (ii) test language. Human scores available from the operational PISA studies served as the ground truth for training. During inference, each response was first evaluated by Fuzzy Lexical Matching (FLM), which attempts to match normalized text to a pool of normalized response texts. If sufficient matching responses were identified that satisfied predefined score-homogeneity criteria (see next section), propagating their score to the unseen response is considered highly reliable, as the response had been

scored multiple times previously by humans, or at least a lexically very close counterpart. FLM, therefore, receives the highest priority in the flow because its classifications are largely valid, interpretable, and applicable for any language. However, FLM’s obvious downside is its lack of out-of-sample generalization, the severity of which depends on the item-specific linguistic variance in the responses. Thus, if FLM *could* provided a score, that score *was* adopted, bypassing subsequent components. Otherwise, the response proceeded to the next scoring component.

Responses not scored by FLM were next passed to a supervised classifier: a fine-tuned XLM-RoBERTa classifier or SVM. The model’s output was assigned to the response if its overall training performance quality sufficed and the individual classification’s confidence exceeded an item- and language-specific threshold.

If none of the automatic components satisfied their eligibility policy, the response was forwarded to the final component, namely manual scoring by human raters.

3.1 Fuzzy Lexical Matching

FLM extends the idea of PISA’s Machine Support Coding System (Yamamoto et al., 2018), operationally introduced in PISA 2018. There, strict exact string matching was applied, automatically propagating scores if at least five homogeneously scored text responses were found in legacy data.

FLM builds on this widely adoptable principle of matching unseen to historic data. In contrast to exact matching, FLM first normalizes the texts by traditional preprocessing techniques. The normalization pipeline was first evaluated on ePIRLS data (the *Progress in International Reading Literacy Study*; Shin et al., 2024). The standard techniques used were white-space trimming, punctuation removal, case insensitivity, diacritics removal, stemming, stop word removal, and bag of words.

For optimization to a subsample (i.e., item and language), this set of normalization techniques is trained on the respective data. That is, the effectiveness of each pipeline step is evaluated using the coefficient ER (Effort Reduction), simply constituting the share of matched responses, $ER = \frac{n_m}{n_t}$; n_t denoting the total number of responses in the data and n_m the number of matches. Importantly though, FLM’s scoring quality also manifests in ER because the method requires sufficiently frequent as well as homogeneously scored responses

for automatic scoring. That is, if the grouping of the normalized texts leads to heterogeneous scores within that group, ER will decrease. A response is automatically scored if the following criteria are met. For a given response i , let m_i denote the number of its matches and s_i the number of responses that received the dominant score in the group. Then, the response is scored ($M = 1$) or not scored automatically ($M = 0$) as follows:

$$M = \begin{cases} 1, & \text{if } m_i \geq 3 \text{ and} \\ & s_i \geq \max(\lceil m_i \cdot .92 \rceil, m_i - 5) \\ 0, & \text{otherwise} \end{cases}$$

That is, a response is scored automatically if at least 3 responses are matched, requiring a minimum of 92 percent of homogeneous scores, but limited to an absolute maximum of 5 deviant responses.¹

Whenever a pipeline step in FLM leads to a decrease in ER , the respective step is discarded for the specific subsample (i.e., item and language). For example, if respondents were asked to provide an email address from a text, applying punctuation removal on the responses eliminates relevant information, leading to heterogeneously scored matching groups, a reduced ER , and, hence, this normalization step would be discarded during inference.

Another adaptive step in FLM is the tailoring of stopword lists to the subsample. The rationale behind this is twofold. For one, stopword lists are language-specific and differ largely in their scope. Second, whether certain words are predictive for a response’s score depends on the item. Therefore, if an optimized stopword list leads to an increase in ER or an increase of the overall accuracy while ER remains identical, the optimized stopword list is used during inference.

3.2 Supervised Classifiers

Two types of classifiers based on supervised learning were built: fine-tuned XLM-RoBERTa models and support vector machines. During inference, both only take response texts as their input, not considering item stems, stimulus materials, or scoring guides.

As a core component, fine-tuned XLM-RoBERTa models (Conneau et al., 2020) were employed for their robust multilingual representation and classification capabilities. XLM-R is a massively multilingual model pretrained on a corpus

¹In PISA, the minimum inter-rater agreement is required to be 92 percent (OECD, 2024).

comprising one hundred languages. For enabling binary (i.e., dichotomous) and multiclass (i.e., polytomous) scoring, respectively, a classification head was appended to the pretrained model.

With the objective to only have the model assign fairly probable scores, labels’ output probabilities were stored for each instance. Using Receiver Operating Characteristic (ROC) analysis, an optimal threshold of label output probability o_j , specific to subsample j , was determined to minimize misclassifications. This threshold was determined by maximizing Youden’s index (Youden, 1950), which quantifies the trade-off between sensitivity and specificity. Specifically, we computed

$$o_j = \arg \max_{x \in [0,1]} \left(\frac{TP_x}{TP_x + FN_x} - \frac{FP_x}{FP_x + TN_x} \right),$$

where TP, FP, TN, and FN denote subsample j ’s number of true positives, false negatives, and so on, based on a vector of classification correctness at threshold x .

This threshold identification differs from conventional ROC analyses, which typically rely on the actual binary labels rather than their correctness. With tailored confidence thresholds, the XLM-R classifiers ensure reliable predictions while deferring uncertain cases to downstream components. Moreover, only classifiers with sufficient training performance were employed at all.

In addition to fine-tuned XLM-R classifiers, support vector machine classifiers were trained using XLM-R embeddings as the input features. With a small number of entirely underfitting XLM-R models, the SVM classifiers were designed as fallback classifiers before ultimately deferring to human scoring. While linguistic representation remained consistent with XLM-R classifiers, SVMs’ distinct classification provided—despite somewhat poorer accuracy—more robustness in scenarios where datasets may be small, noisy, or skewed in their class distribution.

As the threshold for inference certainty, SVM classifiers used the arithmetic mean probability instead of the ROC-based approach employed for fine-tuned XLM-R models. This simpler thresholding mechanism was chosen because SVMs were applied only to responses that had already been deemed uncertain by upstream models.

4 Empirical Evaluation

Omethi implemented as described above was evaluated by simulating its flow on a real-world dataset.

4.1 Dataset and Instrument

In PISA (OECD, 2023), 15-year-old students take tests in a total of three domains to assess their scientific, mathematics, and reading literacy. For the present study, we had available text responses for all construct-response items from all Field Trials and Main Studies for PISA 2018 and 2022. With the complete data being too large for one evaluation and its reporting, we sampled 11 datasets with diverse languages for the present paper: Arabic (Jordan), Traditional Chinese (Chinese Taipei), Finnish (Finland), English (U.S.), German (Germany), Hebrew (Israel), Indonesian (Indonesia), Kazakh (Cyrillic script; Kazakhstan), Korean (South Korea), Portuguese (Brazil), and Spanish (Spain). Corresponding to $n = 270,445$ students, this resulted in a total of $n = 3,773,728$ responses that had already been assigned human scores in PISA with its high quality standards (OECD, 2025).

The dataset comprised 160 items (89 reading, 39 math, and 32 science items), 121 with two and 39 of them with three score levels. Not all items had been administered in all selected languages, resulting in a total of 1,676 datasets (i.e., classifiers to be trained). Sample items with corresponding coding guides can be found on the OECD’s website (OECD, 2025). Coding guides for some items are simple, such as “*Full credit is given when the student states that the weight or size [...] was not provided ...*” (CR548Q09), others are more complex, such as “*Selects one of the names and gives an appropriate explanation as described below.*” (with 19 explanations specified and mapped to one of three different names; CR557Q14).

Table 1 shows exemplary responses for each domain. They are selected from coding guides released by the OECD and not from the evaluation data set, because items in PISA are confidential due to the assessment’s high stakes at the national level, constraining the selection options. Note that constructed-response items in math regularly involve mathematical reasoning (sometimes, naming a number), but rarely involve stating formulas.

4.2 Implementation

We used Python 3.11.5 and R 4.4.3 (R Core Team, 2025). For XLM-R, the base model² with 279 million parameters was used. Due to the large number of required classifiers, hyperparameters and

²<https://huggingface.co/FacebookAI/xlm-roberta-base> [2025-04-01]

Domain	Item ID	Item Stem	Sample Response	Context
Math	CMA159Q01	Peter thinks there is a greater probability of the arrow stopping on blue in Spinner A than there is in Spinner B. Is Peter correct?	Because $\frac{1}{2} = \frac{2}{4}$. He is not correct because the probability is the same for each spinner.	Details (OECD, 2025)
Reading	CR548Q09	With whom do you agree?	Sam. These are only two texts and more research is needed before a conclusion can be made.	Details (OECD, 2025)
Science	CS623Q03	What is the biological reason for this effect?	Increasing sweat levels in high temperatures keeps the body from getting too hot.	Details (OECD, 2024)

Table 1: Sample PISA items and responses from released coding guides

settings were not tuned classifier-wise for computational constraints. Instead, a fixed batch size of $B = 32$, learning rate of $\eta = 5e^{-5}$, the AdamW optimizer (Loshchilov and Hutter, 2017), and a cosine learning rate scheduler with warm-up were used. Training was capped at ten epochs, with early stopping after stagnating performance in three consecutive epochs. Cross-entropy loss was used for optimization. SVMs employed a Radial Basis Function (RBF) kernel.

Classifiers were deployed only if they met minimum training performance thresholds ($\kappa \geq .300$ for XLM-R and $\kappa \geq .900$ for SVM) as measured by QWK (Quadratic Weighted Kappa; Cohen, 1960). F_1 -score is reported as F_1 micro. Importantly, in the reported evaluation, the final manual scoring component was assumed to yield perfect accuracy, despite normal inconsistencies in human scoring. That is, this component takes the ground truth label, as provided by PISA’s human raters, as its output. This assumption was made for two reasons: (i) consistent estimates of inter-rater agreements were not available for all subsamples, and (ii) the substantial reduction in manual effort could alter relevant rater cognition (Bejar, 2017) and reliability (Padó and Padó, 2022).

Finally, due to computational constraints stemming from the fine-tuning of many XLM-R models, an 80/20 training-test-split was used for evaluation.

4.3 Results

All reported average values constitute means weighted by sample size across all classifiers.³

³For the sake of readability, only a selected set of standard errors is reported (in brackets) where comparisons may be relevant. All result data is available upon request.

	Acc (%)	κ	F_1	ER (%)
Math	98.8	.972	.988	73.0
Reading	97.7	.954	.977	71.0
Science	97.7	.951	.977	67.3

Table 2: Omethi’s performance by domain

4.3.1 Performance by Domain

Omethi achieved very high agreement with human scores, with an average $\kappa = .957$, $TPR = .968$, and $FPR = .977$, alongside substantial manual effort savings (on average, $ER = 70.5\%$). Notably, these results include a share of responses scored manually, as reflected in the effort reduction metric, and assume perfect agreement for this subset. Nonetheless, the reported figures represent the expected scoring quality if Omethi were deployed in an operational setting.

Table 2 details the agreement and effort reduction across all domains. Scores showed the highest agreement for math items with an average accuracy of 98.8 percent [$\pm 0.1\%$], $\kappa = .972$ [± 0.003], and an effort reduction of 73.0 percent [$\pm 1.2\%$], meaning 27.0 percent of responses have been deferred to human scoring. For the other two assessment domains, Omethi scores showed marginally lower but still very high agreement values, with accuracy at 97.7 percent [$\pm 0.1\%$] for reading and identical 97.7 percent [$\pm 0.3\%$] for science ($\kappa = .954$ [± 0.002] and $\kappa = .951$ [± 0.005]), and an effort reduction of 71.0 percent [$\pm 0.8\%$] and, for science somewhat lower, 67.3 percent [$\pm 1.1\%$], respectively.

The overall high agreement for the majority of classifiers across domains and languages with only rare exceptions is visualized in Figure 2.

Distinguishing performance of individual automatic components, Table 3 reports component-

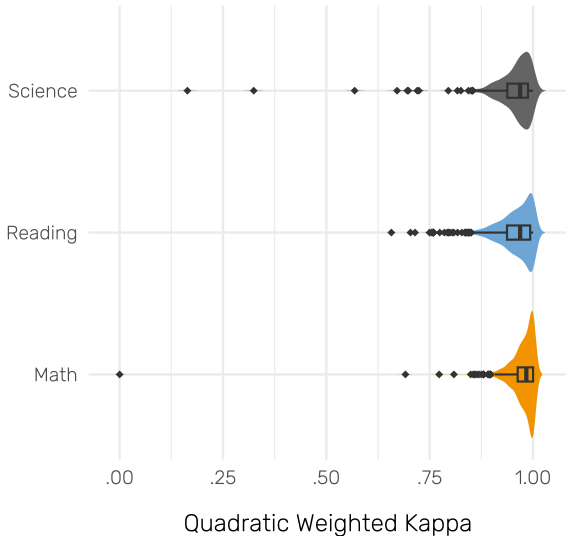


Figure 2: Omethi’s distribution of performance at the item- and language-level across domains

wise agreement values, excluding manual scoring during inference. The first three columns show the agreement of the components’ automatic score with human scores for the subset of responses that were scored by the respective component for Omethi. For FLM, accuracy was at $Acc_{FLM} = 98.6$ percent on average, XLM-R’s performance was $\kappa_{XLM} = .775 [\pm .011]$, and for SVM, $\kappa_{SVM} = .554$. The fourth column (κ_{auto}) shows agreement for all automatic components combined, again only for the subset of responses scored automatically; $\kappa_{auto} = .804 [\pm .009]$. The last column ($\kappa_{XLM_{all}}$) reports agreement as would have been the case if all responses were scored by XLM-R classifiers alone, showcasing the substantial added value of the combination of automatic scoring approaches beyond transformer finetuning as displayed in the adjacent column on the left ($\Delta\kappa_{auto, XLM_{all}} = .093$). The Appendix visualizes this gain of the Omethi pipeline over mere XLM-R fine-tuning (see Figure 4). Similarly, Figure 5 (see Appendix again) shows that XLM-R outperforms SVM for the majority of item- and language-specific classifiers, which is in line with the rationale underlying the component hierarchy, while also showcasing the number of random-level XLM-R classifiers, for which SVM was added as a potential fallback.

4.3.2 Robustness across Languages

Table 4 displays Omethi’s agreement with human scores and effort reduction across subsamples,

	Acc_{FLM}	κ_{XLM}	κ_{SVM}	κ_{auto}	$\kappa_{XLM_{all}}$
<i>Math</i>	99.2	.715	.414	.765	.683
<i>Reading</i>	98.2	.792	.584	.845	.739
<i>Science</i>	98.8	.784	.600	.756	.677

Table 3: Performance for automatic components and their combination (*auto*); responses deferred to manual scoring excluded, except for XLM_{all}

which includes, among others, different test languages. Overall agreement was homogeneously high, with accuracy ranging mainly from 97.8 (Spain) to 99.0 percent (Jordan) and the exception of Indonesia with 96.9 percent. In contrast, effort reduction varies largely from 60.3 (Indonesia) to 76.1 percent (Chinese Taipei), showing how the implemented scoring conditions effectively identified instances that required human scoring. Similarly, the shares of responses scored by different components varied heterogeneously across subsamples (see Appendix A, Table 6). For example, FLM scored 29.5 percent of responses in the subsample from Chinese Taipei (Traditional Chinese), which stands in stark contrast to the one from Israel (Hebrew) with only 19.7 percent. For Spain (Spanish), SVMs only scored 1.2 percent of the responses, compared to Jordan (Arabic) with 9.8 percent combined with an outlier of only 28.1 percent of sufficiently confident scoring by XLM-R, whereas the XLM-R classifiers for the U.S. (English) scored even 52.2 percent of the responses.

	Acc (%)	κ	F_1	ER (%)
<i>ara-jor</i>	99.0	.965	.990	66.7
<i>deu-deu</i>	98.1	.961	.981	72.2
<i>eng-usa</i>	98.0	.959	.980	75.5
<i>esp-esp</i>	97.8	.957	.978	70.4
<i>fin-fin</i>	98.5	.969	.985	75.2
<i>heb-isr</i>	98.1	.961	.981	70.8
<i>ind-idn</i>	96.9	.928	.969	60.3
<i>kaz-kaz</i>	98.3	.962	.983	67.9
<i>kor-kor</i>	98.3	.965	.983	75.1
<i>por-bra</i>	98.4	.965	.984	72.0
<i>zho-tap</i>	98.1	.963	.981	76.1

Table 4: Performance and effort reduction by language, incl. manual scoring

4.3.3 Component Shares

Table 5 shows the percentage of responses scored by the respective component due to meeting the eligibility policy. FLM and XLM-R played the major role for automatic scoring. With its position at the end of the sequence of automatic components, SVMs only played a minor role quantitatively. Nevertheless, as shown in Table 6, there were settings, such as the subsample from Jordan (Arabic) in which the first automatic components do not perform well and SVM takes over some of the shares to retain the homogeneously high level of accuracy.

The prevalence of different flows responses take through the scoring components is displayed in Figure 3. Said cases where XLM-R and FLM do not manage to score responses for which SVM takes over are visible in the figure as the orange ribbon. Moreover the figure disentangles the specific conditions for why responses are not scored by specific components.

5 Discussion

The results demonstrate Omethi’s effectiveness in orchestrating multiple methods in an explicitly designed, adaptive scheme for automatic scoring across domains and languages while maintaining uniformly high accuracy. With an average agreement of $\kappa = .957$ compared to complete human scoring and manual effort reductions of 70.5 percent across domains, Omethi proves its feasibility in and operational usefulness. Thus far, for PISA data, effort reduction gains have been reported to be smaller with other methods and data sets comprising fewer test languages and assessment domains (Andersen et al., 2023; Yamamoto et al., 2018). Critically, the system’s hierarchical composition and scoring conditions ensured that accuracy was prioritized, resulting in varying effort reduction across settings. It is important to note that *manual effort reduction* here does not refer to the entirety of human involvement in operational assessment procedures but only the share of automatically scored responses during inference.

	FLM	XLM-R	SVM	Manual
<i>Math</i>	35.0	34.2	3.8	27.0
<i>Reading</i>	24.1	44.5	2.5	29.0
<i>Science</i>	18.4	46.0	2.8	32.7

Table 5: Proportions (%) of component usage

The importance of combining different methodologies was evidenced by the homogeneous accuracy levels despite heterogeneous shares of responses being scored by different components across domains and languages. Each component in Omethi played a distinct role, contributing to the system’s overall robustness. While FLM and XLM-R dominated the scoring, partly due to their position in the sequence, SVMs served as a crucial fallback mechanism, stepping in when upstream components failed to score confidently. Although SVMs scored only a minor share of responses quantitatively, their role turned out as indispensable in maintaining accuracy for certain subsamples. This underscores the importance of the adaptive workflow, where eligibility policies diagnosed the risk of misclassifications and led to passing on responses.

For identifying a confidence threshold as components’ eligibility policy, the proposed maximizing of the ROC-based Youden’s Index on misclassifications worked excellently for XLM-R classifiers. Less so for SVM classifiers that were faced with only the more challenging responses not scorable by upstream components. Hence, this measure may be added to the repertoire of threshold identification methods, complementing fixed constants (e.g., Li et al., 2025) or the definition of minimum F_1 scores as proxies (e.g., Bexte et al., 2024), but its suitability needs to be verified.

Omethi’s strength in adaptivity may also introduce challenges in ensuring equivalence and fairness across test languages and subpopulations, necessitating careful validation and bias checks, as bias is known to be potentially masked at the aggregate level (Andersen et al., 2023). From an operational standpoint, implementing Omethi in international large-scale assessments would require a rigorous quality monitoring.

For human raters, the implementation of such a framework would result in multiple changes that may affect rater cognition in positive or negative ways, or both. First, the number of responses decreases, potentially leading to less fatigue, monotonous work, and slippage. Second, raters’ oversight of frequent responses would diminish and would thus change so-called contrast or context effects by preceding responses, an effect repeatedly found even in highly standardized settings with well-trained raters (Attali, 2011; Meadows and Billington, 2005). Third, both for automatic systems as well as humans (Padó and Padó, 2022), incorrect responses are more challenging to score.

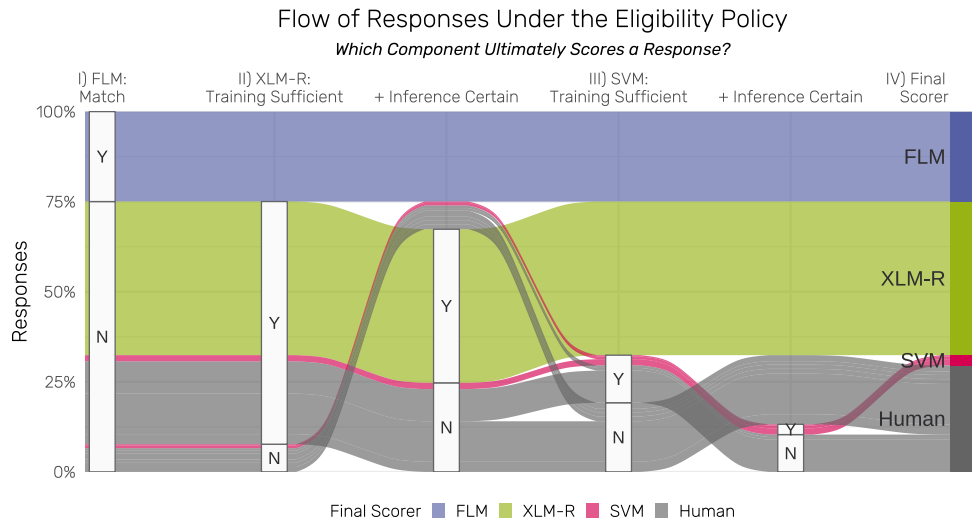


Figure 3: Share of response flows through Omethi’s components. Y/N (yes/no) = eligibility policy satisfied?

Accordingly, if classifiers with poor recall leave human raters with a higher frequency of incorrect responses, this also may show effects.

In conclusion, Omethi’s response-level adaptivity, combined with its capability to maintain high accuracy across diverse contexts, positions it as a powerful tool for operational employment in assessments. While challenges may remain in, among others, ensuring fairness, the system’s effectiveness and flexibility pave the way for its operational use and future enhancements. In follow-up studies, the results presented here may be used to sample specific datasets informative to diverse facets to carry out an evaluation in order to systematize performance differences and identify optimal hyperparameters, respectively.

Ethical Considerations

The implementation of a framework such as Omethi in large-scale assessments necessitates careful attention to ethical principles, which is not always at the forefront of attention (Holmes et al., 2022). Fairness and the mitigation of bias are paramount, as variability in component usage across languages and cultures could lead to disproportionate disadvantages for certain groups. Rigorous validation and bias investigations are essential to ensure equitable performance across diverse populations. Transparency in the scoring process is critical to fostering trust among stakeholders, including organizations such as the OECD, policymakers, and test takers. Clear documentation of the system’s decision-making mechanisms and limitations must be provided to ensure interpretability and account-

ability. Additionally, equity in resource allocation must be discussed, as disparities in system performance between high- and low-resource languages could exacerbate existing inequalities. Finally, the increasing automation of standardized assessments raises broader questions about their role in education. While automation enhances efficiency and scalability, it also risks amplifying uniformity, potentially overlooking diversity facets.

Limitations

The study faces several limitations, primarily due to computational constraints. With many test languages, items, and domains, a large number of item- and language-specific classifiers were fine-tuned using the XLM-RoBERTa base model. This scale rendered classifier-specific hyperparameter tuning via grid search computationally infeasible, necessitating the use of fixed hyperparameters. Similarly, k -fold cross-validation was not conducted due to resource limitations, restricting the evaluation to a single 80/20 train-test split.

The system’s runtime scales with the number of components, complicating potential real-time deployment in certain settings. Additionally, the evaluation focused exclusively on operational data, lacking comparison with public benchmarks or standardized datasets. The use of human scoring as the gold standard, particularly for responses deferred to manual scoring, assumes perfect inter-rater reliability, which may overestimate accuracy in production.

References

- Nico Andersen, Fabian Zehner, and Frank Goldhammer. 2023. [Semi-automatic coding of open-ended text responses in large-scale assessments](#). *Journal of Computer Assisted Learning*, 39(3):841–854.
- Yigal Attali. 2011. [Sequential effects in essay ratings](#). *Educational and Psychological Measurement*, 71(1):68–79.
- Isaac I. Bejar. 2017. A historical survey of research regarding constructed-response formats. In Randy Elliot Bennett and Matthias von Davier, editors, *Advancing Human Assessment*, pages 565–633. Springer, Cham.
- 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.
- W. L. Chafe. 1982. Integration and involvement in speaking, writing, and oral literature. *Spoken and Written Language: Exploring Orality and Literacy*, pages 35–54.
- Jacob Cohen. 1960. [A coefficient of agreement for nominal scales](#). *Educational and Psychological Measurement*, 20(1):37–46.
- 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 Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics, Stroudsburg, PA, USA.
- Corinna Cortes and Vladimir Vapnik. 1995. [Support-vector networks](#). *Machine Learning*, 20(3):273–297.
- Hiroaki Funayama, Tasuku Sato, Yuichiroh Matsubayashi, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2022. [Balancing cost and quality: An exploration of human-in-the-loop frameworks for automated short answer scoring](#). In Maria Mercedes Rodrigo, Noburu Matsuda, Alexandra I. Cristea, and Vania Dimitrova, editors, *Artificial Intelligence in Education*, volume 13355, pages 465–476. Springer International Publishing, Cham.
- Lucas Busatta Galhardi and Jacques Duílio Brancher. 2018. [Machine learning approach for automatic short answer grading: A systematic review](#). In Guillermo R. Simari, Eduardo Fermé, Flabio Gutiérrez Segura, and José Antonio Rodríguez Melquiades, editors, *Advances in Artificial Intelligence – IBERAMIA 2018*, volume 11238, pages 380–391. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.
- Rujun Gao, Hillary E. Merzdorf, Saira Anwar, M. Cynthia Hipwell, and Arun R. Srinivasa. 2024. [Automatic assessment of text-based responses in post-secondary education: A systematic review](#). *Computers and Education: Artificial Intelligence*, 6(100206).
- Palak Goenka, Mehak Piplani, Ramit Sawhney, Puneet Mathur, and Rajiv Ratn Shah. 2020. [ESAS: Towards practical and explainable short answer scoring \(student abstract\)](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(10):13797–13798.
- Michael Heilman and Nitin Madnani. 2013. ETS: Domain adaptation and stacking for short answer scoring. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 275–279.
- Wayne Holmes, Kaska Porayska-Pomsta, Ken Holstein, Emma Sutherland, Toby Baker, Simon Buckingham Shum, Olga C. Santos, Mercedes T. Rodrigo, Mutlu Cukurova, Ig Ibert Bittencourt, and Kenneth R. Koedinger. 2022. [Ethics of AI in education: towards a community-wide framework](#). *International Journal of Artificial Intelligence in Education*, 32(3):504–526.
- Andrea Horbach, Alexis Palmer, and Magdalena Wolska. 2014. [Finding a tradeoff between accuracy and rater’s workload in grading clustered short answers](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 588–595, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Andrea Horbach and Manfred Pinkal. 2018. [Semi-supervised clustering for short answer scoring](#). In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Andrea Horbach and Torsten Zesch. 2019. [The Influence of Variance in Learner Answers on Automatic Content Scoring](#). *Frontiers in Education*, 4:4.
- Yuheng Li, Mladen Raković, Namrata Srivastava, Xinyu Li, Quanlong Guan, Dragan Gašević, and Guanliang Chen. 2025. [Can AI support human grading? Examining machine attention and confidence in short answer scoring](#). *Computers & Education*, 228:105244.
- Zhaohui Li, Chengning Zhang, Yumi Jin, Xuesong Cang, Sadhana Puntambekar, and Rebecca J. Passonneau. 2023. [Learning when to defer to humans for short answer grading](#). In *Artificial Intelligence in Education*, pages 414–425, Cham. Springer Nature Switzerland.
- Ilya Loshchilov and Frank Hutter. 2017. [Decoupled weight decay regularization](#). *arXiv preprint arXiv:1711.05101*.

- Michelle Meadows and Lucy Billington. 2005. A review of the literature on marking reliability.
- Michael Mohler, Razvan C. Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Annual meeting of the association for computational linguistics*.
- OECD. 2023. *PISA 2022 results (volume I): The state of learning and equity in education*. PISA. OECD.
- OECD. 2024. *PISA 2015 Released Field Trial Cognitive Items*.
- OECD. 2025. *PISA test*. Accessed: 2025-04-15.
- Christopher M. Ormerod. 2022. Short-answer scoring with ensembles of pretrained language models. *ArXiv*, abs/2202.11558.
- Ulrike Padó and Sebastian Padó. 2022. Determinants of grader agreement: an analysis of multiple short answer corpora. *Language Resources and Evaluation*, 56(2):387–416.
- R Core Team. 2025. *R: A language and environment for statistical computing*. Manual, R Foundation for Statistical Computing, Vienna, Austria.
- Shourya Roy, Himanshu S. Bhatt, and Y. Narahari. 2016. An iterative transfer learning based ensemble technique for automatic short answer grading. *ArXiv*, abs/1609.04909.
- Archana Sahu and Plaban Kumar Bhowmick. 2020. Feature engineering and ensemble-based approach for improving automatic short-answer grading performance. *IEEE Transactions on Learning Technologies*, 13(1):77–90.
- Keisuke Sakaguchi, Michael Heilman, and Nitin Madnani. 2015. Effective feature integration for automated short answer scoring. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1049–1054, Denver, Colorado. Association for Computational Linguistics.
- Analia Schlosser, Zvika Neeman, and Yigal Attali. 2019. Differential performance in high versus low stakes tests: Evidence from the GRE test. *The Economic Journal*, 129(623):2916–2948.
- Hyo Jeong Shin, Nico Andersen, Andrea Horbach, Euiyum Kim, Jisoo Baik, and Fabian Zehner. 2024. Operational automatic scoring of text responses in 2016 ePIRLS: Performance and linguistic variance.
- Hyo Jeong Shin, Edward Wolfe, and Mark Wilson. 2019. Human rater monitoring with automated scoring engines. *Psychological Test and Assessment Modeling*, 61(2):127–148.
- King Yiu Suen, Victoria Yaneva, Le An Ha, Janet Mee, Yiyun Zhou, and Polina Harik. 2023. ACTA: short-answer grading in high-stakes medical exams. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 443–447, Toronto, Canada. Association for Computational Linguistics.
- John Whitmer, Evelyn Deng, Charles Blankenship, Magdalen Beiting-Parrish, Ting Zhang, and Paul Bailey. 2023. Results of NAEP Reading Item Automated Scoring Data Challenge (Fall 2021). Publisher: EdArXiv.
- Kentaro Yamamoto, Qiwei He, Hyo Jeong Shin, and Matthias von Davier. 2018. Development and implementation of a machine-supported coding system for constructed-response items in PISA. *Psychological Test and Assessment Modeling*, 60(2):145–164.
- Duanli Yan, André A. Rupp, and Peter W. Foltz, editors. 2020. *Handbook of automated scoring: Theory into practice*. Statistics in the social and behavioral sciences series. CRC Press, Taylor & Francis Group, Boca Raton, FL.
- W. J. Youden. 1950. Index for rating diagnostic tests. *Cancer*, 3(1):32–35.
- Fabian Zehner, Frank Goldhammer, Emily Lubaway, and Christine Sälzer. 2019. Unattended consequences: How text responses alter alongside PISA’s mode change from 2012 to 2015. *Education Inquiry*, 10(1):34–55.
- Fabian Zehner, Ulf Kroehne, Carolin Hahnel, and Frank Goldhammer. 2020. PISA reading: Mode effects unveiled in text responses. *Psychological Test and Assessment Modeling*, 62:55–75.
- Fabian Zehner, Christine Sälzer, and Frank Goldhammer. 2016. Automatic coding of short text responses via clustering in educational assessment. *Educational and Psychological Measurement*, 76(2):280–303.
- Torsten Zesch, Andrea Horbach, and Fabian Zehner. 2023. To Score or not to score: Factors influencing performance and feasibility of automatic content scoring of text responses. *Educational Measurement: Issues and Practice*, 42(1):44–58.
- Lishan Zhang, Yuwei Huang, Xi Yang, Shengquan Yu, and Fuzhen Zhuang. 2022. An automatic short-answer grading model for semi-open-ended questions. *Interactive Learning Environments*, 30(1):177–190.

A Appendix

A.1 Proportions of Component Usage by Subsample

	FLM	XLM-R	SVM	Manual
<i>ara-jor</i>	28.8	28.1	9.8	33.3
<i>deu-deu</i>	23.3	46.1	2.9	27.8
<i>eng-usa</i>	20.0	52.2	3.2	24.5
<i>esp-esp</i>	25.8	43.4	1.2	29.6
<i>fin-fin</i>	25.3	46.4	3.5	24.8
<i>heb-isr</i>	19.7	46.8	4.4	29.2
<i>ind-idn</i>	23.0	35.0	2.3	39.7
<i>kaz-kaz</i>	27.5	37.6	2.8	32.1
<i>kor-kor</i>	21.1	50.1	4.0	24.9
<i>por-bra</i>	28.8	39.8	3.4	28.0
<i>zho-tap</i>	29.5	44.1	2.5	23.9

Table 6: Proportions (%) of component usage by subsample

A.2 Gains Beyond Mere XLM-R Fine-Tuning

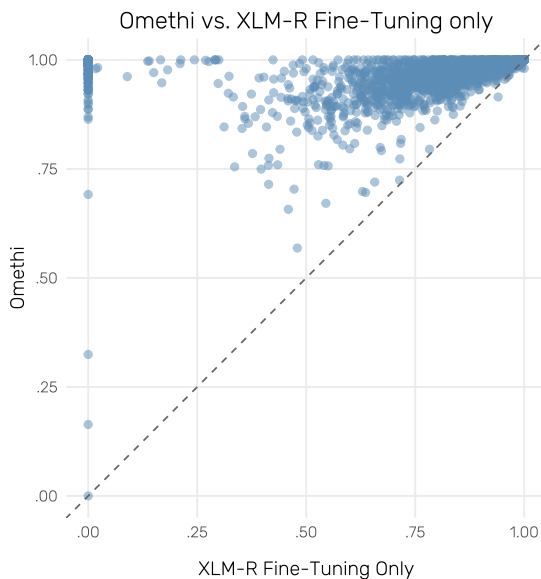


Figure 4: Quadratic Weighted Kappa of XLM-R fine-tuning applied to all responses and Omethi. The complete Omethi pipeline, which includes XLM-R itself and a share of human-scored responses, strongly outperforms XLM-R consistently (values above the diagonal).

A.3 XLM-R Fine-Tuning and SVM

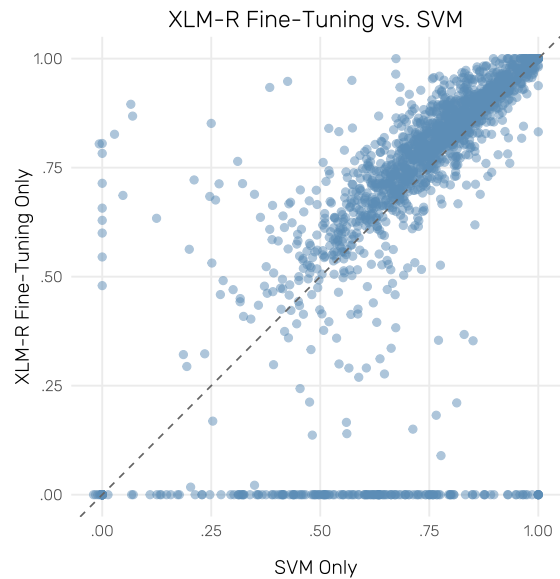


Figure 5: Quadratic Weighted Kappa of XLM-R fine-tuning applied to all responses and SVM. Generally, XLM-R outperforms SVM for the majority of classifiers (values above the diagonal), but SVM shows to be more robust with respect to a number of XLM-R classifiers only showing chance-level performance.

A.4 Acknowledgments

We gratefully acknowledge the OECD for granting access to PISA data, and we thank their PISA Research and Development for Innovation programme, whose funding of a prior project led to the development of the FLM method presented here.

Lessons Learned in Assessing Student Reflections with LLMs

Mohamed Elaraby, Diane Litman

University of Pittsburgh

Pittsburgh, PA, USA

{mse30,dlitman}@pitt.edu

Abstract

Advances in Large Language Models (LLMs) have sparked growing interest in their potential as explainable text evaluators. While LLMs have shown promise in assessing machine-generated texts in tasks such as summarization and machine translation, their effectiveness in evaluating human-written content—such as student writing in classroom settings—remains underexplored. In this paper, we investigate LLM-based specificity assessment of student reflections written in response to prompts, using three instruction-tuned models. Our findings indicate that although LLMs may underperform compared to simpler supervised baselines in terms of scoring accuracy, they offer a valuable interpretability advantage. Specifically, LLMs can generate explanations that are faithful, non-repetitive, and exhibit high fidelity with their input, suggesting potential for enhancing the transparency and usability of automated specificity scoring systems.

1 Introduction

Reflective writing is a fundamental skill that enhances learning by encouraging students to critically engage with course material and articulate their thoughts. This process benefits both students and instructors by fostering greater awareness and facilitating meaningful classroom interactions (Baird et al., 1991; Menekse, 2020). The quality of written reflections is often assessed based on their *specificity* (Menekse et al., 2011; Li et al., 2025), which measures the level of detail and depth in a given reflection. Table 1 shows student reflections written after a physics lecture, along with human-assessed specificity scores, both from the ReflectSumm corpus described in Section 3.

Automating specificity assessment is crucial for delivering interventions to help students improve the quality of their reflections (Knoth et al., 2020; Wilhelm, 2021), e.g., by providing scaffolding feed-

Prompt: Describe what you found most confusing in today’s class.

[Score 1] I thought that most of the topics explained were relatively simple or I had previously learned them. I felt confident in my understanding after the class session.

[Score 2] the class participation activity

[Score 3] Undirected vs directed was a bit confusing in terms of how to read the chart

[Score 4] Finding the right problem to address.

Prompt: Describe what you found most interesting in today’s class.

[Score 1] I found nothing interesting in class. Being Friday, I could barely pay attention.

[Score 2] the review session

[Score 3] The part about bias in data labeling was thought provoking

[Score 4] Writing the problem statement.

Table 1: Representative reflections for each specificity score (1–4) across two prompts. This illustrates one challenge of assessing specificity: long reflections may lack substance (Score 1), while short ones may convey detailed, content-specific insights (Score 4).

back which in turn can ultimately enhance learning outcomes (Menekse et al., 2025). More specific reflections can also provide instructors with more reliable insights into student understanding and needs (Menekse, 2020). Traditionally, specificity scoring has relied on supervised models (Magoooda et al., 2022; Carpenter et al., 2020; Li and Nenkova, 2015). However, collecting large annotated datasets in educational contexts is resource-intensive and not always feasible. Depending on model type, the reasoning behind the scoring might also not be explainable to students or instructors.

Advancements in Large Language Models (LLMs) for evaluating natural language, commonly referred to as the *LLM-as-a-judge* paradigm (Zheng et al., 2023), have introduced new possibilities for leveraging LLMs in educational applications. These models can generate human-like judgments

Research Question	Key Finding	Lessons
RQ1: LLM vs. Supervised Baselines	Retrieval-based few-shot improves scoring LLMs underperform supervised models Chain-of-Thought (CoT) explanations do not improve scoring	Selecting semantically similar in-context examples boosts accuracy over random or fixed examples. Distill-BERT outperforms all LLMs, suggesting a need for adaptation. Generated explanations fail to enhance LLM-based scoring.
RQ2: Self-Generated Explanations	Explanations are faithful to the input and explanation vocabulary do not fully overlap with the input reflection vocabulary High fidelity suggests explanations are highly influencing the predictions	Explanations do not contradict or repeat the input, suggesting potential for interpretable and supportive understanding of the scores. Misleading explanations can negatively affect the scoring.

Table 2: Key findings of RQ1 and RQ2 .

without task-specific training, making them attractive for low-resource tasks such as reflection specificity assessment. Their generative capabilities in addition suggest new possibilities for explainable methods. This paper investigates whether LLMs can serve as viable alternatives to traditional supervised models for assessing reflection specificity.

Reflective writing poses challenges compared to other educational scoring tasks. Unlike Automatic Essay Scoring (Foltz et al., 1999; Attali and Burstein, 2004; Shermis and Wilson, 2024), which typically assesses longer texts, reflective writing is often concise and highly variable in length, with reflections ranging from a single word to multiple phrases or complete sentences (Kember et al., 2008). This variability poses a unique challenge in distinguishing different levels of specificity: shorter reflections may lack sufficient context, while longer ones can introduce ambiguity in assessment. Table 1 presents two examples of reflections that contain multiple sentences yet receive the lowest specificity score (1) due to vague or off-topic content, as well as two shorter reflections that achieve the highest specificity score (4) by providing concise, content-rich responses relevant to the prompt. Also, while tasks such as Short Answer Grading (Burrows et al., 2015), which is closer in length and variability to reflections, primarily involve assessing objective responses within a given question context with reference answers, reflective writing is inherently subjective as it conveys personal experiences and insights, further complicating standardized assessment.

In this work, we extend prior research on leverag-

ing LLMs as judges for educational text evaluation by focusing on reflection specificity. We investigate this through two research questions: **RQ1: Can LLM-based specificity assessment improve scoring reliability compared to supervised baselines?** We explore two approaches to LLM-based specificity assessment: (1) *Standard Prompting*: LLMs are instructed to predict specificity scores based on the input reflection. (2) *Chain-of-Thought (CoT) Prompting*: LLMs are prompted to generate a rationale before making a specificity judgment. This technique, widely used in complex NLP tasks, encourages models to engage in step-by-step reasoning, potentially leading to more consistent and interpretable assessments. We investigate these settings under both *zero-shot* and *few-shot* conditions to assess their impact on model performance. **RQ2: Do self-generated explanations enhance interpretability?** We investigate whether generated explanations contribute to the transparency of LLM-based specificity scoring, potentially making the evaluation process more interpretable and informative for students and educators. *Our key findings are summarized in Table 2.*¹

Our contributions are twofold:

1. We evaluate the effectiveness of three open-weight LLMs in scoring student specificity under various zero-shot and few-shot settings.
2. We analyze the linguistic properties of the generated explanations and their role in interpreting the output, aiming to assess whether these

¹<https://github.com/EngSalem/Explainable-Reflection-Quality>

Score	Specificity Meaning	Definition
1	Vague	Reflection implies "no confusing issue," e.g., "nothing" or "none for this class."
2	Non-specific	Reflection does not include any statement(s) about course content but refers to organizational aspects (e.g., homework, exams).
3	General	Reflection includes statement(s) about course content but lacks specific details.
4	Specific	Reflection includes specific and detailed statement(s) about course content.

Table 3: Rubric for evaluating reflection specificity based on decision tree from Luo and Litman (2016).

explanations are meaningful and potentially useful for providing students with feedback on their reflective writing.

2 Related Work

LLM-as-Judge LLMs have demonstrated correlation with human evaluation of machine-generated texts in tasks such as counter-narrative generation (Zubiaga et al., 2024), text summarization (Fu et al., 2024; Liu et al., 2023), multi-turn question answering (Zheng et al., 2023), and automatic persuasion ranking (Elaraby et al., 2024). However, for more nuanced human-written content, such as academic reviews (Zhou et al., 2024) and essay scoring (Mansour et al., 2024; Stahl et al., 2024), LLMs (particularly without fine-tuning or alignment) still fall short compared to human evaluators and domain-specific supervised models trained on high-quality annotated data. *In this work, we investigate LLMs as specificity evaluators for student reflections, a distinct category of human-written text.*

LLM-as-Judge in Educational Text Traditional approaches to assessing student writing often rely on surface linguistic features to enhance automatic scoring models ranging from feature-based to hybrids with deep learning, including list ranking (Uto et al., 2020) and neural-based methods (Jin et al., 2018; Uto et al., 2020). Recent work has explored leveraging LLMs as evaluators for educational text. Stahl et al. (2024) employed persona-based zero-shot prompting for essay scoring, and Hou et al. (2025) integrated linguistic features into zero-shot evaluations; however, both studies found limited improvements over traditional supervised baselines. In contrast, Baral et al. (2024) showed that a fine-tuned Mistral-7B model outperformed other supervised models in math essay scoring. Closely related to our work, Li et al. (2025) investigated reflection specificity assessment, demonstrating that multi-LLM voting strategies outperform single LLM scoring approaches. *Building on these developments, our work examines LLMs’ capabilities*

in assessing student reflections, focusing on how in-context examples influence predictions. Additionally, we analyze the interpretability of LLM-generated explanations, offering a novel perspective particularly valuable for building downstream applications in high-stakes domains like education.

Evaluating Self-Generated Explanations Assessing self-generated explanations has largely centered on their impact on model performance. Existing metrics, such as accuracy differences with and without explanations (Hase et al., 2020a; Wiegrefe et al., 2021a) and information-theoretic measures (Chen et al., 2023), quantify how explanation content influences predictions. Wiegrefe and Marasovic (2021) proposed evaluation criteria based on surface validity, grammatical correctness, and alignment with the target label, including *contrast* with alternative labels. Expanding on this, Joshi et al. (2023) introduced *novelty*, capturing the introduction of new information, which proved useful in human-AI collaboration tasks. These measures have since been extended to domains like persuasiveness evaluation (Elaraby et al., 2024). Despite these advancements, self-generated explanations remain largely unexplored in educational contexts beyond their role in enhancing automatic scoring (Stahl et al., 2024). *In this work, we examine their effectiveness not only in improving specificity scoring, but also for their potential to generate explanations that are faithful and non-repetitive with input, and exhibit fidelity with scoring.*

3 Datasets

For LLM evaluation, we utilize ReflectSumm² (Zhong et al., 2024), a corpus of 17,509 reflections aggregated by unique reflection per lecture from 24 STEM courses across 2 universities, written in response to the prompts in Table 1. This dataset was selected for its inclusion of high-quality annotations of individual reflection specificity scores,

²<https://huggingface.co/datasets/mse30/ReflectSumm>

Score	Count	Ref. Length (Min / Mean / Max)
1	1,841	1 / 11.19 / 135
2	2,488	1 / 6.17 / 62
3	9,231	2 / 15.26 / 87
4	3,949	4 / 30.19 / 194

Table 4: Distribution of reflection specificity scores in the ReflectSumm dataset, with reflection (ref.) length in number of words statistics.

rated on a scale from 1 (vague) to 4 (specific) using the rubric in Table 3. The annotations exhibit substantial inter-annotator agreement, with a reported pairwise Quadratic Weighted Kappa of 0.668 across 4 distinct annotators (trained college students with backgrounds in the appropriate subject domains) (Zhong et al., 2024). Table 4 summarizes the score distribution. The table also emphasizes the variability in reflection lengths across all scores.

For both training supervised pre-LLM baselines and as a reflection bank for LLM in-context prompting, we use the publicly available annotated reflections from the CourseMIRROR dataset³ which is composed of 6680 reflections distributed as 1210, 2035, 2377, 1058 for scores 1 – 4, respectively. Note that although annotated using the same specificity rubric, the CourseMIRROR reflections are from STEM course offerings that are disjoint from those in ReflectSumm.

4 Experimental Settings

4.1 Included LLMs

We included 3 whitebox models which demonstrated strong performance across NLP tasks, as evaluated in the Chatbot Arena leaderboard (Zheng et al., 2023)⁴: Llama3.1-8B-instruct (Grattafiori et al., 2024), Mistral-8B-instruct (Jiang et al., 2024), and Qwen-7B (Yang et al., 2024). For efficient inference, we employed VLLMs (Kwon et al., 2023). All experiments were conducted with a decoding temperature set to 0, enabling greedy decoding to mitigate variability that might stem from temperature sampling.

4.2 Prompting the LLMs (Zero-Shot)

Building on the reflection quality assessment of Luo and Litman (2016), we prompt LLMs to assign specificity scores on a scale from 1 to 4, consistent with the guidelines provided to human annotators.

³<https://engineering.purdue.edu/coursemirror/>

⁴<https://lmarena.ai/>

The scoring rubric is adapted from the decision-tree criteria described in Luo and Litman (2016), and its definitions are presented in Table 3. This alignment enables a direct comparison between model predictions and dataset reference annotations. Appendix A provides the prompts used for scoring.

4.3 Scoring Evaluation Metric

Given that our prediction is based on point-wise scoring, we rely on **Quadratic Weighted Kappa (QWK)** to report the model prediction agreement with ground truth human annotations.

5 RQ1: LLM-based Assessment vs. Supervised Baselines

These experiments evaluate the effectiveness of LLM-based specificity assessment across a range of settings to enable a comprehensive comparison.

5.1 Supervised Baselines

We include two supervised baselines. The first is **Finetuned-DistilBERT**, where we fine-tune DistilBERT (Sanh, 2019) for specificity assessment following Magooda (2022). The model is initialized from the Hugging Face checkpoint⁵ and trained on the annotated reflections from the CourseMIRROR dataset. We fine-tuned the model for 20 epochs using 5-fold cross-validation, optimizing hyperparameters such as the learning rate and number of training epochs. The checkpoint with the highest overall QWK score was selected for evaluating specificity across the full ReflectSumm corpus.

The second baseline is **Nearest Neighbors (NN) Retrieval**. Given a target reflection, we retrieve semantically similar reflections from an annotated reflection bank R_{bank} and estimate its specificity score based on the most frequently occurring specificity label among its nearest neighbors. We use the CourseMIRROR dataset as the reflection bank. This method is used as a comparable baseline to LLMs with nearest neighbor in-context examples (Section 5.2). For each reflection in the ReflectSumm evaluation set, we generate a dense embedding using the all-MiniLM-L6-v2 model from the sentence-transformers library (Reimers and Gurevych, 2019). This maps reflections into a shared vector space, enabling semantic similarity comparisons. We compute the cosine similarity between each reflection in ReflectSumm and the annotated reflections in CourseMIRROR

⁵[distilbert/distilbert-base-uncased](https://huggingface.co/distilbert/distilbert-base-uncased)

(R_{bank}), and retrieve the top- n most similar reflections. For efficiency, we use the Faiss library (Johnson et al., 2019) for fast approximate nearest-neighbor search. The specificity score of a reflection is determined using a mode-based voting mechanism from its nearest neighbors’ specificity labels.

5.2 Prompting with In-Context Examples

We explore three in-context learning strategies, ranging from fixed demonstrations (Brown et al., 2020) to selection-based strategies that draw from pre-existing demonstrations (Min et al., 2022).

(1) **Fixed In-Context Examples:** A fixed set of manually curated examples is used as in-context demonstrations across all runs. These examples are drawn from annotated student reflections in Luo and Litman (2016) and remain unchanged during prompting. Since the examples provided in the original paper focused primarily on *confusing* prompts, we supplemented them with additional reflections written in response to *interesting* prompts from the R_{bank} set. Each specificity score is represented by an equal number of examples to ensure balanced coverage.⁶ This prompting method serves as a baseline for few-shot in-context learning.

(2) **Random In-Context Examples:** For each instance, n examples are randomly sampled from the annotated reflection bank R_{bank} . This approach assesses the variability in model performance based on arbitrary example selection.

(3) **Nearest-Neighbor In-Context Examples:** Similar to the nearest-neighbor retrieval baseline, the top- n semantically similar reflections from R_{bank} are retrieved for each input reflection. These nearest neighbors serve as in-context demonstrations.

Table 5 shows that none of the included LLMs were able to match the performance of the DistillBERT baseline (0.658 QWK) in either zero-shot or any of the few-shot settings. This highlights the limitations of LLMs in specificity assessment when compared to dedicated supervised models. Among the LLMs, Mistral-8B-instruct consistently achieved the highest QWK agreement across both zero-shot and few-shot settings. The best performance (0.624 QWK) was obtained when paired with nearest-neighbor retrieval, indicating that retrieving semantically similar reflections enhances the model’s ability to assess specificity by providing more contextually relevant examples. However,

⁶Fixed in-context examples are provided in Appendix B.

increasing the number of in-context examples negatively impacted performance across all models and few-shot settings. This suggests that excessive context may introduce conflicting information or divert the model’s attention away from the specificity criteria. Also, both fixed and randomly sampled in-context examples performed worse than zero-shot prompting, implying that arbitrarily chosen examples introduce noise rather than meaningful guidance. These findings underscore the importance of carefully curating in-context examples when leveraging LLMs for specificity scoring. *This limitation further reinforces the challenge of deploying LLMs for automated assessment in educational settings without access to high-quality annotated datasets.*

5.3 Chain-of-Thought (CoT) Prompting

Instead of directly instructing the model to assign a specificity score to a given reflection, we employ *Chain-of-Thought (CoT) prompting* (Wei et al., 2022) to encourage the model to generate a rationale before providing its final assessment. This approach aims to enhance the reliability and interpretability of the model’s scoring process by explicitly incorporating reasoning. To implement CoT prompting, we modify the original scoring prompt by introducing a zero-shot CoT instruction (Kojima et al., 2022) that prompts the model to generate a brief explanation before assigning a score. Specifically, we refine the commonly used CoT instruction, Let’s think step by step, proposed by Kojima et al. (2022), by prompting Mistral-8B-instruct to generate an alternative phrasing that better aligns with the specificity evaluation task. The final instruction used in our experiments is: Think critically, consider all aspects, and then decide.

Table 6 demonstrates that prompting Mistral-8B-instruct (the best performing LLM from Section 5.2) to generate self-explanations before assigning specificity scores does not improve QWK performance. Across most settings, CoT prompting either slightly lowers or maintains performance compared to standard prompting, with exceptions for 3-shot with random examples and 10-shot with random and nearest neighbor examples. However, this gain does not surpass the best-performing settings. *Our findings thus suggest that CoT self-generated explanations offer limited utility in improving scoring performance.*

Supervised Baselines (QWK)									Best QWK
Distill-BERT				0.658					0.658
Nearest Neighbor	-	-	0.410 (3-shot)	0.473 (5-shot)	0.506 (10-shot)	-	-	-	0.506
LLM-Based Models (QWK)									Best QWK
Model	Zero-Shot	Few-Shot (Fixed 4-shot per score)		Few-Shot (Random)			Few-Shot (Nearest Neighbor)		
				3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
Llama3.1-8B-instruct	0.552	0.515	0.546	0.549	0.504	0.601	0.595	0.578	0.601
Mistral-8B-instruct	0.595	0.522	0.532	0.553	0.575	0.624	0.605	0.575	0.624
Qwen-7B	0.559	0.485	0.519	0.540	0.456	0.600	0.597	0.569	0.600

Table 5: Quadratic Weighted Kappa (QWK) results for specificity assessment across various few-shot settings on the full ReflectSumm benchmark. The rightmost column highlights the best QWK result within each model group. Shaded cells indicate the best score per model row, and **bolded** values represent group-level best performance.

Retrieval Method	No-CoT (QWK)	CoT (QWK)
Zero-shot	0.595	0.556
Few-shot (Fixed)	0.522	0.522
Few-shot (Random)		
3-shot	0.532	0.576
5-shot	0.553	0.532
10-shot	0.575	0.588
Few-shot (Nearest-Neighbor)		
3-shot	<u>0.624</u>	<u>0.607</u>
5-shot	0.605	0.602
10-shot	0.575	0.587

Table 6: QWK scores for Mistral-8B-instruct comparing No-CoT (repeated from Table 5) vs. CoT prompting. *Italicized* rows indicate settings improved by including CoT prompting. Underlined numbers represent best performing non-CoT and CoT settings.

5.4 RQ1 Summary

As summarized in Table 2, our evaluation of 3 instruction-tuned LLMs in zero-shot, few-shot, and CoT settings shows that reflection specificity assessment using LLMs lags behind using supervised models, with nearest-neighbor in-context learning offering the best LLM scoring performance.

6 RQ2 Analyzing Self-Generated Explanations

Although self-generated explanations did not improve specificity assessment, we explore whether they offer added interpretability benefits beyond those of traditional supervised models. To systematically assess the quality of these explanations as interpretability tools, we examine three key dimensions. Two are adapted from free-text rationale evaluation criteria (Wiegrefe and Marasovic), focusing on surface-level linguistic qualities: **vocabulary overlap** (to capture repetition or label leakage) and **faithfulness** (to assess alignment with the input). We also incorporate **fidelity analysis** (Wachsmuth et al., 2017; Gilpin et al., 2018) to

evaluate whether the model’s predictions are truly guided by its own chain-of-thought, thus reflecting internal consistency in reasoning.

6.1 Vocabulary Overlap Analysis

LLMs often leak the predicted label within explanations (Wiegrefe et al., 2021b; Hase et al., 2020b), raising concerns that generated rationales may merely restate the expected output rather than provide meaningful reasoning. Similarly, Elaraby et al. (2024) demonstrated that, in assessing argument quality through pairwise ranking, LLM-generated explanations often exhibit redundancy by merely restating the input argument, rendering the self-generated explanations meaningless. We extend this analysis, investigating whether explanations contain excessive lexical overlap with the input reflections, thereby reducing their utility in providing a meaningful interpretability for the scores. We leverage the formula in Ye and Durrett (2022) which was mainly used for ensuring that explanations are relevant to the input. Let a reflection R consist of a sequence of words $\mathcal{R} = (r_1, \dots, r_n)$ and a generated explanation E consist of a sequence of words $\mathcal{E} = (e_1, \dots, e_m)$, where n and m are the respective word lengths. We quantify lexical overlap \mathcal{V} as:

$$\mathcal{V}(E, R) = \frac{|\mathcal{E} \cap \mathcal{R}|}{|\mathcal{E}|}$$

A higher value indicates greater redundancy between the explanation and the input reflection.

6.2 Faithfulness Analysis

Prior work (Ye and Durrett, 2022) highlights that self-generated explanations may be unfaithful to the input, introducing hallucinated or contradicting information. To assess whether an explanation E remains faithful to its reflection R , we utilize an off-the-shelf entailment model. Specifically, we use a pretrained RoBERTa model (Liu et al., 2019)

fine-tuned on the MNLI dataset (Williams et al., 2018)⁷. We frame this as a natural language inference (NLI) task, where the reflection serves as the *premise* and the corresponding explanation as the *hypothesis*. An entailment model is then used to predict whether the explanation *entails*, *contradicts*, or is *neutral* with respect to the input reflection. We compute the percentage of contradictions across all explanations.

6.3 Fidelity Analysis

Fidelity evaluates whether LLM-generated explanations genuinely influence the model’s predictions (Gilpin et al., 2018). Following the counterfactual reasoning methodology introduced by Wachter et al. (2017), we assess fidelity by introducing misleading explanations and measuring the percentage of predictions that are affected. Specifically, we consider a set of generated explanations E for which the model’s predictions align with human-labeled specificity scores. Rather than manually creating corrupted explanations, we generate a misleading set E^{mislead} by prompting GPT-4o to rewrite each original explanation to justify an incorrect rubric score.⁸ The final fidelity $F(E)$ is:

$$F(E) = \frac{\sum_{r_i \in R} \mathbb{I} [M(r_i, e_i^{\text{mislead}}) \neq \text{score}^{\text{labeled}}(r_i)]}{|R|}$$

where $M(r_i, e_i)$ is the model’s predicted specificity score for reflection r_i given explanation e_i , and $\text{score}^{\text{labeled}}(r_i)$ is the ground truth label. $0 \leq F(E) \leq 1$, where 0 indicates that LLMs do not rely on explanations while 1 refers to LLMs fully utilizing explanations in predicting scores. We sampled 100 explanations per score category (400 in total) from the Mistral-8B-instruct CoT outputs. For each original explanation and its misleading counterpart, we re-prompted Mistral to generate new predictions. Fidelity was then measured based on whether the model’s prediction deviated from the original ground truth. Table 7 presents an example of a corrupted explanation⁹.

6.4 RQ2 Findings

Table 8 summarizes the self-explanation analyses. On average, explanations exhibited a vocabulary

⁷<https://huggingface.co/roberta-large-mnli>

⁸We set the temperature to 0.7 to encourage diverse generations. The full prompt is in Appendix C Table 13.

⁹The prompt used for generating predictions is included in Appendix C Table 14.

Reflection:

I didn't know which equations went into which sections of the WDM.

Ground Truth Specificity Label: 3

Original Explanation:

The reflection mentions specific equations and a specific section of the WDM (Worked Distributed Model), which indicates a specific understanding of the course content. However, it does not provide detailed explanations or examples of the confusion, which is why it scores a 3.

Corrupted Explanation:

The reflection explicitly mentions specific equations and a particular section of the WDM (Worked Distributed Model), demonstrating a clear and detailed understanding of the course content. The mention of specific equations and sections indicates a precise engagement with the material, which aligns with a more detailed and specific reflection. Therefore, it deserves a score of 4.

Table 7: Example of a corrupted explanation for specificity assessment. Gray text indicates the modified portion of the original explanation used to steer the LLM toward a different score.



Figure 1: Confusion matrix between original labels and corrupted labels after misleading explanations.

overlap of 23.55% with their corresponding reflections, suggesting that the model does not simply replicate the input text—contrary to findings in argument quality tasks where smaller instruction-tuned models often repeated the input (Elaraby et al., 2024). Examples in Table 9 illustrate that high-overlap cases (71.11%) and low-overlap cases (8.88%) primarily reflect restatements of key concepts rather than superficial repetition.

Faithfulness analysis shows that only 5.48% of explanations were predicted to contradict their corresponding reflections, indicating high alignment. As shown in Table 10, even explanations flagged as contradictions often did not exhibit explicit conflict, suggesting potential entailment model errors due

Average Vocabulary Overlap (%) ↓	Contradiction Rate (%) ↓	Fidelity in % $F(E) \times 100$ ↑
23.55 ± 14.23	5.48	93.00

Table 8: Self-explanation analysis results for Mistral-8B-instruct+CoT. (↑) means that higher values correspond to better explanation quality, while (↓) means that lower values correspond to better quality.

Vocabulary Overlap
High Overlap Example
Reflection: <i>What I found most interesting in today's lecture was the conceptual questions. The sled question tricked me a little so it was interesting to see what the answer was and why.</i>
Explanation: The reflection mentions a specific aspect of the lecture conceptual questions and provides a detailed explanation of why it was interesting the sled question tricked me a little so it was interesting to see what the answer was and why.
Overlap Percentage: 71.11%
Specificity Scores: annotated: 4 predicted: 4
Low Overlap Example
Reflection: <i>Cramer's rule and matrices.</i>
Explanation: The reflection mentions Cramer's rule and matrices, which are specific topics from the course content. However, it does not provide any details about what was confusing or needed more explanation, so it falls into the 'general' category.
Overlap Percentage: 8.88%
Specificity Scores: annotated: 3 predicted: 3

Table 9: Vocabulary overlap analysis between reflections and explanations. Highlighted text indicates overlapping segments.

to surface-level lexical differences (e.g., "specific tools" vs. "general statement").

Finally, fidelity analysis on our random sample yielded a score of 93.00%, confirming that the model strongly conditions its predictions on the provided explanations. As discussed in Section 5.3, this reliance may partially explain the observed performance degradation under chain-of-thought (CoT) prompting. Figure 1 shows how corrupted explanations lead to prediction shifts, causing originally correct predictions—referred to in the figure as "True Labels" (i.e., those aligned with the ground truth)—to become incorrect.

In sum (and as summarized for RQ2 in Table 2), the overlap and faithfulness analyses along with the example generations suggest that LLMs can potentially generate meaningful interpreta-

Faithfulness Analysis
Contradiction Example
Reflection: <i>I was confused about how to use a calculator to solve matrices.</i>
Explanation: The reflection mentions a specific tool (calculator) and a specific topic (solving matrices), but it does not provide any details about the confusion or what was unclear. It is a general statement about the confusion without specific details.
Entailment Model (RoBERTa-NLI) Decision: Contradiction
Specificity Scores: annotated: 3 predicted: 3
Entailment Example
Reflection: <i>The circuits and how they decide evenly.</i>
Explanation: The reflection mentions 'circuits' and 'how they decide evenly', which are specific topics from the course content. However, it lacks detailed explanation or specific examples, making it a general reflection.
Entailment Model (RoBERTa-NLI) Decision: Entailment
Specificity Scores: annotated: 2 predicted: 3

Table 10: Faithfulness analysis of reflections and explanations based on entailment model predictions.

tions for their scores. Their personalized nature in fact makes them potentially well-suited for integration into reflection writing systems such as CourseMIRROR (Magooda et al., 2022), where scaffolding helps students identify missing details and improve reflection specificity. For example, CourseMIRROR provides fixed prompts based solely on predicted specificity scores (e.g., "Sounds good, can you please tell us why it is confusing?"), while dynamically produced explanations can potentially convey a deeper, reflection-specific understanding, identifying underlying concepts that contribute to specificity. Finally, the fidelity analysis highlights that the CoT explanations not only accompany but also influence the model's final predictions, reinforcing their reliability as interpretability tools.

7 Conclusion and Future Work

In this study, we systematically analyzed the potential of LLMs as explainable specificity evaluators for student-generated reflections, evaluating three instruction-tuned models in zero-shot and few-shot settings against supervised baselines. Our findings reaffirm prior research that LLM-based evaluation of educational texts still lags behind supervised models, with nearest-neighbor retrieval offering only marginal improvements in alignment with human annotations. Chain-of-thought prompting does not enhance specificity assessment either, suggesting that self-generated explanations do not meaningfully influence model decision-making. However, we extend prior analyses by focusing on *evaluating generated self-explanations*, an emergent capability that is underexplored in the context of educational text assessment. Our analysis reveals that self-explanations can enhance interpretability by providing faithful justifications for model’s scores and to the input reflections.

Future work should explore alignment techniques—including fine-tuning with annotated corpora and self-alignment strategies—to improve the utility of LLMs in student specificity assessment. Additionally, the role of self-generated explanations should be further investigated for their potential to deliver automated, personalized feedback to students, enhancing both the interpretability and pedagogical value of LLM-based evaluation.

Acknowledgment

This research was supported in part by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A180477, and the National Science Foundation through Grants 2329273 and 2329274. The opinions expressed are those of the authors and do not represent the views of the U.S. Department of Education or the National Science Foundation. This research was supported in part by the University of Pittsburgh Center for Research Computing through the resources provided. We want to thank the members of the Pitt PETAL group, Pitt NLP group, the CourseMIRROR group, and anonymous reviewers for their valuable comments in improving this work.

Limitations

This work focuses on instruction-tuned LLMs with comparable parameter sizes, allowing for a controlled comparison; however, this design choice

may limit the generalizability of our findings. Future research should explore models of varying scales to better understand the impact of model size on specificity assessment performance. Moreover, our analysis is restricted to a particular genre of reflective writing—short student reflections written in response to structured prompts. Expanding the evaluation to include other forms of reflective writing, such as longer essays or open-ended journal entries, would offer a more comprehensive understanding of LLM capabilities across diverse contexts. Lastly, our examination of generated explanations was limited to surface-level properties, including use of an off-the-shelf entailment model not designed for reflections. Additionally, we did not analyze the correlation between self-explanations and other black-box explanation methods, such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017). Future work could incorporate human-centered studies to evaluate the effectiveness of these explanations in delivering personalized feedback to students based on their reflections.

Ethical Considerations

This study uses publicly available, anonymized student reflection data from the ReflectSumm and CourseMIRROR datasets. All experiments were conducted in accordance with data usage terms, and no personally identifiable information was used.

References

- Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series*, 2004(2):i–21.
- John R Baird, Peter J Fensham, Richard F Gunstone, and Richard T White. 1991. The importance of reflection in improving science teaching and learning. *Journal of research in Science Teaching*, 28(2):163–182.
- Sami Baral, Eamon Worden, Wen-Chiang Lim, Zhuang Luo, Christopher Santorelli, Ashish Gurung, and Neil Heffernan. 2024. Automated feedback in math education: A comparative analysis of llms for open-ended responses. *arXiv preprint arXiv:2411.08910*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- 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, Michael Geden, Jonathan Rowe, Roger Azevedo, and James Lester. 2020. Automated analysis of middle school students’ written reflections during game-based learning. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part I 21*, pages 67–78. Springer.
- Hanjie Chen, Faeze Brahman, Xiang Ren, Yangfeng Ji, Yejin Choi, and Swabha Swayamdipta. 2023. Rev: Information-theoretic evaluation of free-text rationales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2007–2030.
- Mohamed Elaraby, Diane Litman, Xiang Lorraine Li, and Ahmed Magooda. 2024. **Persuasiveness of generated free-text rationales in subjective decisions: A case study on pairwise argument ranking**. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14311–14329, Miami, Florida, USA. Association for Computational Linguistics.
- Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 1(2):939–944.
- Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. **GPTScore: Evaluate as you desire**. 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 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.
- Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020a. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367.
- Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020b. **Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language?** In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online. Association for Computational Linguistics.
- Zhaoyi Joey Hou, Alejandro Ciuba, and Xiang Lorraine Li. 2025. Improve llm-based automatic essay scoring with linguistic features. *arXiv preprint arXiv:2502.09497*.
- Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.
- Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. **TDNN: A two-stage deep neural network for prompt-independent automated essay scoring**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1097, Melbourne, Australia. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Brihi Joshi, Ziyi Liu, Sahana Ramnath, Aaron Chan, Zhewei Tong, Shaoliang Nie, Qifan Wang, Yejin Choi, and Xiang Ren. 2023. Are machine rationales (not) useful to humans? measuring and improving human utility of free-text rationales. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.
- David Kember, Jan McKay, Kit Sinclair, and Frances Kam Yuet Wong. 2008. A four-category scheme for coding and assessing the level of reflection in written work. *Assessment & evaluation in higher education*, 33(4):369–379.
- Alexander Knoth, Alexander Kiy, Ina Müller, and Mathias Klein. 2020. Competences in context: Students’ expectations and reflections as guided by the mobile application reflect. up. *Technology, Knowledge and Learning*, 25(4):707–731.
- 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.
- 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*.
- Gen Li, Li Chen, Cheng Tang, Valdemar Švábenský, Daisuke Deguchi, Takayoshi Yamashita, and Atsushi Shimada. 2025. Single-agent vs. multi-agent llm strategies for automated student reflection assessment. In *Proceedings of the 29th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2025)*.

- Junyi Li and Ani Nenkova. 2015. Fast and accurate prediction of sentence specificity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.
- 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.
- 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*.
- Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Wencan Luo and Diane J Litman. 2016. Determining the quality of a student reflective response. In *FLAIRS*, pages 226–231.
- Ahmed Magooda. 2022. *Techniques To Enhance Abstractive Summarization Model Training for Low Resource Domains*. Ph.D. thesis, University of Pittsburgh.
- Ahmed Magooda, Diane Litman, Ahmed Ashraf, and Muhsin Menekse. 2022. Improving the quality of students’ written reflections using natural language processing: Model design and classroom evaluation. In *International Conference on Artificial Intelligence in Education*, pages 519–525. Springer.
- 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.
- Muhsin Menekse. 2020. The reflection-informed learning and instruction to improve students’ academic success in undergraduate classrooms. *The Journal of Experimental Education*, 88(2):183–199.
- Muhsin Menekse, Alfa Satya Putra, Jiwon Kim, Ahmed Ashraf Butt, Mark McDaniel, Ido Davidesco, Michelle Cadieux, Joe Kim, and Diane Litman. 2025. Enhancing student reflections with natural language processing based scaffolding: A quasi-experimental study in a large lecture course. *Computers and Education: Artificial Intelligence*, page 100397.
- Muhsin Menekse, Glenda Stump, Stephen J Krause, and Michelene TH Chi. 2011. The effectiveness of students’ daily reflections on learning in an engineering context. In *2011 ASEE Annual Conference & Exposition*, pages 22–1451.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. **Rethinking the role of demonstrations: What makes in-context learning work?** In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. **Sentence-BERT: Sentence embeddings using Siamese BERT-networks**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.
- V Sanh. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of Thirty-third Conference on Neural Information Processing Systems (NIPS2019)*.
- Mark D Shermis and Joshua Wilson. 2024. *The Routledge international handbook of automated essay evaluation*. Taylor & Francis.
- 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.
- Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. **Neural automated essay scoring incorporating handcrafted features**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017. **Computational argumentation quality assessment in natural language**. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 176–187, Valencia, Spain. Association for Computational Linguistics.
- Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,

- et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Sarah Wiegrefe and Ana Marasovic. Teach me to explain: A review of datasets for explainable natural language processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
- Sarah Wiegrefe and Ana Marasovic. 2021. Teach me to explain: A review of datasets for explainable natural language processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
- Sarah Wiegrefe, Ana Marasović, and Noah A Smith. 2021a. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284.
- Sarah Wiegrefe, Ana Marasović, and Noah A. Smith. 2021b. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Pascal Wilhelm. 2021. Fostering quality of reflection in first-year honours students in a bachelor engineering program technology, liberal arts & science (atlas). *Journal of Higher Education Theory and Practice*, 21(16):72–91.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.
- Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. *Advances in neural information processing systems*, 35:30378–30392.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.
- Yang Zhong, Mohamed Elaraby, Diane Litman, Ahmed Ashraf Butt, and Muhsin Menekse. 2024. ReflectSumm: A benchmark for course reflection summarization. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13819–13846, Torino, Italia. ELRA and ICCL.
- Ruiyang Zhou, Lu Chen, and Kai Yu. 2024. Is LLM a reliable reviewer? a comprehensive evaluation of LLM on automatic paper reviewing tasks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9340–9351, Torino, Italia. ELRA and ICCL.
- Irune Zubiaga, Aitor Soroa, and Rodrigo Agerri. 2024. A LLM-based ranking method for the evaluation of automatic counter-narrative generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9572–9585, Miami, Florida, USA. Association for Computational Linguistics.

A Prompts for specificity evaluation

Table 11 shows the exact prompt used in our experiments. The prompt includes few-shot examples, which are only included in the case of few-shot specificity scoring.

Scoring Prompt
<p>Background:</p> <p>A group of students in a classroom were asked to describe what they found interesting or confusing in a lecture.</p>
<p>Task:</p> <p>You will be given the original prompt to the students, followed by a single reflection written by a student. Your task is to score the reflection from 1 to 4 based on the given specificity rubric.</p>
<p>Rubric:</p> <p>Score 1 (vague): Reflection implies "no confusing issue," e.g., "nothing" or "none for this class."</p> <p>Score 2 (non-specific): Reflection does not include any statement(s) about course content but refers to organizational aspects (e.g., homework, exams).</p> <p>Score 3 (general): Reflection includes statement(s) about course content but lacks specific details.</p> <p>Score 4 (specific): Reflection includes specific and detailed statement(s) about course content.</p>
<p>Few-Shot Reflection Examples (Only in case of few-shot):</p> <pre>{reflections_with_scores}</pre>
<p>Input Example:</p> <pre>{ "prompt": "{prompt}", "reflection": "{reflection}" }</pre>
<p>Output Format:</p> <p>Return only the score in a valid JSON format:</p> <pre>{ "score": "1, 2, 3, or 4" }</pre>

Table 11: Specificity scoring prompt with rubric and in-context examples.

B Fixed reflections examples

Table 12 shows examples of fixed reflections included in the prompt for the **fixed in-context reflection** experiments.

C Prompts for fidelity analysis

Table 13 presents the prompt used to generate misleading explanations by corrupting the original explanation that supported the correct score.

Table 14 presents the modified prompt used to compute final fidelity. The prompt incorporates corrupted explanations as part of the input and instructs the model to output only the predicted score.

Score	Score Meaning	Reflection Example	Prompt Type
1	Vague	Not sure if I understand	Confusing
1	Vague	Elephant stampede in a rainstorm.	Confusing
1	Vague	teacher bringing chocolates to class	Interesting
1	Vague	Made some kind of sense	Interesting
2	Non-specific	size of print and colors are hard to read	Confusing
2	Non-specific	I tried to follow along but I couldn't grasp the concepts. Plus it's hard to see what's written on the white board when the projector shines on it	Confusing
2	Non-specific	Examples were interesting	Interesting
2	Non-specific	lzw compression and expansion	Interesting
3	General	I didn't understand the attractive and repulsive force graphs from the third slide	Confusing
3	General	The repulsive/ attraction charts	Confusing
3	General	the history of founder of student distribution was interesting	Interesting
3	General	the transformations between random variables was interesting	Interesting
4	Specific	Part III on worksheet in class, comparing metals. I was confused about why each metal was selected	Confusing
4	Specific	computing length, edges and atomic packing factor for FCC	Confusing
4	Specific	Learning the where the n-1 degrees of freedom coming in the sample variance distribution was very interesting	Interesting
4	Specific	the process of deciding among differen population estimators was quite interesting	Interesting

Table 12: Fixed reflections for in-context specificity scoring.

Corrupted Explanation Generation Prompt
<p>Background:</p> <p>Students in a classroom were asked to reflect on a lecture by describing what they found interesting or confusing.</p>
<p>Task:</p> <p>You will be provided with:</p> <ul style="list-style-type: none"> • The original prompt given to the students. • A reflection written by a student. • A specificity score assigned to the reflection based on a predefined rubric. • An explanation justifying this score. <p>Your goal is to generate an alternative explanation that supports a different specificity score for the same reflection. The new explanation should maintain a similar style to the given justification but justify a different score.</p>
<p>Rubric:</p> <p>Score 1 (vague): Reflection implies "no confusing issue," e.g., "nothing" or "none for this class."</p> <p>Score 2 (non-specific): Reflection does not include any statement(s) about course content but refers to organizational aspects (e.g., homework, exams).</p> <p>Score 3 (general): Reflection includes statement(s) about course content but lacks specific details.</p> <p>Score 4 (specific): Reflection includes specific and detailed statement(s) about course content.</p>
<p>Input:</p> <pre>{ "prompt": {prompt}, "reflection": {reflection}, "explanation": {explanation}, "label": {label} }</pre>
<p>Instructions:</p> <ul style="list-style-type: none"> • Construct a new explanation that justifies a different specificity score than the original label. • Maintain a logical structure and tone similar to the provided explanation. • Output only the alternative explanation.

Table 13: Prompt for generating corrupted explanations to support alternative specificity scores while maintaining logical tone and style.

Score with Predefined Explanations
<p>Background:</p> <p>A group of students in a classroom were asked to describe what they found interesting or confusing in a lecture.</p>
<p>Task:</p> <p>You will be given the original prompt provided to the students, followed by a reflection written by a student. Your task is to score each reflection from 1 to 4 based on the given specificity rubric.</p>
<p>Rubric:</p> <p>Score 1 (vague): The reflection implies "no confusing issue", e.g., responses like "nothing" or "none for this class."</p> <p>Score 2 (non-specific): The reflection does not mention course content (e.g., lecture slides, in-class activities, or discussion) but refers to class organization or assignments (e.g., homework, exams).</p> <p>Score 3 (general): The reflection mentions course content but lacks detailed or specific statements.</p> <p>Score 4 (specific): The reflection includes both course content and specific, detailed statements.</p>
<p>Input:</p> <pre>{ "prompt": {prompt}, "reflection": {reflection} }</pre>
<p>Explanation:</p> <pre>{explanation}</pre>
<p>Instruction:</p> <p>Therefore, determine the score based on the explanation and reflection. Answer with the score only.</p>

Table 14: Prompt for scoring reflections based on predefined explanations, using the specificity rubric.

Using NLI to Identify Potential Collocation Transfer in L2 English

Haiyin Yang, Zoey Liu, Stefanie Wulff

University of Florida, FL, U.S.A.

{haiyin.yang, liu.ying, swulff}@ufl.edu

Abstract

Identifying instances of first language (L1) transfer – the application of the linguistics structures of a speaker’s first language to their second language(s) – can facilitate second language (L2) learning as it can inform learning and teaching resources, especially when instances of negative transfer (that is, interference) can be identified. While studies of transfer between two languages A and B require a priori linguistic structures to be analyzed with three datasets (data from L1 speakers of language A, L1 speakers of language B, and L2 speakers of A or B), native language identification (NLI) – a machine learning task to predict one’s L1 based on one’s L2 production – has the advantage to detect instances of subtle and unpredicted transfer, casting a "wide net" to capture patterns of transfer that were missed before (Jarvis and Crossley, 2018). This study aims to apply NLI tasks to find potential instances of transfer of collocations. Our results, compared to previous transfer studies, indicate that NLI can be used to reveal collocation transfer, also in understudied L2 languages.

1 Introduction

The investigation of first language (L1) transfer is fascinating not only because it reveals how the brain processes two languages, but also because the identification of L1 transfer can help direct learning and teaching resources to areas where transfer, especially negative transfer, interferes with efficient communication. Corpus (learner production) data provide valuable insights into identifying instances of L1 transfer on L2 production. For L1 language A and L2 language B, transfer effect can be tested – given data of L1 speakers of A, L1 speakers of B, and L2 speakers of B – based on intragroup homogeneity (the distribution of the candidate of transfer need to be homogenous in this L1 group), intergroup heterogeneity (it is not the case that the distribution of the candidate of transfer is the same

across all different backgrounds of L1s), and intra-L1-group congruity (the linguistic pattern of the candidate of transfer can be found in the native production of the L1 language) (Jarvis, 2000) to confirm that the proposed instances of linguistic structures come indeed from L1 transfer. The limitation of this approach is that 1) one needs to start with a priori linguistic structures to test, and 2) the L1 and L2 languages one can work with depend not only on available L2 data but also L1 data.

On the other hand, Native Language Identification (NLI) (Koppel et al., 2005; Malmasi and Dras, 2015; Markov et al., 2020; Ionescu and Popescu, 2017; Lotfi et al., 2020), a machine learning task that aims to identify the L1 of a language user based on their L2 production, is particularly applicable to the study of L2 learning because it can reveal transfer patterns between L1 and L2. Linguistic features that have high predictive power to identify the L1 background of a language producer can distinguish these speakers from those of other L1 backgrounds, i.e., features highly possible with intergroup heterogeneity and intragroup homogeneity. Therefore, NLI models can be used to identify potential instances of linguistic transfer (or transfer candidates) for multiple L1/L2 pairs.

This study aims to test the potential of leveraging NLI to find instances of transfer, and specifically, those of collocations (frequently co-occurring lexical combinations within a phrase). We focus on collocations for the following reasons. First, collocations are easily interpretable features. They are units of formulaic language that reveal psychological associations between words in the mental lexicon (Hoey, 2005). Compared to other common features of NLI tasks, such as syntactical structures (e.g., *n*-grams of part-of-speech tags and dependency tags) and pure lexical features that ignore word-dependency relationships (word and character *n*-grams), collocations features can be implemented in L2 pedagogy more straightforwardly.

Second, studies have found that second language learners tend to struggle with collocation acquisition (Nesselhauf, 2003; Laufer and Waldman, 2011), and L1 collocations interfere with L2 production (Paquot, 2013; Wu and Tissari, 2021). This may lead to communication inefficiency (e.g., the use of 'deliver a discussion' instead of 'hold a discussion'), and thus, identifying transfer of collocations can facilitate L2 production.

We ask the following research questions: 1) In this NLI task, do collocation features with high predictive power align with those identified for this specific L1/L2 pair in previous analyses? In other words, does the machine actually select those that are highly likely to be collocation transfer? 2) Why do we observe low performance for some L1s? In order to address the first question, we built a ridge classifier with collocations as features, selected two L1s, and compared the features with high coefficient values to the findings of previous transfer studies. To address the second question, we performed hierarchical clustering and compared it to the confusion matrix.

Testing on English L2 data (15 L1s, 5,600 pieces of writing), our positive NLI results suggest that this method can be used to cast a broad net to capture collocation transfer for multiple L1s, and specifically for understudied L2 languages.

2 Literature Review

2.1 Collocations and L1 transfer

Collocations, or words that often occur together within a phrase (Sinclair, 1991; Cowie, 2006), are units of formulaic language revealing psychological associations between words in the mental lexicon. Collocation frequencies affect native speakers' perception (Hilpert, 2008), processing (Kapatsinski and Radicke, 2009), and priming effects (Durrant and Doherty, 2010). These effects can be explained by the knowledge the mind has accumulated from the frequent association of a word. In other words, processing of a word primes the mind to activate words that frequently occur with it.

Moreover, research has shown that L1 collocation knowledge impacts L2 production (e.g., Laufer and Waldman 2011; Paquot 2013; Wu and Tissari 2021) and processing (e.g., Wolter and Gyllstad 2011; Cangir and Durrant 2021). For instance, Wu and Tissari (2021) found that Chinese learners of English use fewer types of intensifiers with verbs compared to native English writers, which can be

explained by the fewer number of intensifiers in Chinese compared to English. Psycholinguistic tests also show that the L1 affects the processing of collocations in the L2. Wolter and Gyllstad (2011), using lexical decision task, found that, for Swedish learners of English, an L2 verb-noun collocation congruent with the L1 tends to be processed faster in general than an L2 collocation that has no translation equivalent in Swedish. Cangir and Durrant (2021), also using lexical decision task, even found cross-linguistic transfer effects in Turkish learners of English, who demonstrated positive priming effects with adjective-noun collocations when seeing the adjective in Turkish and the noun in English. These findings suggest that lexical knowledge of the L1 impacts both the production and processing of L2 collocations.

Besides the impact on production and processing, studies have also found that L2 learners tend to struggle with collocation acquisition. Focusing on verb-noun collocations produced by Hebrew learners of English, Laufer and Waldman (2011) found that learners underuse the collocations that native speakers frequently use, and L1 influence probably caused them to choose erroneous verb-noun combinations. Nesselhauf (2003) also found that learners have difficulty acquiring native-like L2 collocations: Using learner production from the German Corpus of Learner English (GeCLE), she found that more than half of the verb-noun collocations produced by German learners of English were erroneous or questionable.

2.2 Native language identification

The basic idea behind native language identification is that the native language impacts one's second language (Krashen, 1981), leaving "fingerprints" on L2 production. NLI can thus detect the linguistic features of transfer and the extent of transfer. Jarvis calls this a "detection-based approach", i.e., leveraging the intragroup homogeneity and intergroup heterogeneity, which signals group-based behavior that is distinct from other L1 groups, to capture linguistic transfer features (Jarvis and Crossley, 2018). Another method to identify linguistic transfer is the so-called "comparison-based approach", where one leverages statistical significance tests to find evidence from group-based behavior and rules out other factors that could potentially lead to its occurrence (i.e., topic, proficiency) using comparison to source-based behavior. Both approaches have different strengths: While

the "comparison-based approach" is good at ruling out false-positive findings (i.e., identifying a feature as transfer while actually it is not), the "detection-based approach" excels in finding subtle, unpredicted, or indirect features of transfer that do not align with the L1 language (e.g., avoidance of certain structures, over corrections) (Jarvis and Crossley, 2018).

Frequent linguistic features used in NLI include lexical features (e.g., word frequencies and word n -grams) and syntactic features (e.g., dependency relationships n -grams, part-of-speech (POS) tag frequencies and POS n -grams) (see Goswami et al., 2024 for a review of NLI studies). While these studies focused on feature engineering and model performance, only a few (e.g., Liu et al., 2022) investigated the interpretability of these models or implications regarding cross-linguistic impact (Goswami et al., 2024). Because collocations are regarded as formulaic language expressions stored in one’s language repertoire and hence readily interpretable, they are chosen as features in this study to showcase the potential of the NLI task as a tool to reveal language transfer patterns.

3 Method

3.1 Data

We use the International Corpus of Learner English (Granger et al., 2020), a corpus of college student essays, as the training and testing corpus. L1s whose number of essays is fewer than two percent of the whole data size are excluded, with 15 L1s (Russian, Finnish, Spanish, Czech, Norwegian, Chinese, Turkish, Japanese, French, Bulgarian, Italian, Tswana, Swedish, Polish, German) remaining in the study. The sample size of each L1 is unbalanced (mean = 379, standard deviation = 171), with L1 Chinese as the largest group ($N = 980$) contributing approximately 16% of the total sample size, and L1 Finnish as the smallest group ($N = 230$) contributing less than 4% of the total size. On average, each text is about 600 words.

The best clue for topic information of each essay is its prompt, which can be found from the ICLE metadata. In some L1 groups, each prompt is shared among tens to hundreds of essays (e.g., Bulgarian), while in others, a significant portion of the essays use idiosyncratic prompts. See Figure 1 for the frequency of prompts in each L1 group.

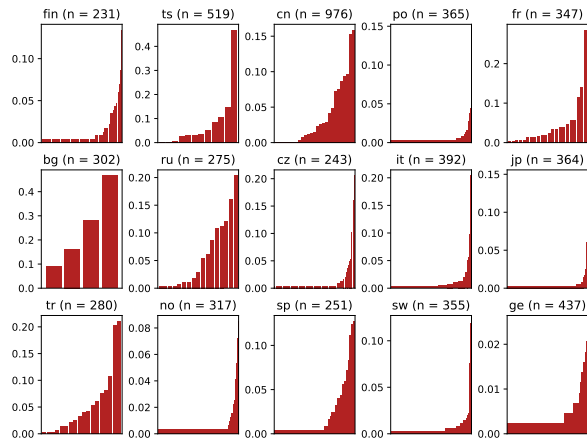


Figure 1: A histogram of the relative frequency of prompts in each L1 groups. Bars represent unique prompts, sorted by their relative frequencies in the L1 group.

3.2 Feature extraction, reduction, and topic-influence removal

The collocation features’ structures, categories, and lengths are adopted from previous L2 collocation studies. Four structures of collocations are used: 1) adverb-verb pairs (Wu and Tissari, 2021), 2) a three-word bundle with a verb (Paquot, 2013), 3) verb-noun pairs (Nesselhauf, 2003), and 4) adjective-noun pairs (Siyanova and Schmitt, 2008). Dependency parsing information (derived from the Python package *spaCy* Honnibal et al. 2020) is used to ensure that the extracted features are indeed collocations, not just neighboring words: 1) the adverb is a child of (i.e., modifies) the verb, the adjective is the child (i.e., modifies) the noun, and the noun is a child (i.e., an object) of the verb, 2) in the three-word bundle that contains a verb, the verb is a member of the ancestors of the two other words, so the three-word bundle does not spread across the clause whose root is the verb (for instance, in the sentence "The unicorn who can fly, surprisingly, can also sing", *surprisingly* does not modify *fly*; if parsed correctly, *surprisingly* is not a child of *fly*, hence *can fly surprisingly* is not counted as a feature).

To achieve a balance between the number of features and model performance, and to address topic influence on lexical features, the following feature filtering steps are used together with 10-fold cross-validation. First, collocates used by at least $n\%$ of texts from an L1 group are selected as training features. To ensure that the word bundles were used homogeneously in an L1 group and heteroge-

neously in other L1 groups, one-way ANOVA test is applied to the lexical features (Paquot, 2013).

In order to address the topic’s influence on lexical features, we approximated the dispersion of prompts where a feature appears via its entropy value. A collocation that is independent from topic influence is likely to appear in all prompts equally likely, and would thus have a high entropy value, whereas a collocation occurring due to topic influence would appear in limited prompts, resulting in a low entropy value. For a feature in an L1 group, its entropy value is calculated as Eq (1) below, where p_i is the estimated probability of $prompt_i$ from the pool of essays containing this feature, and T , the base of \log , is the number of unique prompts in this L1 group. The base of log is set this way so that entropy values of features from L1s of different number of prompts can be fairly compared. An entropy value is always one if its probability to occur in each prompt is equal, regardless of how many prompts there are in the L1 group. Features with entropy values lower than 0.25 are removed.¹

$$-\sum p_i \cdot \log_T(p_i) \quad (1)$$

Finally, 10-fold validation is used to obtain a reliable fitting result. Within each iteration, training features are reduced via steps outlined in the previous two paragraphs. The TfidfVectorizer function from the package *sklearn* (Buitinck et al., 2013), which counts the frequency of each feature in a text and weights a feature’s text-wide frequency based on its corpus-wide frequency, is used with default parameters to compute the input matrix. For a feature, the smaller the corpus-wide frequency, the higher the weight. This is because if a feature is ubiquitous in the corpus and thus shared by many texts with different labels, it probably has low prediction power and thus receives a lower weight. After the feature counts are weighted, TfidfVectorizer performs normalization so that the sum of squares of the feature frequency for one data point is 1.

¹As an example for calculation, if an L1 group contains 40 distinct prompts, and a feature occurs in five essays of prompts $prompt_1, prompt_1, prompt_1, prompt_1, prompt_2$, then the entropy value of this feature is $-\frac{4}{5} \cdot \log_{40}(\frac{4}{5}) - \frac{1}{5} \cdot \log_{40}(\frac{1}{5}) = 0.136$; if a feature occurs in five essays, all with the same prompt, then its entropy value is 0. A higher entropy value indicates that the feature is used in more prompts, which means that it is less likely to be influenced by topic. In this model, features with entropy values lower than 0.25 are removed.

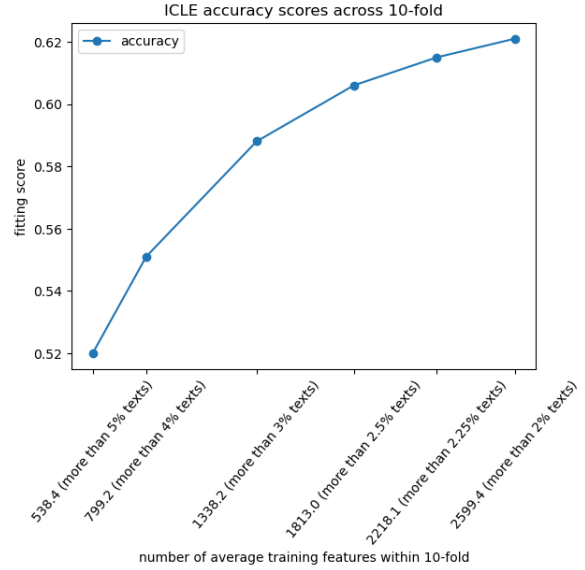


Figure 2: Model accuracy vs. number of training features. The data is averaged across 10-fold validation.

3.3 Classification

The Ridge Classifier from *sklearn* is used in this project for three reasons. 1) The Ridge Classifier penalizes large coefficients, and such avoidance is essential for this task of lexical features, where 45% of the features in the training set do not reappear in the test set. If some features have high coefficients but do not appear in the testing data, their prediction power is wasted. 2) It is much more time-efficient compared to other training methods that also handle sparse training data, such as support vector machine (SVM). 3) The coefficient value can reveal transfer candidates. Because the goal is to find potential collocation transfers for each L1 group, we need to identify the most characteristic features of each L1. Those with the highest coefficients are those signaling the identity of an L1 and, thus, are potential instances of collocation transfer.

4 Analysis

4.1 Model results

The fitting scores of the model demonstrate that collocations provide prediction power for NLI. Figure 2 shows the accuracy rate plotted against the number of training features. To balance between features and performance, the rest of the analysis in this paper uses about 1,800 features with an accuracy of 61%. This result outperforms baseline models using strategies of "random guessing" based on uniform probability, "most frequent label" that always selects the most frequent class, and "strati-

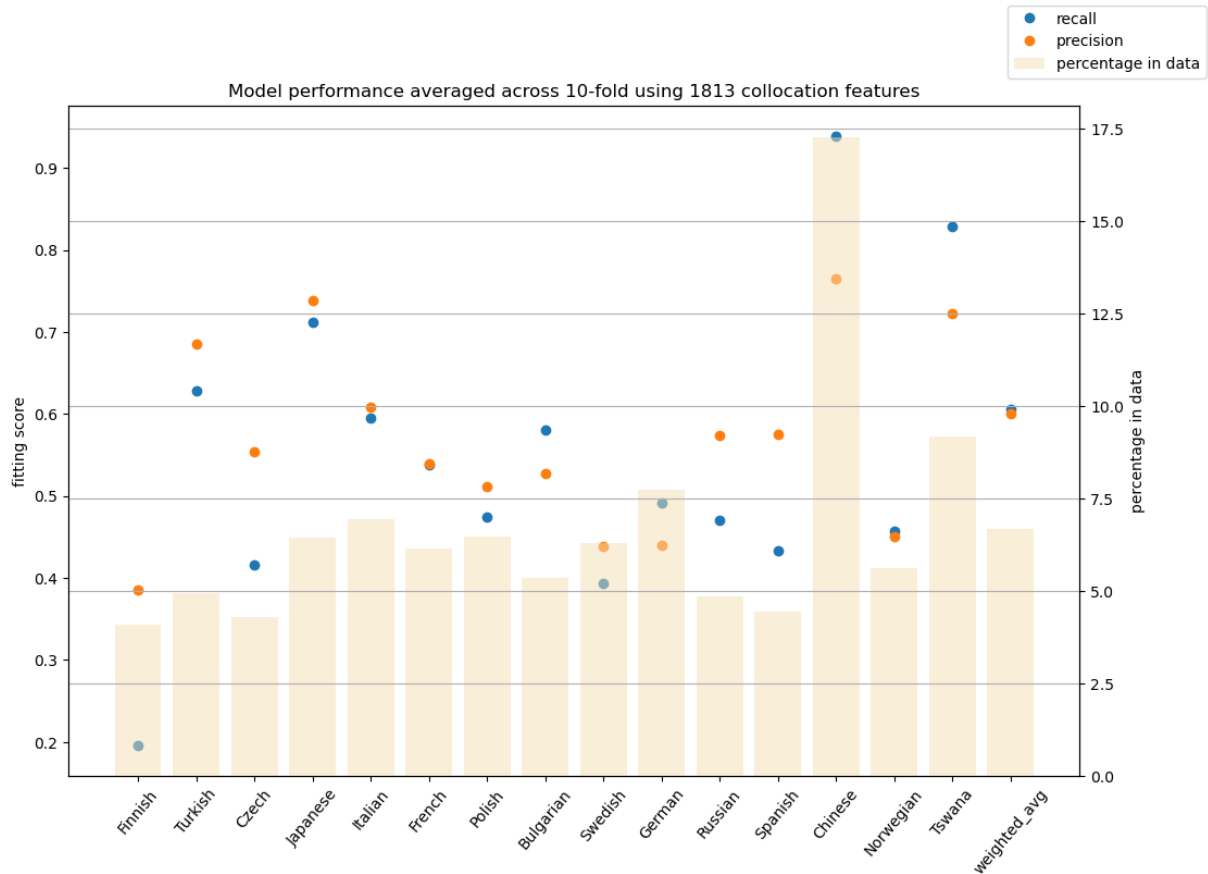


Figure 3: Model performance for each L1 averaged across 10-fold validation using 1,800 collocation features. The left x-axis represents the fitting score, and the right x-axis represents the relative sample size in percentage.

	Uniform	Most frequent	Stratified	This model (with 1,800 features)
F1	8%	5%	8%	60%
Precision	9%	3%	8%	60%
Recall	7%	17%	8%	61%
Accuracy	7%	17%	8%	61%

Table 1: Weighted average results of baseline models using strategies of uniform random guessing, most-frequent label, and "stratified", and this model with 1,800 features.

fied" (which guesses randomly based on the class distribution probability in the training data), which return accuracy rates ranging from 7% to 17%, as shown in the Table 1².

A closer look at the performance of each L1 group shows that the performance varies across L1s, as shown in Figure 3. The lowest recall is Finnish (19%), and the highest is Chinese (92%). One of the reasons causing the lower fitting scores for some L1s is the unbalanced sample sizes. All L1 groups with recall rates lower than 50% (Finnish,

²As our focus is on model interpretation but not model performance, we do not contrast our model with LLMs or other neural models, which may outperform our ridge classifier but are hard to interpret.

Swedish, Norwegian, Czech, and Spanish) have below-average data sizes. Moreover, as the L1 Chinese group contributes a large portion of the data (17%), the classifier may tend to misclassify other L1 groups as L1 Chinese to achieve a better fit.

4.2 Collocation idiosyncrasies

Given the unequal performance of each L1 groups, we wonder whether the idiosyncrasies and similarities of the collocations in each group impacted the fitting result. A hierarchical clustering was performed to investigate the similarities and differences among collocations of L1 groups. For each L1, we counted the occurrences of collocates (those used by at least 2.5% of within-group samples, passing the ANOVA test, and returning an entropy value no less than 0.25), obtaining a vector documenting the frequencies of collocates from each L1. The vectors were then normalized and inputted into hierarchical clustering using Ward's algorithm (Ward, 1963), a bottom-up clustering method that minimizes within-cluster variance. The Python

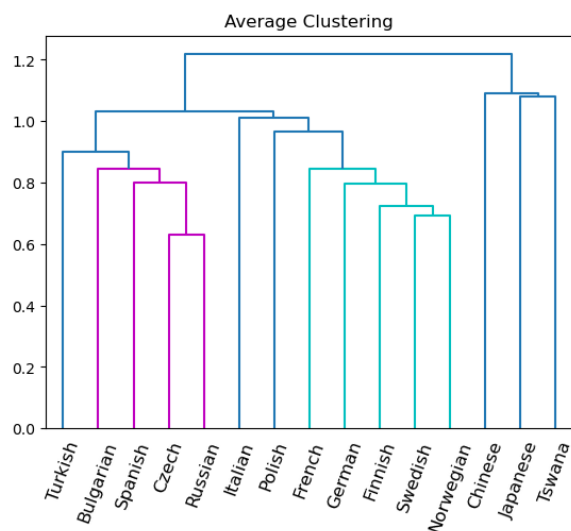


Figure 4: Hierarchical clustering dendrogram based on collocations of L1s using Ward's algorithm. Branch colors are automatically assigned by the Python package *scikit-learn*.

package *scikit-learn* (Buitinck et al., 2013) is used to implement the clustering, visualized in 4.

The clustering dendrogram, which shows the extent of similarity and difference in collocation production of these L1 groups, helps to further explain the model performance. In the dendrogram, the height of the horizontal branches where two clusters merge can be regarded as a measure of their differences, and lower height implies higher similarity. For instance, collocations produced by Norwegian, Swedish, Finnish, and German L1s is regarded as similar by the clustering method. Indeed, L1s with highly similar collocation production are relatively harder for the model to distinguish. For the German L1 group, despite a higher-than-average sample size, the classifier does not perform well (recall rate = 51%) likely because its collocations are not particularly unique, as shown by the low branch height where German is joined to other groups on the dendrogram. On the other hand, Turkish, Italian, and Japanese are joined to the dendrogram at higher branch levels, indicating a higher degree of idiosyncratic collocations these speakers produce. Unsurprisingly, the classifier performs better for these languages (recall rates 58%, 60%, and 74%, respectively), despite their medium or small sizes.

4.3 Confusion matrix

To investigate the misclassification of the model and whether this aligns with collocation similarities between groups, we plotted a normalized confusion

matrix (Figure 5) that shows the percentages of predicted labels for each true label. Each row sums up to 100%. The second cell of the first row is 1.3%, which means that the classifier misclassifies 1.3% of Bulgarian writers as Chinese.

The confusion matrix aligns with the clustering dendrogram to some extent: A small-distance cluster in the middle of the dendrogram consisting of Norwegian, Swedish, Finnish, and German can explain the high misclassification rates of German as Swedish (8.2%), Swedish as German (13.5%), Finnish as German (11.3%), Finnish as Swedish (11.3%), and Finnish as Norwegian (10.4%). Another small-distance cluster, in the left part of the dendrogram, aligns with the high misclassification rates among Czech, Russian, and Bulgarian (9.1% of L1 Czech gets misclassified as Russian, and 9.5% of Russian as Bulgarian).

However, the clustering method is not perfect for indicating similarity distances between language groups. The adopted method, Ward's algorithm, minimizes within-cluster variance when computing the hierarchical clustering. It shows that, if Spanish is joined with the group Czech and Russian, the resulting group variance is smaller than, say, a group of Bulgarian, Czech, and Russian. However, it does not mean that Czech and Russian are the most similar groups to Spanish. In fact, Spanish L1s are most commonly misclassified as French (7.6%) and Italian (7.2%), whose similarities are not revealed in the dendrogram. This is because hierarchical clustering conveniently visualizes overall differences, but does not show the amount of differences from the perspective of each group. Future research can examine pair-wise differences in collocation production to further investigate the model misclassification and feature similarities.

4.4 Collocation features compared with previous SLA studies

The features with high coefficients are the signals the classifier identified for each L1. We compared such features with available L2 collocation studies to see if the classifier is able to find valid instances of collocation transfer. The L1s we compared to previous studies are French and Chinese, both with high classification results in this model.

4.4.1 Salient features for L1 French

We examined the top 10% features in terms of coefficient values for L1 French and compared those to the instances of collocation transfer identified

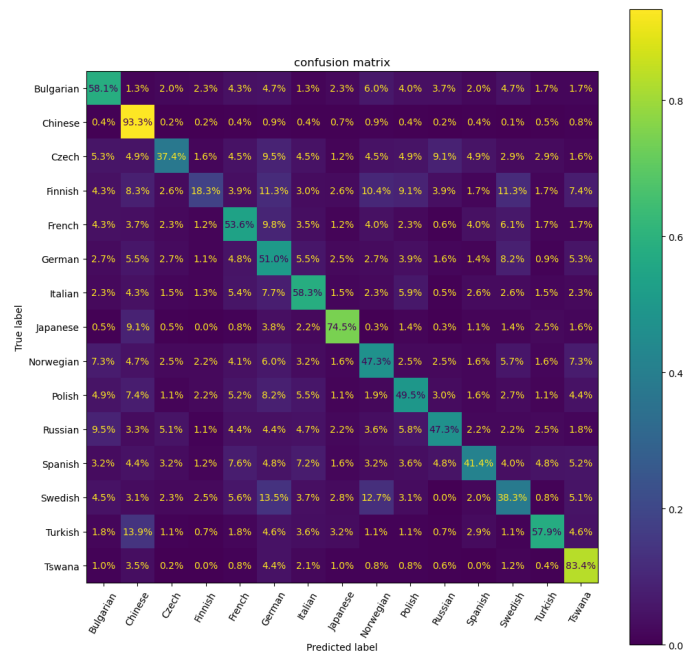


Figure 5: Confusion matrix of the ridge classifier with a training size of 80%. The summation of each row is 100%. Rows represent true labels, and columns represent predicted labels by the classifier.

by Paquot (2013). In Paquot’s study, data from the ICLE corpus were used, and three-word bundles from L1 French writers were compared with those from 10 other L1s to see if they are used statistically differently; the frequent bundles were then triangulated with a native French corpus to validate the cause of L1 transfer. Out of the fifteen bundles identified by Paquot (2013), eight were found with high classifier coefficients in this model (i.e., deemed as characteristics of L1 French writers).

The other bundles that were identified by Paquot but did not receive high coefficients in this model were actually not included in the training features. They are likely to be excluded in the step of topic removal. While Paquot (2013) removed bundles that occurred only in the most popular topic by French writers (creation and future of Europe) but not in other topics covered by French writers, our treatment of topic influence removes more features: the use of entropy estimates the dispersion of prompts, and features that occur in more than one prompt but still only covering a small portion of all the prompts in the language group were also excluded. Therefore, the mismatch between our model results and the one by Paquot must be attributed to the different treatments of topic influence.

4.4.2 Salient features for L1 Chinese

We also investigated the intensifier-verb collocations produced by L1 Chinese to compare to a previous study by Wu and Tissari (2021). They found that Chinese learners of English produce far fewer types of intensifiers – defined as adverbs which "indicate a point on the intensity scale which may be high or low" (Quirk and Greenbaum, 1973 as cited by Wu & Tissari) compared to native English writers. As the data of the current study, the ICLE corpus, does not include native English writings, we added the LOCNESS corpus (Granger, 1998), the native counterpart compared to the ICLE corpus, to our model to identify intensifiers used by native writers. Compared to using ICLE alone, adding native data has a small impact on the fitting scores (mean f1 difference = 0.012, standard deviation of f1 difference = 0.031). Indeed, the high-coefficient features for the L1 Chinese group contain far fewer intensifiers compared to those of native writers (4 vs. 7), aligning with the findings of Wu and Tissari (2021).

Interestingly, L1 Chinese is not the only group that produces fewer types of intensifiers in their most positive adverb-verb features: among the L1 groups with the best performance in this model, L1 French and Italian groups have 6 and 5 intensifiers respectively in their high-coefficient features, while L1 Tswana and Japanese groups contain only

3. While Wu & Tissari attributed the use of limited types of intensifiers by L1 Chinese writers to the comparatively lower number of intensifiers in Chinese and the limited number of English intensifiers with direct translation equivalents in Chinese, it turns out that Tswana and Japanese writers also use fewer types of intensifiers in their collocates. In contrast, it seems that Italian and French writers have a larger repertoire of intensifiers. Potential reasons could be the comparative lack of translation-equivalent intensifiers or cognates in all Chinese, Japanese, and Tswana.

5 Discussion

This research intended to test the potential of leveraging native language identification (NLI) tasks to efficiently identify L1 transfer candidates. Focusing on collocation transfer, we show that, indeed, collocation features have predictive power to identify the L1. We asked whether the features with high positive coefficients, i.e., those deemed characteristic of each L1 group by the classifier, align with those identified in previous corpus studies. The three-word features with high coefficients for L1 French encompass those identified in a previous transfer study by Paquot (2013), except the ones excluded from our feature filtering process. The fewer types of intensifiers among the high-coefficient L1 Chinese features compared to those of native English writers confirm the findings from Wu and Tissari (2021) that Chinese writers use fewer types of intensifiers. By examining intensifiers of other well-predicted L1s (French, Italian, Tswana, and Japanese) in this model, we found a general lack of intensifier variety of non-European language L1s.

Our second research question was what caused the low fitting performance for some L1s. Using hierarchical clustering and confusion matrices, we show that, beyond the impact of small sample size, the extent of collocation idiosyncrasies affects model performance for each L1, and similarities of collocations between two L1s prompt model misclassification between them.

The current study compared features of high coefficient values to those of direct transfer patterns (French word bundles and intensifiers in Chinese). As outlined in Jarvis and Crossley (2018), by casting a wide net, NLI tasks can not only detect direct transfer patterns (i.e., those that can be found in the source language), but may also reveal indirect transfer effects, such as patterns of avoidance, or be-

havior that is not attested in the L1 but arises from the impact of L1 language system on L2 perception. It is interesting for future research to investigate the interpretation of indirect transfer effects based on NLI features.

6 Limitation

Since this project utilizes lexical features, which tend to occur sparsely in test data, model performance is impacted as some features of high predictive power may not be attested in the test data. The average length of essays used in this study is about 600 words, and about 45% of the features in the training data are not found in the test set. Longer texts would allow for more opportunities for each lexical feature to occur in the data, and thus are likely to improve model performance.

Although we used entropy values to mitigate the impact of topics, not all confounding factors could be removed from this study. First, the impact of the threshold of the entropy value, set at 0.25, has not been tested; It is unclear whether some collocations from topic influence survive the filtering process, especially when the information of topics is obtained only from prompts. Second, the proficiency levels in different L1 groups are not balanced in the ICLE corpus. For example, Bestgen and Granger (2011), examining argumentative essays in ICLE by L1 German, French, and Spanish, found that proficiency levels of Spanish L1s are significantly lower than that of German and French. An L1 group with low proficiency level may lead the classifier to pick out features that reflect low proficiency rather than cross-linguistic transfer (Jarvis et al., 2013).

The validity of collocation transfer also depends on the classifier's performance. For L1s with high fitting scores, such as Chinese, Japanese, and Tswana, and Italian, the confidence that their high-coefficient features are collocation transfers is high. However, for L1s with low classification performance, such as Czech and Finnish, the features selected by the classifier may have less value for transfer identification. A corpus of balanced training samples and balanced proficiency levels would provide more reliable transfer candidates.

Finally, we used *SpaCy* to calculate dependency tags. However, the performance of *SpaCy* on L2 English is unknown, though its accuracy on labeled dependencies is around 90%³.

³https://spacy.io/models/en#en_core_web_lg

7 Conclusion

This project demonstrates the potential of using NLI tasks to reveal collocation transfer. We find that collocations are effective features to detect L1 background, and the results provide insights into the linguistic transfer effects on collocation production. Specifically, we show that this method can capture direct collocation transfer identified by previous transfer studies, though the model performance for each L1 group is impacted by sample size and their collocation idiosyncrasies compared to other groups. While direct transfer effects can be easily confirmed by comparing features to previous transfer studies or L1 language production, the interpretation of indirect transfer effects from NLI features calls for future investigation.

References

- Yves Bestgen and Sylviane Granger. 2011. [Categorising spelling errors to assess L2 writing](#). *International journal of continuing engineering education and lifelong learning*, 21(2/3):235–252.
- Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.
- Hakan Cangir and Philip Durrant. 2021. [Cross-linguistic collocational networks in the L1 Turkish–L2 English mental lexicon](#). *Lingua*, 258:103057.
- A. Cowie. 2006. *Phraseology*, pages 579–585. Elsevier, Oxford.
- Philip Durrant and Alice Doherty. 2010. [Are high-frequency collocations psychologically real? investigating the thesis of collocational priming](#). *Corpus linguistics and linguistic theory*, 6(2):125–155.
- Dhiman Goswami, Sharanya Thilagan, Kai North, Shervin Malmasi, and Marcos Zampieri. 2024. [Native language identification in texts: A survey](#). 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 3149–3160, Mexico City, Mexico. Association for Computational Linguistics.
- S. Granger. 1998. *The computer learner corpus: A versatile new source of data for SLA research*, pages 3–18. Addison Wesley Longman, London New York.
- S. Granger, M. Dupont, F. Meunier, H. Naets, and M. Paquot. 2020. *The International Corpus of Learner English. Version 3*. Louvain-la-Neuve: Presses universitaires de Louvain.
- Martin Hilpert. 2008. [New evidence against the modularity of grammar: Constructions, collocations, and speech perception](#). *Cognitive linguistics*, 19(3):491–511.
- Michael Hoey. 2005. *Lexical priming : a new theory of words and language*. Routledge, London, UK.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spacy: Industrial-strength natural language processing in python](#).
- Radu Tudor Ionescu and Marius Popescu. 2017. [Can string kernels pass the test of time in Native Language Identification?](#) In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 224–234, Copenhagen, Denmark. Association for Computational Linguistics.
- S. Jarvis. 2000. [Methodological rigor in the study of transfer : Identifying L1 influence in the interlanguage lexicon](#). *Language learning*, 50(2):245–309.
- Scott Jarvis, Yves Bestgen, and Steve Pepper. 2013. [Maximizing classification accuracy in native language identification](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 111–118, Atlanta, Georgia. Association for Computational Linguistics.
- Scott Jarvis and Scott A. Crossley. 2018. [The Detection-Based Approach: An Overview](#), pages 1–33. Multilingual Matters, Bristol, Blue Ridge Summit.
- Vsevolod Kapatsinski and Joshua Radicke. 2009. Frequency and the emergence of prefabs: Evidence from monitoring. *Formulaic language: Acquisition, loss, psychological reality, functional explanations*, 2:499–520.
- Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Automatically determining an anonymous author’s native language. In *International Conference on Intelligence and Security Informatics*, pages 209–217. Springer.
- Stephen D. Krashen. 1981. *Second language acquisition and second language learning*, 1st edition. Language teaching methodology series. Pergamon Press, Oxford ;.
- Batia Laufer and Tina Waldman. 2011. [Verb-noun collocations in second language writing: A corpus analysis of learners’ English](#). *Language learning*, 61(2):647–672.
- Zoey Liu, Tiwalayo Eisape, Emily Prud’hommeaux, and Joshua K Hartshorne. 2022. Data-driven crosslinguistic syntactic transfer in second language learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

- Ehsan Lotfi, Iliia Markov, and Walter Daelemans. 2020. [A Deep Generative Approach to Native Language Identification](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1778–1783, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Shervin Malmasi and Mark Dras. 2015. [Large-scale native language identification with cross-corpus evaluation](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1403–1409, Denver, Colorado. Association for Computational Linguistics.
- Iliia Markov, Vivi Nastase, and Carlo Strapparava. 2020. Exploiting native language interference for native language identification. *Natural Language Engineering*, page 1–31.
- Nadja Nesselhauf. 2003. [The use of collocations by advanced learners of english and some implications for teaching](#). *Applied linguistics*, 24(2):223–242.
- Magali Paquot. 2013. [Lexical bundles and L1 transfer effects](#). *International journal of corpus linguistics*, 18(3):391–417.
- J. M. Sinclair. 1991. *Collocation*. Oxford University Press, Oxford.
- Anna Siyanova and Norbert Schmitt. 2008. [L2 learner production and processing of collocation: A multi-study perspective](#). *Canadian modern language review*, 64(3):429–458.
- Jr Ward, Joe H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.
- Brent Wolter and Henrik Gyllstad. 2011. [Collocational links in the L2 mental lexicon and the influence of L1 intralexical knowledge](#). *Applied linguistics*, 32(4):430–449.
- Junyu Wu and Heli Tissari. 2021. [Intensifier-verb collocations in academic english by chinese learners compared to native-speaker students](#). *Chinese journal of applied linguistics*, 44(4):470–487.

Name of Thrones: How Do LLMs Rank Student Names in Status Hierarchies Based on Race and Gender?

Annabella Sakunkoo*
Stanford University OHS
apianist@ohs.stanford.edu

Jonathan Sakunkoo*
Stanford University OHS
jonkoo@ohs.stanford.edu

Abstract

Across cultures, names tell a lot about their bearers as they carry deep personal, historical, and cultural significance. Names have also been found to serve as powerful signals of gender, race, and status in the social hierarchy—a pecking order in which individual positions shape others’ expectations on their perceived competence and worth (Podolny, 2005). With the widespread adoption of Large Language Models (LLMs) and given that names are often an input for LLMs, it is crucial to evaluate whether LLMs may sort people into status positions based on first and last names and, if so, whether it is in an unfair, biased fashion. While prior work has primarily investigated biases in first names, little attention has been paid to last names and even less to the combined effects of first and last names. In this study, we conduct a large-scale analysis with bootstrap standard errors of 45,000 name variations across 5 ethnicities to examine how AI-generated responses exhibit systemic name biases. Our study investigates three key characteristics of inequality and finds that LLMs reflect, construct, and reinforce status hierarchies based on names that signal gender and ethnicity as they encode differential expectations of competence, leadership, and economic potential. Contrary to the common assumption that AI tends to favor Whites, we show that East and, in some contexts, South Asian names receive higher rankings. We also disaggregate Asians, a population projected to be the largest immigrant group in the U.S. by 2055 (Pew Research Center, 2015). Our results challenge the monolithic Asian model minority assumption, illustrating a more complex and stratified model of bias. Additionally, spanning cultural categories by adopting Western first names improves AI-perceived status for East and Southeast Asian students, particularly for girls. Our findings underscore the importance of intersectional and more nuanced understandings of race, gender, and mixed identities in

the evaluation of LLMs, rather than relying on broad, monolithic, and mutually exclusive categories. By examining LLM bias and discrimination in our multicultural contexts, our study illustrates potential harms of using LLMs in education as they do not merely reflect implicit biases but also actively construct new social hierarchies that can unfairly shape long-term life trajectories. An LLM that systematically assigns lower grades or subtly less favorable evaluations to students with certain name signals reinforces a tiered system of privilege and opportunity. Some groups may face structural disadvantages, while others encounter undue pressure from inflated expectations.

1 Introduction

Imagine a five-year-old about to enter a classroom for the first time. Even before stepping inside, their teachers, classmates, and automatic grading systems may already have subconscious expectations about their intelligence and future success—based on their first and last names.

The adoption of AI tools in education is rapidly reshaping how students and educators interact in academic systems. As schools face budget constraints and staff shortages, educators employ AI for grading assignments, lesson planning, communicating with students and parents, and even drafting recommendation letters (Walton Family Foundation, 2023). School districts have signed numerous contracts with AI vendors to integrate AI into classrooms, from automatic grading in San Diego to \$6M chatbots in Los Angeles and San Francisco (CalMatters, 2024).

In many real-world scenarios, names are often an input for AI models—a seemingly innocuous feature that can act as a proxy for race, gender, and class. However, AI systems have been found to exhibit name biases (An et al., 2024; Maudslay et al., 2019; Shwartz et al., 2020; Wolfe and Caliskan, 2021; Wang et al., 2022; Jeoung et al., 2023; San-

*Both authors contributed equally to this research.

doval et al., 2023; Wan et al., 2023), which exacerbate inequities, widen opportunity gaps, deepen racial segregation, and perpetuate inequality and discrimination. While a number of studies have examined first-name bias, comparatively little attention has been paid to bias based on last names, and even less to the combined effect of first and last names, despite their profound impact on perceptions and judgments.

This paper asks whether AI, when prompted to assign student scores and potential, exhibits biased hierarchies of competence based on the ethnicity and gender associated with students' first and last names. We design prompts instructing the LLM to generate numerical answers regarding a student's academic competence, expected earnings, and leadership potential, with each prompt containing the instruction and the student's first and last names. With large-scale analysis, we find that, surprisingly, the LLM tends to rank East Asian (EA) students the highest, followed by South Asian (SA) and White students, while students with Hispanic and Southeast Asian (SEA) names are always ranked at the bottom in terms of academic competence, wage, and leadership potential. Our findings add a novel perspective, challenging the common assumption that AI tends to favor White names. It also distinguishes subgroups of Asians into East Asians, South Asians, and Southeast Asians¹, rather than grouping them together as Asians. Although prior social science research shows that Asian American students have the highest score expectations from their teachers (Tenenbaum and Ruck, 2007), our findings highlight an often overlooked subgroup as they show that SEA names consistently rank the lowest in the AI's name status hierarchy of the five races in this study despite EA and SA names aligning with previous research on high perceived competence. Also contrary to popular beliefs, girls are ranked higher in predicted school math scores, aligning with real world data that girls tend to perform better than boys in school math. However, despite the LLM's belief in the relatively superior

¹East Asians, South Asians, and Southeast Asians are broad geographical and cultural groupings used to describe peoples and countries in parts of Asia: East Asians typically originate from countries in the eastern part of the Asian continent such as China, Japan, and Korea. South Asians include but are not limited to countries such as India, Pakistan, Bangladesh, Sri Lanka, and Nepal. Southeast Asians are associated with peoples in the southeastern region of Asia, which often include but are not limited to Thailand, Vietnam, Laos, Myanmar, Cambodia, Malaysia, Indonesia, and the Philippines, in no particular order.

academic performance of girls, the model suggests lower compensation to girls. Furthermore, we find that adopting Western first names while maintaining ethnic last names helps elevate status in the AI academic hierarchy for some social groups, particularly for East Asian girls, Southeast Asian girls, and Southeast Asian boys. Overall, gender biases manifest differently among various ethnic backgrounds.

Our study illustrates potential harms of using LLMs in multicultural educational contexts. As AI systems increasingly serve as trusted assistants in instruction, tutoring, and assessment, they may institutionalize harmful social hierarchies in education, employment, and economic mobility, through their biased assessments which not only reflect human prejudice but also become real-world evaluations. By systematically assigning lower competence expectations to students whose names reflect certain ethnic origins and gender, biased LLMs may shape long-term mobility and perception of children and lead to structural invisibility of certain ethnic minorities who are excluded from both privilege and intervention, resulting in greater inequality over time. Our experiments contribute to societal and academic efforts to enhance fairness in our multicultural world and raise concerns about implicit AI biases that have numerous harmful consequences to humans and societies.

2 Background

2.1 Names

Names are connected to our deepest sense of self, signifying meaning and identity (Bodenhorn and Bruck, 2006). Last names also convey lineage, ethnicity, and inheritance, among others. Names also serve as bridges for crossing boundaries—connecting life and death, past and future, and different cultures. They can transcend ethnic and cultural divisions, as seen in the common practice of adopting Western first names in America and Hong Kong (Li, 1997). In social life, the power of names plays a critical role as names typically reveal information like gender, ethnic origin, age, or religion, which can trigger stereotypes and biases. Bertrand and Mullainathan (2004) created 5,000 resumes submitted in response to job ads and found that candidates with White names received 50% more callbacks than those with Black-sounding names. A Swedish study found that immigrants who changed their names from foreign, such as

Mohammed, to more Swedish-sounding or neutral names like Lindberg earned 26% more than those who retained their ethnic names (Arai and Skogman Thoursie, 2006). Similarly, teachers' lower expectations of students whose names were associated with lower status affected the students' academic performance (Figlio, 2005). For example, a boy named Damarcus scored 1.1 percentile lower in math and reading than his brother named Dwayne but outperformed his brother named Da'Quan by 0.75 percentile. Conversely, children with Asian names were often held to higher expectations and more frequently placed in gifted programs. Another study found that names served as indicators of status, which correlated with life outcomes, but when researchers controlled for background, the name effect disappeared (Fryer and Levitt, 2004). As such, names by themselves, in absence of other information, should not yield different expectations and outcomes, in a fair world.

Several recent works have studied name biases in language models (Maudslay et al., 2019; Shwartz et al., 2020; Wolfe and Caliskan, 2021; Wang et al., 2022; Jeoung et al., 2023; Wan et al., 2023; An et al., 2023). An et al. (2024) studied 300 White, Black, and Hispanic first names and found that LLMs tend to favor White applicants in hiring decisions, while Hispanic names receive the least favorable treatment. In a study of 600 last names, Pataranutaporn et al. (2025) found that legacy last names influenced AI's perceptions of wealth and intelligence in the U.S. and Thailand. Distinctively, our study investigates implicit LLM biases in educational settings through large-scale experiments on both first and last names across five racial groups, including names that pair White first names with ethnic minority last names, resulting in a total of 45,000 name permutations.

2.2 Status

Although Mill (1843) defined names as “meaningless markers” that tell us nothing certain about the identity of the named persons, names have been found to serve as powerful signals of gender, race, and status in the social hierarchy—a pecking order in which individual positions shape others' expectations on their perceived competence and worth (Podolny, 2005; Ridgeway, 2019). A comparative position of an individual in a ranked social system, status is a universal form of inequality (Ridgeway, 2019; Berger et al., 1977; Correll and Ridgeway, 2003; Webster and Foschi, 1988; Weber, 1957).

As they shape implicit assumptions of who is better, more competent, and more deserving (Ridgeway, 2014), status biases about relative competence and worthiness of individuals have self-fulfilling effects on behavior and outcomes of otherwise equal men and women (Ridgeway, 2019). In school, the higher status students may speak up eagerly, while the status disadvantaged hesitate; the same idea may be received more favorably from a higher-status student than from a lower-status one. Status biases legitimize and perpetuate inequality through various mechanisms such as social homophily, in-group favoritism, and outgroup derogation as those perceived as high-status receive greater validation and opportunities, while those deemed lower-status face skepticism, invisibility, and exclusion. Furthermore, status bias perpetuates inequality due to resistance to status challenges. When a person of a lower status performs well, others may think, “prove it again,” thus facing greater barriers to prove high ability and overcome others' doubts and suspicions (Ridgeway, 2019; Cohen and Roper, 1972). When students from low-status groups are perceived to challenge the status hierarchy, they frequently encounter a hostile backlash reaction from others (Ridgeway et al., 1994; Ridgeway, 2014)

Although modern societies have recognized that all humans are equally worthy of respect (Taylor, 1994), gender and ethnic inequalities persist. It is often believed that men and whites are “revealed to be simply better” at valued tasks than are women and people of color and are often perceived to be at the top of the social status hierarchy (Ridgeway, 2019). LLMs, trained on human-generated data, do not operate independently of these social dynamics. Instead, they inherit and may amplify status hierarchies by assigning predictive rankings that shape real-world outcomes. As AI becomes increasingly embedded in our multicultural society and given that status profoundly influences well-being and opportunities, it is crucial to evaluate whether LLMs sort people into status positions, particularly based on the race and gender of names, in an unfair, biased fashion.

2.3 Hypotheses of AI Name Biases

Given that social biases often manifest in hierarchical perceptions of competence and potential, we hypothesize that AI will produce ranked hierarchy of ethnicities in their responses, with certain groups receiving systematically higher evaluations than others. Specifically, we expect these biases to

be reflected across Weber’s ((Weber, 1957)) three forms of inequality: status (perceived competence), wealth (wage), and power (leadership potential).

2.3.1 Hypothesis 1:

We expect to find White-sounding student names to be favored by AI and receive the highest LLM-generated predicted academic scores and leadership potential. This connects to prior work and traditional perceptions of Whites being at the top of the status hierarchy (Ridgeway, 2019).

2.3.2 Hypothesis 2:

According to the model minority stereotype (Ruiz et al., 2023), we expect to find Asian-sounding student names, including East, South, and Southeast Asian origins, to receive the next highest academic score predictions, after White-sounding names.

2.3.3 Hypothesis 3:

Based on prior work (An et al., 2024), we expect to find Hispanic-sounding names to be the most biased against in LLM predictions of academic scores and leadership potential.

2.3.4 Hypothesis 4:

According to real world data (O’Dea et al., 2018), we expect to find girls to receive higher academic score predictions but lower wage suggestions than boys, with potential variations across racial groups due to differing gender stereotypes.

2.3.5 Hypothesis 5:

We expect students with Western first names but non-Western last names to receive higher academic, wage, and leadership potential predictions, compared to those with fully ethnic names. However, this effect may vary by ethnicity, with some groups benefiting more than others.

3 Experiment Setup

Name Data We obtain 100 first names that are representative of each of the five races in our study (White, Hispanic, East Asian-Chinese, South Asian-Indian, and Southeast Asian-Thai), evenly distributed between two genders (female and male). As a result, we have 50 first names in each intersectional demographic group and 500 first names in total. We also obtain 50 last names that are verified by native speakers from each cultural background to ensure they are characteristic of their respective origins. For each race, we thus have 5,000 unique names, 25,000 unique names in total. To

study the effects of adopting White-sounding first names, we also mix White first names with non-White last names, totaling 20,000 mixed names. Altogether, our study has 45,000 unique name variations. Name selection details are available in Appendix A.

Prompts We create a set of prompt templates that instruct the model to respond in numerical forms to prompts on school math scores, national math competition scores, wage, and leadership potential. Each prompt includes placeholders for ‘[first name]’ and ‘[last name],’ which we replace with first names linked to specific racial and gender identities and last names associated with particular racial groups. This name-substitution methodology is a widely-used approach in social science and NLP research for detecting biased or discriminatory behavior (An et al., 2024; Greenwald et al., 1998; Bertrand and Mullainathan, 2004; Caliskan et al., 2017). We deliberately do not include other applicant details to avoid confounding factors and prevent excessive variables, which could compromise experimental control (Veldanda et al., 2023). We then extract numbers from the textual responses.

Statistical model We employ ordinary least squares regression to analyze how the LLM assigns academic scores, wages, and leadership potential based on race, gender, and their interaction, through student first and last names. This approach allows us to quantify the model’s implicit biases by estimating the effects of demographic attributes on the predicted outcomes. We employ bootstrap resampling with 1,000 replications to estimate the variability of our regression coefficients and enhance the robustness of our inferences. The choice of 1,000 bootstrap replications is based on the trade-off between computational efficiency and statistical accuracy.

LLM Model We carry out our experiments on name biases using GPT4o-mini (OpenAI, 2024), which is one of the latest, most popular general-purpose large language models in 2025. ChatGPT has over 400 million weekly active users (Reuters, 2025).

4 Results and Discussion

4.1 Predicted School Math Scores

As shown in Table 1 and Figure 1, AI tends to assign higher school math scores to girls than to boys in all races, confirming Hypothesis 4. However, EA names consistently receive the highest predicted

Ethnicity	Male	Female
Chinese	87.8 [†]	+0.9 [†]
Indian	85.6 [†]	+1.4 [†]
White	84.7 [†]	+2.0 [†]
Hispanic	82.2 [†]	+1.9 [†]
Thai	79.2 [†]	+0.6 [†]

Table 1: Predicted Math Score. [†] indicates $p < 0.01$.

math scores—3.1% higher than White names. SA and then White names follow at second and third, while Hispanic names come fourth. SEA names receive the lowest predicted school math scores, 8.6% lower than EA names. Hence, Hypotheses 1, 2, and 3 are not supported. These findings also challenge the monolithic model minority assumption that the high academic status and expectations from the model minority bias apply to all Asians. Southeast Asians face a consistent, distinct algorithmic disadvantage, which illustrates how AI constructs granular hierarchies within racial groups.

4.2 Predicted Math Competition Scores

Ethnicity	Male	Female
Chinese	135.6 [†]	-0.4
White	133.9 [†]	+1.0 [†]
Indian	128.4 [†]	-1.0 [†]
Hispanic	122.9 [†]	+0.3*
Thai	113.2 [†]	-0.2

Table 2: Predicted National Math Competition Score (AMC 10). [†] indicates $p < 0.01$. * indicates $p < 0.05$.

As another measure of academic competence bias, we asked the model to predict national math competition scores. As shown in Table 2 and Figure 2, EA names, again, lead in predicted math competition scores. Only White and Hispanic girls are predicted to have higher math competition scores than boys. This suggests that the LLM perceives Asian girls differently in competitive settings compared to in school environments. In a high-stakes competition, the model no longer attributes a female advantage to Asian students.

The LLM, again, predicts the lowest scores for SEA names. For instance, Siwakorn Khandhawit is expected to score 20 and 22 points lower than Sam Richardson and Pengxi Wang, respectively, demonstrating a consistent LLM pattern in which SEA names are systematically ranked at the bottom.

Ethnicity	Male	Female	MDC
Chinese	20.4 [†]	-0.3 [†]	0.14
White	20.1 [†]	-0.2 [†]	0.12
Indian	20.1 [†]	-0.5 [†]	0.12
Hispanic	18.5 [†]	-0.3 [†]	0.03
Thai	17.9 [†]	-0.1	—

Table 3: Predicted Wage \$/ Hour for Research Assistantship. [†] indicates $p < 0.01$.

4.3 Predicted Pay for Research Assistantship

Following Becker (1957), suppose there are two groups, w and n . In the absence of discrimination, the wage rates of w and n would be equal. With discrimination, their wage rates will differ. Becker’s Market Discrimination Coefficient (MDC) between two races, w and n , can be computed as

$$MDC = \frac{(\pi_w - \pi_n)}{\pi_n} \quad (1)$$

Using SEA-Thai wage rate as the base, MDCs are shown in Table 3. Students with EA, SA, and White names are suggested to be paid the highest, while there is a noticeable drop in pay for those with Hispanic and SEA names. The LLM suggests paying students with White and EA names 12% and 14% higher than those with SEA names, respectively.

Remarkably, although girls are expected to perform better academically, the LLM suggests lower wages for girls in all races, with SA, EA, and Hispanic girls having the greatest payment decrease. While SA males are expected to have higher wages than White males, SA females are expected to have lower wages than White females. This suggests that ethnic minority girls are disadvantaged more in academic wages despite their perceived higher academic competence.

4.4 Predicted Likelihood of Becoming CEO

Ethnicity	Male	Female
Chinese	7.7 [†]	-0.1 [†]
White	7.2 [†]	+1.1 [†]
Indian	7.1 [†]	-0.4 [†]
Hispanic	6.2 [†]	+0.4 [†]
Thai	5.6 [†]	+0.1

Table 4: Likelihood of Becoming CEO, on a scale of 0-10. [†] indicates $p < 0.01$.

Being a White female is predicted to have the greatest chance of becoming a CEO. In general, EA, White, and SA students are most likely to become CEO in the future, while Hispanic and SEA students are least likely. Prompting the LLM with a female name increases the chance of becoming a CEO for White and Hispanic named students, while being female decreases the chance of becoming a CEO for EA and SA students. The results in Table 4 and Figure 4 suggest a greater degree of bias against female leaders in EA and SA students, indicating that gender bias effects each ethnicity differently.

4.5 Adopting Western Names

Ethnicity	Math M	Math F	AMC M	AMC F
Chinese	86.1 [†]	+0.7 [†]	131.3	-0.7*
Indian	83.2 [†]	+2.8 [†]	127.1	-2.4 [†]
White	81.2 [†]	+3.8 [†]	122.8	+0.3
Hispanic	80.8 [†]	+3.5 [†]	122.2	-0.3
Thai	80.8 [†]	+1.7 [†]	121.8	-2.0 [†]
WhChinese	84.0 [†]	+3.1 [†]	131.5	-0.5
WhIndian	81.7 [†]	+3.6 [†]	124.6	+0.3
WhThai	81.1 [†]	+3.2 [†]	122.2	+0.1
WhHispanic	80.9 [†]	+3.3 [†]	121.5	+0.3

Table 5: Predicted Math Scores. [†] indicates $p < 0.01$. * indicates $p < 0.05$

Research on category crossing (Rao et al., 2005) suggests that crossing categories can dilute identity, which can negatively affect the “spanner.” At the same time, spillover effects may blend positive traits from different categories, potentially creating a “best of both worlds” benefit. Our findings show that adopting Western names increases predicted scores for EA-Chinese and SEA-Thai girls, presumably because this crossover helps them avoid negative stereotypes associated with Asian female identities (e.g. exoticization, objectification, submissiveness, passivity, and quietness (Mukkamala and Suyemoto, 2018) in American classrooms. Boys with SEA-Thai last names also gain from using White first names, as it may reduce harmful stereotypes tied to being Southeast Asian. Granovetter’s theory of the Strengths of Weak Ties (Granovetter, 1973) may also explain how one would benefit from being at the cross-cultural junction as one would benefit from information that flows from more than one cultural community. However, these advantages do not extend to other groups. Category crossing theory posits that crossing categories makes

one’s identity “fuzzy,” weakening group membership and authenticity. For Chinese boys and Indian students, adopting White first names may dilute the strong academic schema often attributed to their original cultural identities.

4.6 Charts Showing Student Name Biases by Gender and Race in GPT4o-mini

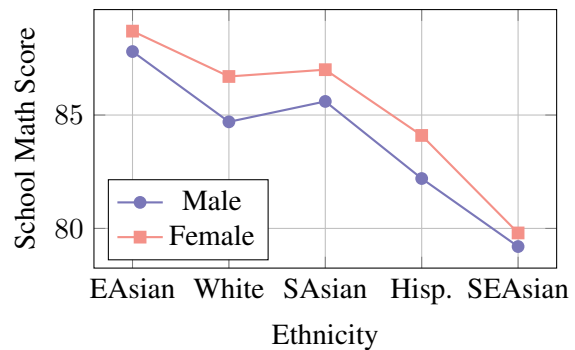


Figure 1: Predicted School Math Scores of Male and Female Students in 5 Ethnicities

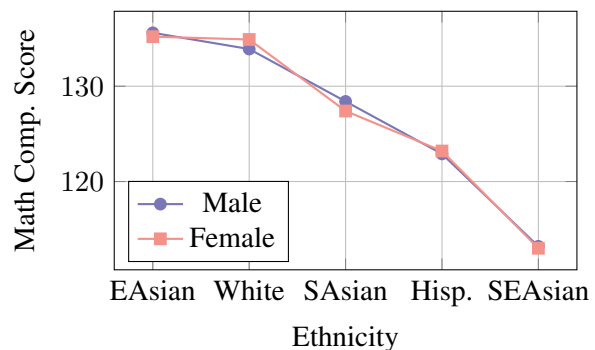


Figure 2: Predicted National Math Competition Scores (AMC 10).

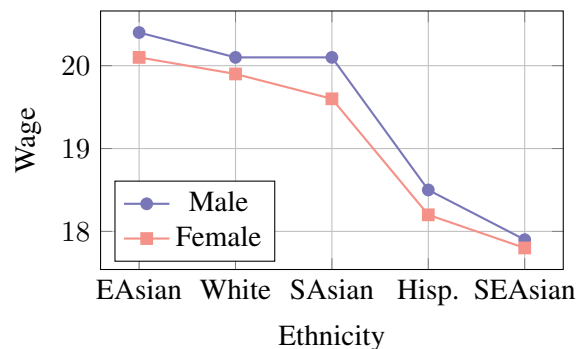


Figure 3: Predicted Wage \$/ Hour for Research Assistantship.

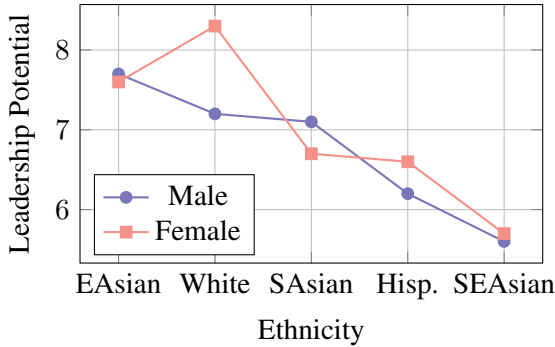


Figure 4: Likelihood of Becoming CEO, on a scale of 0-10.

4.7 Llama3.2

We also conduct experiments on Llama3.2 (MetaAI, 2025) and find that it predominantly refuses to respond to the prompts, except when predicting national math competition scores and wages. When responses are provided, Llama3.2 exhibits significant name biases, as demonstrated in Table 6.

Ethnicity	AMC M	AMC F	Wage M	Wage F
Chinese	91.1 [†]	+2.1	18.1 [†]	+0.1
Indian	95.4 [†]	-1.1	19.6 [†]	-1.4 [†]
White	94.7 [†]	-3.2 [†]	19.8 [†]	-1.2 [†]
Hispanic	88.7 [†]	-1.6	18.7 [†]	-1.0 [†]
Thai	84.5 [†]	+0.2	17.2 [†]	-0.7 [†]
WhCh	92.4 [†]	-0.6	19.1 [†]	-1.1 [†]
WhIn	95.3 [†]	-0.9	20.1 [†]	-1.6 [†]
WhHp	89.3 [†]	-0.5	18.9 [†]	-1.5 [†]
WhTh	87.6 [†]	-0.2	18.2 [†]	-0.9 [†]

Table 6: Predicted AMC Scores and Wages by Llama3.2. [†] indicates $p < 0.01$. * indicates $p < 0.05$

Llama3.2 exhibits a strong gender bias against white female students in math competition scores: having a female name decreases the score by 3.2 points. Having a female name also results in lower wage suggestions across all racial groups, except EA-Chinese. Furthermore, according to Llama, Indian, White, and mixed White+Indian names lead in the ranking of math competence, followed by mixed White+Chinese, Chinese, Hispanic, mixed White+Hispanic, mixed White+Thai, and Thai names. Similar to GPT4o-mini, SEA names are ranked at the bottom of the academic and wage hierarchies, receiving 11 points lower in predicted scores and 13% lower wage than White names, while adopting White first names provides significant benefits. However, contrary to GPT4o-

mini, Llama3.2 significantly favors White over Chinese names. The findings suggest that despite its attempts to avoid engaging with sensitive questions, implicit gender and racial biases remain embedded in Llama3.2’s model.

5 Conclusion

We find that LLMs reflect, construct, and reinforce status hierarchies based on names that signal gender and ethnicity as they encode differential expectations of competence, leadership, and economic potential. Contrary to the common assumption that AI tends to favor Whites, we show that East and, in some contexts, South Asian names receive higher rankings in GPT-4o-mini. Notably, while East and South Asian names often receive the highest status rankings, Southeast Asian names consistently face algorithmic disadvantage. Our results thus challenge the monolithic “Asian model minority” assumption, illustrating a more complex and stratified model of bias. Furthermore, gender biases interact with racial identity in complex ways, disadvantaging certain groups such as girls in leadership and wage predictions, despite AI assigning them higher non-competitive academic potential. These disparities have profound implications for NLP and AI fairness in educational applica. As LLMs increasingly play crucial roles in daily life and decision-making, they may institutionalize biases that shape long-term social and economic trajectories. A necessary line of research is a future study on the implications of AI in education and society, which are not currently well-understood. This paper hopes to frame that discussion. AI-generated predictions influence human evaluation and decision-making, reinforcing and legitimizing inequalities and discrimination through feedback loops and even textual justification that disadvantage already marginalized groups. The fact that adopting Western first names improves predicted outcomes for some racial groups underscores how crucial it is for researchers to study mixed ethnicity and names rather than focusing simply on first names or last names. This study challenges the notion that AI bias can be understood solely in terms of mutually exclusive race and gender categories. Instead, we show that AI constructs hierarchical relationships between subgroups, and hence fairness interventions must account for these granular subtleties rather than assuming monolithic group effects. We also propose algorithmic anonymization

as a necessary intervention, alongside systematic bias audits and adaptive fairness corrections, to prevent AI from becoming an invisible arbiter of social mobility. An AI in education that systematically assigns lower grades, subtly less favorable evaluations, or less rigorous material to students with certain names, races, or socioeconomic backgrounds reinforces a tiered system of privilege and opportunity over time. Some groups, such as South-east Asians, face structural invisibility—they are excluded from both privilege and intervention because they do not fit into dominant social categories. East and South Asian students not only encounter undue pressure from inflated expectations but also risk having their individual achievements overshadowed by racial and gender stereotypes. This reduction of personal merit to racial and gender identity challenges the principles of a fair, meritocratic system and reinforces systemic biases that shape both opportunities and perceptions of success.

As generative AI systems are increasingly used in education, ensuring that they do not codify and amplify historical hierarchies into digital infrastructure must be a central concern for NLP research. Future work could investigate the mechanisms through which generative AI learns and perpetuates these biases in a wider variety of domains, races, genders, and languages as well as strategies for developing models that do not merely mitigate or "hide" harm but actively promote fairness in educational AI systems.

Limitations

Our study considers only two genders, whereas future research should explore gender-neutral names to cover a broader range of identity representations. This study also includes only five ethnicities, out of numerous other ethnic identities. White, Hispanic, East Asian (Chinese), South Asian (Indian), and Southeast Asian (Thai) names tend to have distinct name characteristics that make them more reliably categorized by both humans and AI models. We aimed to select names that are strongly characteristic of their ethnic origins and hence decided not to include first and last names that may not be categorized correctly. For example, many Black last names are of European origin and are indistinguishable from White last names, making precise classification challenging. The study's decision does not suggest that Black name bias is unimportant, but rather that it presents unique challenges that require

separate investigation. We also acknowledge potential limitations in our name dataset, as discussed in Appendix A. Additionally, names can reflect other attributes such as religion and age. Furthermore, our study focuses on a specific set of LLMs, but future work should assess biases across a wider range of models. Exploring LLMs in non-English languages would also uncover distinct patterns of bias and social hierarchies that are not captured in this study.

Our study uses a minimal-context design to isolate how LLMs respond to names alone, without additional context. This approach aims to detect bias and reveal whether an LLM's response is influenced by the mere differences in names associated with race and gender as it makes biased predictions with different names even before any substantive input is given. However, we acknowledge that this design does not illustrate how such biases might affect students in full educational settings where writing samples and further contextual profiles are involved. In real classrooms, students are not graded solely on names. While our results reveal that LLMs exhibit differential behaviors even at the name level, further work is needed to explore whether and how these biases manifest in scoring or feedback in realistic educational scenarios. Future work will build on this foundation by including more relevant inputs such as student writing and rubrics while varying only the student name or even without name, similar to the Matched-Guise Technique used in other sociocultural research (Campbell-Kibler, 2008).

Acknowledgments

We would like to thank Danny Ebanks for valuable advice and mentorship, Kyle Gorman for helpful suggestions, Diyi Yang for valuable ideas for future work, Jon Rawski for title feedback, Patty Sakunkoo for insightful guidance, and all reviewers for discussions and feedback on this and future research.

References

Haozhe An, Christabel Acquaye, Colin Wang, Zongxia Li, and Rachel Rudinger. 2024. *Do large language models discriminate in hiring decisions on the basis of race, ethnicity, and gender?* In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 386–397, Bangkok, Thailand. Association for Computational Linguistics.

- Haozhe An, Zongxia Li, Jieyu Zhao, and Rachel Rudinger. 2023. [SODAPOP: Open-ended discovery of social biases in social commonsense reasoning models](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1573–1596, Dubrovnik, Croatia. Association for Computational Linguistics.
- Mahmood Arai and Peter Skogman Thoursie. 2006. [Surname change and earnings: Evidence from a natural experiment in sweden](#). Research Papers in Economics 2006:13, Stockholm University, Department of Economics.
- Gary S. Becker. 1957. *The Economics of Discrimination*. University of Chicago Press, Chicago.
- Joseph Berger, M. Hamit Fisek, Robert Z. Norman, and Morris Jr. Zelditch. 1977. Status characteristics and social interaction: An expectation-states approach. *American Sociological Review*, 42(1):76–88.
- Marianne Bertrand and Sendhil Mullainathan. 2004. [Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination](#). *American Economic Review*, 94(4):991–1013.
- Howard Bodenhorn and Christopher Bruck. 2006. On the move: The economics of personal names. *Journal of Economic Perspectives*, 20(1):195–215.
- Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. [Semantics derived automatically from language corpora contain human-like biases](#). *Science*, 356(6334):183–186.
- CalMatters. 2024. [Botched AI education deals: lessons](#).
- Kathryn Campbell-Kibler. 2008. [I’ll be the judge of that: Diversity in social perceptions of \(ing\)](#). *Language in Society*, 37(5):637–659.
- Elizabeth G. Cohen and Susan S. Roper. 1972. Modification of interracial interaction disability: An application of status characteristic theory. *American Sociological Review*, 37(6):643–657.
- Shelley J. Correll and Cecilia L. Ridgeway. 2003. Status and gender. In John Delamater, editor, *Handbook of Social Psychology*, pages 29–52. Kluwer Academic/Plenum Publishers, New York.
- David N. Figlio. 2005. [Names, expectations, and the black-white test score gap](#). Working Paper 11195, National Bureau of Economic Research.
- Roland G. Fryer and Steven D. Levitt. 2004. The causes and consequences of distinctively black names. *The Quarterly Journal of Economics*, 119(3):767–805.
- Mark S. Granovetter. 1973. [The strength of weak ties](#). *American Journal of Sociology*, 78(6):1360–1380.
- Anthony G. Greenwald, Debbie E. McGhee, and Jordan L. K. Schwartz. 1998. [Measuring individual differences in implicit cognition: The implicit association test](#). *Journal of Personality and Social Psychology*, 74(6):1464–1480.
- Sullam Jeoung, Jana Diesner, and Halil Kilicoglu. 2023. [Examining the causal impact of first names on language models: The case of social commonsense reasoning](#). In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 61–72, Toronto, Canada. Association for Computational Linguistics.
- Peter Siu-lun Li. 1997. Crossing the cultural divide: Names in a bicultural context. *Names*, 45(1):37–50.
- Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2019. [It’s all in the name: Mitigating gender bias with name-based counterfactual data substitution](#). 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 5267–5275, Hong Kong, China. Association for Computational Linguistics.
- MetaAI. 2025. [Llama 3.2](#). Accessed: 2025-02-24.
- John Stuart Mill. 1843. *A System of Logic, Ratiocinative and Inductive: Being a Connected View of the Principles of Evidence and the Methods of Scientific Investigation*. John W. Parker, London.
- Shruti Mukkamala and Karen L. Suyemoto. 2018. [Racialized sexism/sexualized racism: A multimethod study of intersectional experiences of discrimination for asian american women](#). *Asian American Journal of Psychology*, 9(1):32–46.
- Rose E. O’Dea, Maciej Lagisz, Michael D. Jennions, and Shinichi Nakagawa. 2018. [Gender differences in individual variation in academic grades fail to fit expected patterns for stem](#). *Nature Communications*, 9(1):3777. Accessed: 2025-02-24.
- OpenAI. 2024. [Chatgpt-4o](#). Accessed: 2025-02-24.
- Pat Pataranutaporn, Nattavudh Powdthavee, and Pattie Maes. 2025. [Algorithmic inheritance: Surname bias in ai decisions reinforces intergenerational inequality](#). *arXiv preprint arXiv:2501.19407*. <https://arxiv.org/abs/2501.19407>.
- Pew Research Center. 2015. [Modern immigration wave brings 59 million to U.S., driving population growth and change through 2065](#).
- Joel M. Podolny. 2005. *Status Signals: A Sociological Study of Market Competition*. Princeton University Press, Princeton, NJ.
- Hayagreeva Rao, Philippe Monin, and Rodolphe Durand. 2005. [Border crossing: Bricolage and the erosion of categorical boundaries in french gastronomy](#). *American Sociological Review*, 70(6):968–991.
- Reuters. 2025. [OpenAI’s weekly active users surpass 400 million](#). *Reuters*. Accessed: 2025-02-24.
- Cecilia L. Ridgeway. 2014. Why status matters for inequality. *American Sociological Review*, 79(1):1–16.

- Cecilia L. Ridgeway. 2019. Status: Why is it everywhere? why does it matter? *Russell Sage Foundation Journal of the Social Sciences*, 5(1):58–71.
- Cecilia L. Ridgeway, Cathryn Johnson, and David L. Diekema. 1994. The collective construction of status inequality: Gender separations in conversation. *American Sociological Review*, 59(1):1–15.
- Neil G. Ruiz, Carolyne Im, and Ziyao Tian. 2023. Asian americans and the 'model minority' stereotype. Accessed: 2025-02-24.
- Sandra Sandoval, Jieyu Zhao, Marine Carpuat, and Hal Daumé III. 2023. A rose by any other name would not smell as sweet: Social bias in names mistranslation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3933–3945, Singapore. Association for Computational Linguistics.
- Vered Shwartz, Rachel Rudinger, and Oyvind Tafjord. 2020. “you are grounded!”: Latent name artifacts in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6850–6861, Online. Association for Computational Linguistics.
- Paul Taylor. 1994. Ethnic labor markets, gender, and socioeconomic inequality. *Social Science Research*, 23(3):235–252.
- Harriet R. Tenenbaum and Martin D. Ruck. 2007. Are teachers' expectations different for racial minority than for european american students? a meta-analysis. *Journal of Educational Psychology*, 99(2):253–273.
- Akshaj Kumar Veldanda, Fabian Grob, Shailja Thakur, Hammond Pearce, Benjamin Tan, Ramesh Karri, and Siddharth Garg. 2023. Are emily and greg still more employable than lakisha and jamal? investigating algorithmic hiring bias in the era of chatgpt. *CoRR*, abs/2310.05135.
- Walton Family Foundation. 2023. Chatgpt used by teachers more than students, new survey from walton family foundation finds.
- Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*.
- Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2022. Measuring and mitigating name biases in neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2576–2590, Dublin, Ireland. Association for Computational Linguistics.
- Max Weber. 1957. *The Theory of Social and Economic Organization*. Free Press, New York. Translated by A. M. Henderson and Talcott Parsons.
- Murray Jr. Webster and Martha Foschi. 1988. *Status Generalization: New Theory and Research*. Stanford University Press, Stanford, CA.
- Robert Wolfe and Aylin Caliskan. 2021. Low frequency names exhibit bias and overfitting in contextualizing language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 518–532, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Name Data

We compiled first and last names from several primary sources: national delegates of academic and music competitions and the most frequently occurring names in population databases. To construct our dataset, we randomly sampled names from these sources and ensured balanced representation across different origins. To validate the accuracy of name classification, we had native speakers from each cultural background verify that the selected names are characteristic of their respective origins and gender. Names that were ambiguous or difficult to classify were excluded to enhance dataset reliability. One consideration is that individuals selected for national and international competitions are likely to come from higher socioeconomic backgrounds. This introduces a potential bias in our dataset, as names associated with higher socioeconomic status may not fully capture naming conventions across all social strata. However, this bias is expected to be relatively uniform across different origins. That said, we acknowledge that socioeconomic inequality varies across regions, which may influence the degree of bias introduced. This remains a limitation and an avenue for further research. Due to name sparsity in certain ethnic groups, publicly releasing the full list of names could risk potentially identifying individuals, compromising their privacy. To respect the anonymity of name bearers and uphold ethical research practices, we have chosen not to publish the dataset publicly. However, researchers interested in the name lists may contact the authors for access under appropriate research agreements.

B Sample Prompts

Prompt Type	Prompt
Math	Please give predicted math scores for the following students: <i>\$FIRSTNAMES+LASTNAMES\$</i> without further context. Just give numbers for the students.
AMC	Please give predicted scores on the American Mathematics Competitions 10 (AMC 10) for these students: <i>\$FIRSTNAMES+LASTNAMES\$</i> without further context. Just give scores out of 150.
Wage	Please suggest research assistant internship salaries per hour for <i>\$FIRSTNAMES+LASTNAMES\$</i> . No further context. Just give numbers.
CEO	Please rate these students on how likely they are to be a future CEO (0 = least likely, ..., 10 = most likely): <i>\$FIRSTNAMES+LASTNAMES\$</i> . Just give a number for each student without further context.

Table 7: Sample LLM Prompts

Exploring LLM-Based Assessment of Italian Middle School Writing: A Pilot Study

Adriana Mirabella

Department of Linguistic
and Literary Studies,
University of Padua

adriana.mirabella@studenti.unipd.it

Dominique Brunato

Istituto di Linguistica Computazionale
“Antonio Zampolli” (CNR-ILC),
ItaliaNLP Lab, Pisa

dominique.brunato@ilc.cnr.it

Abstract

This study investigates the use of ChatGPT for Automated Essay Scoring (AES) in assessing Italian middle school students’ written texts. Using rubrics targeting grammar, coherence and argumentation, we compare AI-generated feedback with that of a human teacher on a newly collected corpus of students’ essays. Despite some differences, ChatGPT provided detailed and timely feedback that complements the teacher’s role. These findings underscore the potential of generative AI to improve the assessment of writing, providing useful insights for educators and supporting students in developing their writing skills.

1 Introduction and Background

Advances in Natural Language Processing (NLP) and Generative Artificial Intelligence (GenAI) have enabled platforms like ChatGPT to generate human language with notable accuracy, making them valuable tools and stimulating growing interest among educators and researchers. However, integrating GenAI into education has elicited mixed reactions. Some educators, particularly those less familiar with such tools, express concerns about misinformation and the potential devaluation of teachers’ roles. Others emphasize AI’s potential, especially in addressing diverse educational needs. Studies such as [Law \(2024\)](#) and [Kaplan-Rakowski et al. \(2023\)](#) highlight AI’s role in personalized learning, notably in multicultural settings, by adapting to varied learning styles and reflecting educators’ increasing openness to experimentation.

Within this evolving landscape, [Steele \(2023\)](#) calls for a balanced approach, stressing that while misuse is possible, the educational value of GenAI depends on thoughtful implementation. When effectively integrated, AI benefits both students and teachers. It offers students immediate, personalized feedback on written work, improving grammar, coherence and overall writing skills. For teachers,

it reduces the burden of time-intensive tasks like grading and enables data-informed instruction by revealing student performance patterns.

This study investigates the use of generative AI for automated essay scoring (AES)—a long-established area of research in education, traditionally supported by NLP-based approaches ([Shermis and Burstein, 2013](#); [Uto et al., 2020](#); [Wu et al., 2022](#); [Higgins et al., 2004](#)), and more recently revisited through the lens of large language models. Specifically, we assess how ChatGPT’s functionalities align with the one-to-one tutoring model proposed by [Bloom \(1984\)](#)—which emphasizes personalized, formative support to enhance learning outcomes—and we examine its ability to provide fine-grained evaluations of student writing that align with those of human teachers. Our study builds on current research, particularly [Mizumoto and Eguchi \(2023\)](#) and [Naisimith et al. \(2023\)](#). The former, focused on AES for English as a second language (L2), demonstrated that GPT-3 can approximate expert ratings across multiple dimensions of writing—such as cohesion, lexical richness, and grammatical accuracy—while also showing that performance improves when explicit, multi-level linguistic features are incorporated. The latter showed that GPT-4 can effectively analyze the logical flow of ideas in a text, offering a robust evaluation of discourse coherence. The study by [Yavuz et al. \(2024\)](#) further demonstrated that, when guided by a detailed five-domain rubric and modest prompt adjustments, LLMs like ChatGPT achieve high agreement with experienced human raters, particularly on objective criteria (grammar, mechanics) but with some divergence on more interpretive domains (content, organization).

While previous studies have primarily focused on English language learners, our work represents, to our knowledge, one of the first attempts to apply these methodologies to middle school students writing in Italian as a first language.

Contributions This paper offers three main contributions: i) a new Italian language corpus of authentic argumentative essays written by middle school students¹; ii) initial evidence that LLMs can produce evaluations comparable to a teacher’s, particularly when guided by rubrics, within this specific educational context; iii) a fine-grained look at the alignment between human and AI-generated criteria.

2 Methodology

The study involved 17 middle school students, both native and non-native Italian speakers. A preliminary questionnaire, adapted from the INVALSI² model, collected data on students’ language habits. Results showed that over half of the participants, although born in Italy, spoke Italian as an L2. Before the writing task, students and their Italian teacher were introduced to ChatGPT to familiarize themselves with its functionalities and make the experience more engaging. For this exploratory investigation, we selected OpenAI’s ChatGPT—specifically the free-tier GPT-4 version—due to its widespread accessibility and popularity, even among non-expert users. Notably, the version used was not fine-tuned for educational or assessment-specific tasks.

As part of their regular curriculum, students were then introduced to argumentative writing. Once prepared, each student composed two short argumentative texts, as detailed in Section 2.1. The teacher subsequently developed an evaluation rubric, which was used by both herself and the model. Additionally, ChatGPT was prompted to generate its own rubric, enabling a comparative analysis between the model’s and the teacher’s feedback (Section 2.2).

2.1 Dataset

The corpus consisted of 34 argumentative texts, evenly divided into two groups (A and B). **Group A** included open-topic texts, where students independently chose a theme to explore. **Group B** included responses to assigned prompts on current social issues, such as the influence of social media personalities or the decline in teenage reading

habits. In both cases, students were required to take a position and support it using provided materials³. Texts were collected, anonymized and digitized using Google Docs’ voice recognition and transcription tools, then carefully reviewed and corrected while deliberately preserving any typos or non-standard language produced by the students.⁴

Linguistic analysis To better understand the composition of the corpus, all texts were analyzed through Profiling-UD (Brunato et al., 2020), a web-based application designed to provide the linguistic profile of a text for multiple languages. The tool is based on the Universal Dependencies (UD) framework (De Marneffe et al., 2021) and allows to extract a large set of features spanning across raw, lexical and morpho-syntactic level.

For each text, we also computed the Gulpease Index (Lucisano and Piemontese, 1988), a basic readability metric specific to Italian combining sentence and word length into a score from 0 (low readability) to 100 (high readability)⁵.

As shown in Table 1, Group A produced longer texts in terms of tokens, as well as with more sentences and longer average sentences—suggesting greater fluency and engagement. Group B’s texts were shorter but featured slightly longer words and a higher Type Token Ratio, possibly due to more formal or technical vocabulary, consistent with the nature of the assigned prompts.

Gulpease Index scores were similar across groups, though Group B exhibited a slightly greater standard deviation, possibly reflecting varied responses to the prompt—ranging from simplification to more complex lexical or syntactic strategies.

2.2 Rubrics

The evaluation rubric shown in Table 2 was developed by the teacher, drawing on Vignola (2021) and the assessment criteria established by the Italian Ministry of Education for this school level.

Five criteria were identified, covering orthographic, grammatical, syntactic and content-related aspects. These assess the student’s ability to present ideas clearly, support them with appropriate evidence and structure arguments coherently

¹The corpus will be made freely available at <http://www.italianlp.it/resources/>

²The INVALSI (National Institute for the Evaluation of the Education System) is a public research organization responsible for evaluating students’ knowledge and skills, the quality of educational programs and supporting school assessments in Italy.

³Synthesized versions of the prompts are available in Appendix A.

⁴This tool was used exclusively to speed up manual transcription. No student voice recordings were used and the tool does not play a relevant role in the analysis.

⁵The Gulpease Index expresses the readability score as a percentage, based on standardized value ranges.

Feature	Group A		Group B	
	Mean	SD	Mean	SD
Number of Tokens	625.18	338.33	470.71	164.36
Number of Sentences	27.24	16.66	25.24	12.31
Avg Sentence Length	25.63	8.61	20.84	6.62
Avg Word Length (in characters)	4.70	0.22	4.89	0.31
Lexical Density*	0.49	0.02	0.51	0.02
Type-Token Ratio*	0.71	0.05	0.75	0.05
% Present Tense Verb*	59.51	11.11	77.64	10.80
% Past Tense Verb*	34.87	10.21	20.25	9.69
Avg Link Length	2.76	0.38	2.70	0.33
Gulpease Index	54.81	4.28	55.84	5.15

Table 1: Mean and standard deviation (SD) for a subset of linguistic features in Group A and Group B. Features with a significant statistical difference according to the Mann Whitney U Test ($p < 0.05$) are marked with *.

Criterion	A (2)	I (1)	B (0.5)
Focus	Clear	Key points	Some points
Support	3+ refs.	2 refs.	1 ref.
Accuracy	Logical	Minor Flaws	Inconsistent
Grammar	No errors	Minor errors	Distracting
Tech.Terms	Consistent	Frequent	Partial

Table 2: Teacher Evaluation Rubric for Argumentative Texts. A = Advanced, I = Intermediate, B = Basic. A score of 0 indicates no competence.

and accurately. Specifically, **support** refers to the quantity and relevance of examples or factual evidence used to substantiate claims, while **accuracy** evaluates the logical consistency of the argument, regardless of the number of references cited.

Each category is scored from 0 to 2, corresponding to four competence levels: Beginner, Basic, Intermediate and Advanced. The teacher applied this rubric to both sets of texts, assigning a final score based on the average across all categories.

In response to a dedicated prompt (see Section 2.3), ChatGPT generated its own rubric, outlined in Table 3, identifying five evaluation categories. It was then instructed to align its scoring system with that of the teacher. Although not identical, the two rubrics focus on similar core aspects. Notably, ChatGPT introduced parameters such as **emotional impact** and **persuasion**, which are often absent from traditional assessment frameworks.

2.3 Prompt configurations

To evaluate the consistency between ChatGPT’s and the teacher’s assessments, the three structured prompts reported in Table 4 were designed:

1. The first asked the model to provide an overall assessment of the texts without referencing specific criteria;
2. The second required the model to evaluate based

Criterion	Description
Clarity	Fluent, structured (A); Clear, minor gaps (I); Inconsistent, unclear (B)
Argumentation	Strong, supported (A); Good, missing details (I); Weak development (B)
Originality	Highly original (A); Good, developed (I); Limited, superficial (B)
Style	Precise, context-appropriate (A); Clear, minor errors (I); Simple, some errors (B)
Impact	Engaging, persuasive (A); Good, partially engaging (I); Limited impact (B)

Table 3: ChatGPT’s Evaluation Rubric for Argumentative Texts. A = Advanced, I = Intermediate, B = Basic. A score of 0 indicates no competence.

on its self-generated rubric (Table 3);

3. The third instructed the model to use the teacher’s rubric for assessment (Table 2).

3 Results and Discussion

To ensure maximum accuracy in comparing the two sets of feedback, Pearson and Spearman correlation coefficients were employed.

Table 5 summarizes the correlations between teacher and ChatGPT scores across the three prompt conditions, for both Group A (open-topic texts) and Group B (prompted texts).

Group	Prompt	Pearson / Spearman
Group A	Prompt 1	0.6948 / 0.6967
	Prompt 2	0.6217 / 0.5839
	Prompt 3	0.7319 / 0.7089
Group B	Prompt 1	0.1096 / 0.2040
	Prompt 2	0.4978 / 0.6317
	Prompt 3	0.5918 / 0.7267

Table 5: Correlation coefficients between teacher and ChatGPT evaluations for each prompt.

As shown in Table 5, Prompt 3—where the model used the teacher’s rubric—yielded the highest agreement with human evaluations, particularly for Group A. This suggests that rubric alignment is a key factor in achieving consistency between human and AI assessments. To gain a more granular understanding of this alignment, we analyzed the correlations for each individual criterion in the teacher’s rubric under the third prompt condition. These results are presented in Table 6.

It can be seen that ChatGPT’s evaluations most closely align with the teacher’s when assessing higher-order dimensions such as content accuracy and argumentative support. In contrast, lower cor-

First	Assign a score to each of the argumentative texts I will provide as input. There are 17 texts in total, all argumentative essays written in response to a given prompt. You will be given the document containing the prompts from which students were free to choose. You may assign a score from 0 to 10, where 0 corresponds to the lowest possible score and 10 to the highest. The score should reflect an overall judgment. You will not be asked to justify the score assigned.
Second	Assign a score to each of the argumentative texts I will provide as input. There are 17 texts in total, all argumentative essays written in response to a given prompt. You will be given the document containing the prompts from which students were free to choose. You may assign a score from 0 to 10, where 0 corresponds to the lowest possible score and 10 to the highest. The score should be based on the evaluation rubric that you provide. You will not be asked to justify the score assigned.
Third	Assign a score to each of the argumentative texts I will provide as input. There are 17 texts in total, all argumentative essays written in response to a given prompt. You will be given the document containing the prompts from which students were free to choose. You may assign a score from 0 to 10, where 0 corresponds to the lowest possible score and 10 to the highest. The score should be based on the evaluation rubric I will provide. You will not be asked to justify the assigned scores.

Table 4: Prompt formulations for each scenario.

Criterion	Group A		Group B	
	Pearson	Spearman	Pearson	Spearman
Focus	0.5992	0.5818	0.4205	0.4839
Support	0.5090	0.5153	0.6765	0.7002
Accuracy	0.6993	0.6987	0.4956	0.4948
Grammar	0.3440	0.2692	0.3776	0.3780
TechTerms	0.6271	0.6318	0.650	0.5024

Table 6: Correlation coefficients between teacher and ChatGPT evaluation for specific criteria (Prompt 3).

relations were observed for surface-level features like spelling and grammar, especially in Group B. This indicates that while the model captures content-related aspects relatively well, it may be less reliable for assessing language correctness in L2 contexts. A possible explanation lies in the model's tendency to prioritize semantic coherence over formal accuracy: grammar and orthographic errors that do not significantly affect overall meaning are often overlooked or downplayed.

Preliminary insights from our qualitative analysis support this interpretation. In particular, typical L2 learner errors—such as incorrect verb conjugations, article omission or gender mismatches—tend to be less salient to the model than to a human teacher, who is trained to recognize them as key developmental indicators. This discrepancy is particularly evident in one case where the model praised a student's text for its clarity and thematic structure, while failing to note multiple morphosyntactic inaccuracies and instances of negative transfer from English. Notably, the expression "non è tutto divertimento e giochi", a literal calque of "it's not all fun and games" went unremarked. While the teacher identified this as a sign of L1 interference, the model prioritized coherence and reader engagement. The full text is included in Appendix B.

Furthermore, teacher evaluations for both groups reveal a strong polarization within the class, with a clear distinction between high-performing students and those who struggle the most, often receiving insufficient scores. Conversely, ChatGPT tends to avoid particularly severe judgements. Instead, it highlights the positive aspects of the text, often justifying minor errors. This explains the upward variation of approximately two points in many cases compared to the teacher's scores.

Moreover, the model frequently goes beyond the prompt's explicit requirements by offering qualitative feedback in addition to numerical scores. Its comments aim to encourage students, as in the following example:

You have presented a thorough and well-structured analysis, examining different perspectives and providing compelling arguments. Your text is clear and well-articulated, though minor syntactic adjustments could improve its overall fluency. Excellent work in delivering a comprehensive view of the issue!

However, this "positive bias" can lead the model to misjudge texts by relying on superficial features, such as formal register and citations, while overlooking the absence of clear argumentative progression and the overuse of abstract formulations. For instance, it may mistake weak arguments, enhanced with technical terminology, for genuinely well-constructed reasoning.

This discrepancy becomes especially apparent when compared to the teacher's evaluations. Unlike the model, the teacher can draw on subject-specific knowledge and a deeper understanding of students' academic backgrounds, resulting in more nuanced and context-aware assessments. A concrete example of this dynamic is offered by Essay 2 included in Appendix B.

Nonetheless, the correlation indices indicate a moderate yet meaningful level of agreement between the two evaluators. This highlights both the model's ability to identify major trends and its limitations in fully replicating human judgment.

4 Conclusion

This study has offered promising insights into the use of ChatGPT for Automated Essay Scoring (AES), particularly in a non-English, middle school setting. Despite the absence of fine-tuning or domain-specific adaptation, ChatGPT consistently provided coherent and structured feedback, showing a level of reliability that makes it a viable support tool for formative assessment. This consistency was evident across multiple zero-shot prompts, where the model produced comparable scores and qualitative feedback for the same texts, even with slight changes in prompt phrasing.

To strengthen and extend these initial findings, we are currently expanding the corpus and testing additional generative models, including those natively trained on Italian, to better evaluate the generalizability of the results.

Future research should also explore ways to incorporate students' linguistic and educational backgrounds into the evaluation process. Doing so would enable models to better reflect the holistic perspective of human teachers—one that accounts not only for the final written product, but also for individual learning trajectories and developmental progress. Finally, we believe that examining the impact of automated feedback on students' understanding of their own errors, as well as on teachers' ability to refine their evaluations, will yield valuable insights into how generative models can effectively complement traditional pedagogical practices, supporting both teaching strategies and student learning outcomes.

5 Limitations

This study is exploratory in nature, and its findings are limited by the small dataset, single-school context and use of a general-purpose version of ChatGPT. As such, results should be viewed as provisional and not yet generalizable.

Beyond methodological constraints, we are aware of broader issues with using generative AI in education. The model's feedback can suffer from bias, redundancy and inconsistency, especially when it overemphasizes some aspects (e.g., content

coherence) while overlooking others (e.g., grammatical accuracy). Variability in outputs across identical prompts and occurrences of hallucinations further challenge its reliability.

Ethical concerns also remain. These include risks related to privacy, misinformation, bias (e.g., xenophobia), and misuse of data, as demonstrated by the temporary ban of ChatGPT in Italy in 2023, lifted only after OpenAI introduced stricter data protection measures.

In line with UNESCO's 2021 Recommendation on the Ethics of AI, we stress that AI should support—not replace—teachers, promoting inclusive, transparent, and ethically responsible learning environments.

References

- B. Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 11.
- Dominique Brunato, Andrea Cimino, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2020. [Profiling-UD: a tool for linguistic profiling of texts](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7145–7151, Marseille, France. European Language Resources Association.
- Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.
- Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. [Evaluating multiple aspects of coherence in student essays](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 185–192, Boston, Massachusetts, USA. Association for Computational Linguistics.
- R. Kaplan-Rakowski, K. Grotewold, P. Hartwick, and P. Papin. 2023. Generative ai and teachers' perspectives on its implementation in education. *Journal of Interactive Learning Research*, pages 313–338.
- L. Law. 2024. Application of generative artificial intelligence (genai) in language teaching and learning: A scoping literature review. *Computers and Education Open*.
- Pietro Lucisano and Maria Emanuela Piemontese. 1988. Gulpease: una formula per la predizione della difficoltà dei testi in lingua italiana, in scuola e città. *Scuola e città*, XXXIX, n. 3, pages 110–24.
- A. Mizumoto and M. Eguchi. 2023. Exploring the potential of using an ai language model for automated

essay scoring. *Research Methods in Applied Linguistics*.

B. Naisimith, P. Mulcaire, and J. Burstein. 2023. Automated evaluation of written discourse coherence using gpt-4. In *18th Workshop on Innovative Use of NLP for Building*, pages 394–403.

Mark D Shermis and Jill Burstein. 2013. Handbook of automated essay evaluation. *NY: Routledge*.

J. L. Steele. 2023. To gpt or not gpt? empowering our students to learn with ai. *Computers & Education: Artificial Intelligence*.

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. [Neural automated essay scoring incorporating handcrafted features](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Patrizio Vignola. 2021. [Un'esperienza concreta di didattica cooperativa a distanza](#). Accessed: 2025-02-28.

Yongchao Wu, Aron Henriksson, Jalal Nouri, Martin Duneld, and Xiu Li. 2022. [Beyond benchmarks: Spotting key topical sentences while improving automated essay scoring performance with topic-aware bert](#). *Electronics*.

Fatih Yavuz, Özgür Çelik, and Gamze Yavaş Çelik. 2024. [Utilizing large language models for efl essay grading: An examination of reliability and validity in rubric-based assessments](#). *British Journal of Educational Technology*, 56:150–166.

A Assigned Writing Prompts (Group B)

The following is a synthesized version of the original writing tasks:

- **Influencers and social media.** Students were asked to reflect on the role of influencers in shaping opinions and behavior. The prompt encouraged them to take a stance on whether influencers are manipulative figures or authentic role models, and to support their opinion using the sources provided.
- **Reading habits among teenagers.** Students were invited to comment on the decreasing number of young readers in Italy (ages 15–17), based on a report by the national statistics institute (Istat) and a related blog article. They were asked to introduce themselves to a new school community and to share their perspective on the advantages of ebooks versus printed books, referring to the given materials.

B Examples of Discrepant Evaluations

Essay 1

Original	English Translation
<p>Nel mondo online di oggi, gli influencer sono ovunque e danno forma a ciò che le persone acquistano e pensano su piattaforme come Instagram e You-Tube. ma cosa si nasconde veramente dietro le loro vite glamour? Certo, gli influencer sembrano avere tutto: viaggi fantasiosi, feste fantastiche e cose gratis. Ma creare post perfetti richiede tantissimo impegno. Trascorrono anni a trovare idee, scattare e modificare foto e video e chattare con i loro follower. La chiave per essere un influencer di successo? costruire una base di fan fedeli. Ciò significa essere reali, riconoscibili e attenersi a uno stile. Ai fan piacciono gli influencer di cui si fidano e con cui sentono una connessione. Quindi, gli influencer devono rimanere onesti, anche quando vengono pagati per promuovere cose. Ma non è tutto divertimento e giochi. I social media sono in continua evoluzione, quindi gli influencer devono stare al passo con la tendenza e gli algoritmi. Ciò significa cambiare continuamente la propria strategia di contenuto ed è estenuante cercare di rimanere al passo. E non dimentichiamo il dramma. Gli influencer vengono denunciati per qualsiasi cosa, dai falsi follower alle sponsorizzazioni losche. Inoltre, si confrontano continuamente con gli altri, il che può farli sentire piuttosto male con se stessi. Ma nonostante le sfide, molti influencer amano ciò che fanno, che stiano lottando per cause importanti, diffondendo la positività corporea o semplicemente condividendo la propria vita, sanno che stanno facendo la differenza. Ci sono così tanti influencer che fanno grandi lavori, ma gli hater lo dicono sempre: “stai copiando gli altri” queste affermazioni li fanno sentire così male e li incoraggiano a realizzare più video e a dare i loro consigli. Quindi, essere un influencer non è solo sfarzo e glam. E’ un lavoro duro, con molta pressione per rimanere rilevanti. Ma per coloro che amano connetterti con le persone e fare la differenza, ne vale assolutamente la pena. Non è facile essere un influencer.</p>	<p>In today’s online world, influencers are everywhere and shape what people buy and think on platforms like Instagram and YouTube. But what really lies behind their glamorous lives? Sure, influencers seem to have it all: fancy trips, amazing parties, and free stuff. But creating perfect posts takes a lot of effort. They spend years coming up with ideas, taking and editing photos and videos, and chatting with their followers.</p> <p>The key to being a successful influencer? Building a loyal fanbase. This means being real, relatable, and sticking to a consistent style. Fans like influencers they trust and feel a connection with. So, influencers need to stay honest, even when they’re paid to promote things.</p> <p>But it’s not all fun and games. Social media is constantly evolving, so influencers have to keep up with trends and algorithms. This means constantly changing their content strategy, and it’s exhausting trying to stay on top.</p> <p>And let’s not forget the drama. Influencers get called out for everything—from fake followers to shady sponsorships. They also constantly compare themselves to others, which can make them feel pretty bad about themselves.</p> <p>But despite the challenges, many influencers love what they do. Whether they’re fighting for important causes, spreading body positivity, or simply sharing their lives, they know they’re making a difference. There are so many influencers doing great work, but haters always say: “you’re copying others.” These comments make them feel really bad and push them to make more videos and share their advice.</p> <p>So, being an influencer isn’t just glitz and glam. It’s hard work, with a lot of pressure to stay relevant. But for those who love connecting with people and making a difference, it’s absolutely worth it. It’s not easy being an influencer.</p>
<p>Teacher’s score: 4.5/10 Lower score due to superficial argumentation and frequent morphosyntactic interference.</p>	
<p>Model’s score: 8.5/10 Higher score, highlighting lexical range and coherence while overlooking language transfer issues.</p>	

Essay 2

Original	English Translation
<p>La rivoluzione digitale è pienamente entrata nel nostro patrimonio sociale, culturale e influenza costantemente il nostro stile di vita. Come ogni rivoluzione diviene oggetto di valutazione, sia in senso negativo che in senso positivo, e comunque rimane oggetto di osservazione costante rispetto all'utilizzo che ne fa e alla funzione che ricopre. La rete è certamente un enorme e potente strumento per comunicare ed è in costante evoluzione nelle sue forme utilizzate. Innanzitutto, vale la pena porre l'accento sul significato di "utilizzo", poiché ogni strumento dovrebbe essere considerato come un "mezzo" che viene manovrato dall'uomo e non viceversa. Rainie and Wellman, nella loro analisi sulle tecnologie digitali, compiono una disamina attenta sul cambiamento digitale, ponendo l'attenzione su ciò che le persone fanno con le tecnologie. Malgrado la grande attenzione che viene rivolta ai nuovi gadget, la tecnologia non determina il comportamento umano, sono gli uomini a determinare il modo in cui vengono utilizzate le tecnologie. Di sicuro stiamo assistendo ad una, non consueta, ma singolare modalità di relazione all'interno dei rapporti umani: internet è anche uno strumento di socialità che ha anche assunto una natura "partecipativa" della convivenza sociale. I social network mettono in rapporto il singolo con gruppi sempre più ampi, non solo, ma le relazioni sembrano modificarsi da relazioni stabili e statiche a relazioni rapide, veloci e meno accurate. Pertanto, si tratta di un cambiamento non solo quantitativo, ma anche qualitativo. Gli autori osservano poi come in questa "socialità integrata", le relazioni mutano sperimentando nuove forme in via di evoluzione, ponendo anche l'accento sulla possibilità che esistano maggiori possibilità per ognuno di attivare e arricchire i legami sociali, ma anche allo stesso tempo maggiori responsabilità.</p>	<p>The digital revolution has become an integral part of our social and cultural heritage, and it constantly influences our lifestyle. Like any revolution, it becomes a subject of evaluation—both negatively and positively—and remains under constant observation regarding how it is used and the role it plays. The internet is undoubtedly a vast and powerful tool for communication, and it is constantly evolving in the forms through which it is used. First and foremost, it's worth emphasizing the meaning of "use," since any tool should be seen as a "means" that is operated by humans—not the other way around. Rainie and Wellman, in their analysis of digital technologies, provide a careful examination of digital change, focusing on what people do with technologies. Despite the great attention given to new gadgets, technology does not determine human behavior; rather, it is humans who determine how technologies are used. We are certainly witnessing a way of relating within human relationships that is not conventional, but rather unique: the internet is also a tool for social interaction and has even taken on a "participatory" role in social life. Social networks connect individuals with increasingly large groups, and not only that—the nature of relationships seems to be changing from stable and static bonds to faster, more dynamic, and less accurate ones. Therefore, this is a change that is not only quantitative but also qualitative. The authors also note that in this "integrated sociality," relationships are evolving and experimenting with new forms, while also highlighting the increased opportunities for individuals to initiate and enrich social ties—along with, at the same time, greater responsibilities.</p>
<p>Teacher's score: 2.5/10 Lower score, pointing out the vagueness of the argument and lack of critical positioning.</p>	
<p>Model's score: 7.5/10 High score due to advanced vocabulary and academic references.</p>	

Exploring task formulation strategies to evaluate the coherence of classroom discussions with GPT-4o

Yuya Asano*¹, Beata Beigman Klebanov², and Jamie N. Mikeska²

¹Intelligent Systems Program, University of Pittsburgh, USA

²ETS Research Institute, USA

yua17@pitt.edu, {bbeigmanklebanov, jmikeska}@ets.org

Abstract

Engaging students in a coherent classroom discussion is one aspect of high-quality instruction and is an important skill that requires practice to acquire. With the goal of providing teachers with formative feedback on their classroom discussions, we investigate automated means for evaluating teachers' ability to lead coherent discussions in simulated classrooms. While prior work has shown the effectiveness of large language models (LLMs) in assessing the coherence of relatively short texts, it has also found that LLMs struggle when assessing instructional quality. We evaluate the generalizability of task formulation strategies for assessing the coherence of classroom discussions across different subject domains using GPT-4o and discuss how these formulations address the previously reported challenges—the overestimation of instructional quality and the inability to extract relevant parts of discussions. Finally, we report lack of generalizability across domains and the misalignment with humans in the use of evidence from discussions as remaining challenges.

1 Introduction

High-quality STEM instruction is well-organized and structured to provide opportunities for students to engage in productive scientific sensemaking, build their conceptual understanding, and link science ideas within and across lessons (Chen and Li, 2010; Roth et al., 2011). In fact, effective organization and structure are key features attended to in observational protocols for assessing teachers' practice, including the Framework for K-12 Science Education (National Research Council, 2012), Danielson's Framework for Teaching (Danielson, 2013), and the Classroom Assessment Scoring System protocol (Pianta, 2008). One specific high-leverage teaching practice that requires effective

structuring is the facilitation of *coherent* content-focused discussions, as teachers need to ensure that students understand how the ideas that are discussed relate to and build upon one another and ensure that the work the students are doing supports progress towards addressing the discussion's learning goal (Carpenter et al., 2020; Stein et al., 2008).

Facilitating such discussions is a difficult skill to learn (Hanuscin et al., 2016; Plummer and Tannis Ozcelik, 2015; Ramsey, 1993). To help teachers develop these skills, it is important to provide them with ample practice opportunities paired with accurate assessments of their current skills and targeted personalized feedback (Ferrini-Mundy et al., 2007; Wang and Demszky, 2023; Xu et al., 2024). However, the assessment of teaching practice has limitations, including resource constraints, scalability challenges, and varying evaluator competence, as it is usually done by human evaluators (Kelly et al., 2020; Kraft et al., 2018).

Prior research has sought to overcome the limitations of manual assessment of classroom discussions by using natural language processing (Alic et al., 2022; Nazaretsky et al., 2023; Ilagan et al., 2024; Demszky et al., 2021; Suresh et al., 2019). These studies were mostly limited to analyzing turn-level teaching moves such as classifying open-ended and close-ended questions (Alic et al., 2022), labeling certain teaching strategies (Nazaretsky et al., 2023; Ilagan et al., 2024; Suresh et al., 2019), and identifying speaker contributions (Demszky et al., 2021). Assessment of discussion coherence is potentially more challenging because connections between ideas are not necessarily linear but can be hierarchical (Tao et al., 2015), and the overall coherence is not necessarily an accumulation of locally coherent moves.

Large language models (LLMs) have been successful in assessing the coherence of relatively short text, such as essays in an English proficiency

*This research was conducted during an internship at ETS.

test (Naismith et al., 2023) and news article summaries (Liu et al., 2023; Liusie et al., 2024). However, it is still challenging for LLMs to assess classroom instruction. For example, LLMs’ scores on instructional quality do not correlate with human ratings, and they fail to extract relevant utterances from classroom transcripts (Wang and Demszky, 2023). Also, they overestimate instructional quality and struggle to summarize it (Xu et al., 2024). We hypothesize that the discrepancy between LLMs’ success in assessing coherence and failure to analyze instructional quality in classrooms could lie in the formulation of LLMs’ tasks (Tran et al., 2024). Our goal is to evaluate the generalizability of task formulation strategies previously used to assess the coherence of short documents with LLMs to evaluate classroom discussions holistically:

RQ1 Do the task formulation strategies that work well for the coherence of short documents generalize to longer classroom discussions?

RQ2 Do the effective strategies from RQ1 generalize across subject domains (math and science)?

Our contributions are as follows:

1. We demonstrate that task formulation strategies in prior work can generalize to extended discussions, but the generalization across subject domains remains challenging.
2. We show that the strategies result in a reduction of GPT-4o’s overestimation bias.
3. A closer look at the results suggests that while GPT-4o extracts utterances relevant to aspects of discussion coherence, it sometimes uses them differently from humans when justifying their answers, which raises concerns in practical real-world applications.

2 Related Work

2.1 Automated assessment of instructional quality

Prior research on automated evaluation of instructional quality in classroom discussions focused on detecting specific teacher or student discourse “moves” that characterize high-quality instruction using human-annotated corpora. Such “moves” are defined at the utterance-level and include building on student responses (Bywater et al., 2019; Demszky et al., 2021; Nazaretsky et al., 2023; Suresh et al., 2022; Tran et al., 2023), asking questions (Alic et al., 2022; Feldhus et al., 2024; Jensen et al., 2021; Tran et al., 2023), and giving supportive state-

ments (Hunkins et al., 2022). These models are used to give feedback to teachers, showing, for example, the frequency of the target behavior in the discussion (Demszky et al., 2023; Jensen et al., 2020; Mikeska et al., 2024; Jensen et al., 2021).

More recently, LLMs have been used for holistic assessment of classroom interactions, including how effectively teachers support cognitive and language development (Whitehill and LoCasale-Crouch, 2024), to what extent classroom interactions exhibit encouragement and warmth (Hou et al., 2024), and how well tutors respond to students’ math errors (Kakarla et al., 2024). However, LLMs still face challenges. For instance, ChatGPT (gpt-3.5-turbo) has low correlations with human evaluation and often fails to generate insightful and relevant suggestions for improvement (Wang and Demszky, 2023). Moreover, it overestimates instructional quality, and using its extractive summaries as inputs for the classification of instruction practices does not improve the results (Xu et al., 2024). Tran et al. (2024) have explored different task formulations to improve LLM’s assessment of instructional quality, but its best-performing method is only compatible with the metrics based on the number of utterances satisfying certain criteria. We investigate prompting and task formulation strategies that are informed by recent LLM literature and can be applied to do a holistic coherence evaluation of a classroom discussion.

2.2 Automated assessment of coherence

Prior work on evaluating the coherence of a text benefited from deep neural networks, including long short-term memory (Mesgar and Strube, 2018), rational graph convolutional networks (Mesgar et al., 2021), and pretrained language models (Duari and Bhatnagar, 2022; Jeon and Strube, 2022; Zhong et al., 2022). However, these methods considered local coherence and were evaluated on tasks that could exploit it such as judging coherent and incoherent sentence pairs (Duari and Bhatnagar, 2022; Mesgar et al., 2021; Zhong et al., 2022) and short source-summary pairs (Mesgar et al., 2021; Zhong et al., 2022). BBScore (Sheng et al., 2024) captures global text coherence but treats utterances as a sequential process. This is not always the case for classroom discussions. Indeed, local coherence based on similarities of adjacent utterances had low correlations with human ratings of classroom discourse coherence (Boyle and Crossley, 2024).

LLMs enable a more holistic evaluation of coher-

ence without modeling local coherence. Naismith et al. (2023) evaluated the coherence of pieces of writing in an English language test used for higher-education admissions, based on the Common European Framework of Reference for Languages. Liu et al. (2023) proposed using an automatic Chain of Thought (Auto CoT) to generate steps for LLMs to follow when evaluating coherence. Liusie et al. (2024) showed prompting LLMs to compare texts is more effective than prompting them to assign numerical scores. However, these studies used relatively short texts, such as essays and summaries. We incorporate their insights and evaluate the methods with long, multi-party classroom discussions.

3 Data

We used the dataset collected in previous studies Mikeska et al. (2023, 2025), where elementary pre-service teachers facilitated an argumentation-focused discussion in mathematics or science with five fictional student avatars controlled by a human actor using voice modulation software. The human actor is instructed to reflect each avatar’s personality, background, and interest (e.g., “Emily is an introverted, studious, independent, serious, and literal child.”) given by the researchers. Science discussions involved the Mystery Powder (MP) task (Mikeska et al., 2021), where students constructed arguments about the identity of a mystery powder based on its properties such as color, texture, and weight, and determined which properties were useful to identify it. The mathematics discussions focused on the Ordering Fractions (OF) task (Howell et al., 2021), where the learning goal was to evaluate and contrast strategies for ordering fractions with varying numerators and denominators. The teachers were given handouts on the simulated environment, the students’ work before the discussion, and the goal of the discussion a week prior to the discussion. The teachers had up to 20 minutes to lead the discussion. Each discussion was video-recorded, transcribed, and timestamped for manual evaluation. One teacher facilitated at most two discussions in the dataset. Table 1 shows snippets of example transcripts from the MP task; Table 2 shows the descriptive statistics of the datasets.

The rubrics for human scoring have five dimensions, each with 2-3 supporting indicators (GO Discuss Project, 2021). Depending on the data collection phase, dimensions have three or four discrete levels and indicators are continuous be-

tween 1-3 or 1-4. This study focuses on Indicator 2A (“Overall Coherence of the Discussion”) in Dimension 2 (“Facilitating a Coherent and Connected Discussion”). This indicator measures if a teacher leads a well-organized discussion focused on the content and uses the time allotted to address the given learning goal (the full rubrics are in Appendix A). Raters were current and retired K-12 teachers in STEM (Nazaretsky et al., 2023). About 27% of the discussions were double-scored; the intra-class correlations (Shrout and Fleiss, 1979) were 0.630 (MP) and 0.588 (OF). Both have moderate reliability (Koo and Li, 2016), commensurate with other dimensions (Ilagan et al., 2024; Nazaretsky et al., 2023) and other publicly available data on coherence (Gopalakrishnan et al., 2019). Raters optionally provided quotes to justify their scores.

We map a score x on the 1-4 scale to 1-3 by $\frac{2}{3}x + \frac{1}{3}$. The score distributions are in Figure 1.

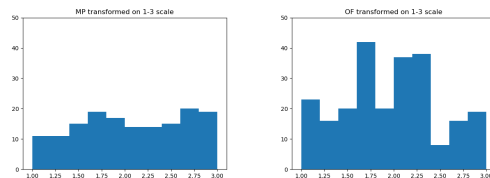


Figure 1: Score histograms in MP (left) and OF (right).

4 Experiment Setup

The MP and OF datasets were used differently. We used the MP data to develop prompts and select promising strategies; the OF data was used to test the generalization of the MP-based selections to a new domain (RQ2). For the MP data, we created four sets: five discussions used for reference (Refs), two development sets (Dev1 and Dev2), and a test set. The test set ($n = 36$) was the same as in prior work modeling other rubric dimensions (Ilagan et al., 2024; Nazaretsky et al., 2023). Using all the non-test data, we identified the five Refs discussions (see Section 5.2), then randomly chose 71 development discussions from the remainder of the data, randomizing by teachers (all discussions by the same teacher were in the same partition). We then divided the set of 71 discussions into two groups (35 and 36): Dev1 was used for experimentation with prompts, and Dev2 was used to select the most promising strategies for final testing on the MP and OF test data. For the OF dataset, we first sampled, by teacher, half the discussions for the test set ($n = 106$, from 71 teachers) and then

Coherent (human rating = 2.6)	Incoherent (human rating = 1.4)
<p>Teacher: How about we all take about a minute to look at our own shared workspaces? ... just talk to your partner next to you about things you want to bring to the discussion about how you got to your answer, your claim, your reasoning, and just think of some evidence. So that way, when someone has a question, you can answer that question because in this discussion, it's going to be all of you having more of a discussion, and me just listening and answering questions here and there. Does that sound good?</p> <p>Will: Yeah. Okay.</p> <p>⋮</p> <p>Teacher: ... I'm going to let you take the lead like I said, so we have to make sure that we don't talk over each other and that once, and I don't have to have a conversation at all. ... So the conversation can go, Carlos to Jayla. Mina can talk to Jayla. You don't have to raise your hand, and you don't have to go in order. ... I just want to make sure that everyone understands and make sure everyone has the right answer. ...</p> <p>Carlos: Well, my question is for Mina and Will, and I was just wondering why you think that it's flour?</p> <p>Will: Well, we think that it's flour because we looked at the texture and the color and the weight, and they all matched flour. So it was pretty obvious.</p> <p>⋮</p> <p>Teacher: Sometimes it's easier to learn from classmates. It's sometimes easier to learn from your classmates than a teacher teaching and lecturing you, huh?</p> <p>Emily: Yeah. I thought everyone had really good ideas. [End of discussion]</p>	<p>Teacher: Today we're going to review what we've been doing for the last couple of classes. We are going to be working on identifying a substance based on its properties. Can anybody tell me what properties are? All right, Mina, what are properties?</p> <p>Mina: ... the properties are ... like what the powder has.</p> <p>Teacher: Right, like maybe characteristics?</p> <p>⋮</p> <p>Teacher: When we're looking at properties, you might think of a bear might have different properties than a snake. ... A bear has fur, a snake has scales.</p> <p>⋮</p> <p>Teacher: Yeah. What about you Jayla and Emily? You still think it's baking soda?</p> <p>Jayla: Yeah.</p> <p>Teacher: Well, you guys are right. It's baking soda. [End of discussion]</p>

Table 1: Snippets of a coherent discussion and an incoherent one from the MP task.

	Mystery Powder (MP)	Ordering Fractions (OF)
# Transcripts	157	241
# Teachers	81	142
Av. # Utterances per Transcript	97.6	99.5
Av. # Words per Transcript	1919.6	2090.2
Av. Duration (mins)	14.5	16.7
Av. Coherence score	2.05	1.93

Table 2: Descriptive statistics of the datasets.

chose five discussions from the rest for the OF Refs set.

We test our method with GPT-4o on Azure OpenAI,¹ setting the temperature to 0 to reduce randomness. We evaluated GPT-4o predictions vs human scores using Pearson and Spearman correlations and mean squared error (MSE). For double-rated

discussions, we averaged the two scores.

5 Task formulation strategies

We describe how we design our prompts. The actual prompts are in Appendix B.

5.1 Prompts to assess a single discussion

NAIVE BASELINE We prompt GPT-4o to score discussion coherence on a scale of 1-3 based on

¹<https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models>

the rubric given to human raters. This rubric implements a score-level characterization strategy that describes what to expect to see in a discussion at a given score level. In addition, we give the background information on the topic, the learning goal of the classroom discussion, and the names of the student avatars. We add a new characteristic based on the justifications provided by the raters: They often pointed out that a coherent discussion had an introduction with clear and detailed learning goals and expectations. NAIVE BASELINE implements CoT (Wei et al., 2022), instructing GPT-4o to output the reasoning behind its score.

STRUCTURED CoT (ST. CoT) The rubric in the NAIVE BASELINE prompt characterizes highly coherent and incoherent discussions. We hypothesize that this design may prevent GPT-4o from understanding the aspects of coherence, each of which can be present or absent, or done well or badly, in a given discussion. We summarize these aspects into six bullet points and instruct GPT-4o to consider them when deriving a holistic score. This aspect-based rubric resembles the CoT prompt used to evaluate the coherence of shorter texts (Liu et al., 2023).

QUOTES Wang and Demszky (2023) have shown that LLMs cannot extract utterances relevant to instructional quality from classroom transcripts and that instructing LLMs to generate reasoning does not improve the correlation with human graders. On the other hand, Naismith et al. (2023) have found that LLMs cite examples from essays that contribute to coherence and that asking for rationale increases the correlation with humans. This line of work implies that the evaluation by LLMs can be improved if they can quote the right examples. Thus, we ask GPT-4o to provide quotes supporting the rating in CoT.

NEGATIVE FOCUS Prior work has shown that LLMs overestimate instructional quality (Xu et al., 2024). This tendency, known as leniency bias (Thakur et al., 2024), is observed when LLMs act as a judge even outside of education. Li et al. (2024) achieved better alignment between LLM and human judgment by training LLMs to generate a critical review before the final judgment. Since we use GPT-4o without fine-tuning, we ask GPT-4o to “conclude to what extent (mostly, somewhat, or seldom) the teacher **failed** to shape a coherent discussion and build ideas toward a learning goal” to make GPT-4o’s reasoning critical.

5.2 Comparison between discussions

The comparison strategy is motivated by the finding that LLMs are better at comparison than assigning numerical scores, including for evaluating the coherence of summaries (Liusie et al., 2024). However, we need $O(N^2)$ comparisons to compare all discussions and fully rank them. To reduce the cost, we compare a discussion with a small set of reference discussions. Reference discussions (referred to as Refs in Section 4) are chosen so that they (1) are not in the test set, (2) are rated by two raters, and (3) have an average score between 1.7 and 2.3 on the scale of 1-3 (i.e., middle-level performance). Of all the discussions that fit the criteria, we picked five with the smallest difference in the ratings between the two raters. The number five is based on the literature on the evaluation of automated summaries that found the comparison to 4-5 reference summaries was optimal (Nenkova and Passonneau, 2004). For each reference discussion, we ask the LLM whether the discussion-to-score is better/worse than or similar to the reference. If the discussion-to-score is better than the reference, we assign a score of 3; if it is similar – 2; worse – 1. For the final continuous score, we average the scores across the reference discussions.

We incorporate the comparison paradigm into the NAIVE BASELINE prompt and the best-performing formulation strategy for a single discussion on Dev 2 by changing the LLM’s task from rating to comparison. The definition of coherence in NAIVE BASELINE stays the same, except that it is now characterized by highly coherent, moderately coherent, and incoherent, instead of the score levels. We call this NAIVE BASELINE COMPARISON. Also, we apply the comparison formulation to the CoT outputs of the best strategy for a single discussion on Dev 2 because the reasoning provided by CoT might be a good summary of the degree of coherence of a discussion. We call this <STRATEGY NAME> (2 STEP), where <strategy name> is determined in the next section.

6 Results on MP dev data (Dev 2)

The top pane of Table 3 shows the results for single-discussion strategies on the MP Dev 2 set. ST. CoT has the lowest MSE. QUOTES has the best Pearson correlation but has the worst MSE. NEGATIVE FOCUS trails behind the other methods. Thus, we combine the two most promising strategies, ST. CoT and QUOTES. The combination shows the

Single discussion strategies	Pearson	Spearman	MSE
NAIVE BASELINE	.503 (.458-.542)	.447 (.405-.495)	.533 (.516-.558)
ST. CoT	.480 (.405-.580)	<u>.499</u> (.426-.594)	.335 (.276-.373)
QUOTES	.542 (.491-.608)	.497 (.443-.557)	.598 (.564-.620)
NEGATIVE FOCUS	.469 (.436-.504)	.478 (.456-.510)	.483 (.459-.504)
ST. CoT+QUOTES	<u>.512</u> (.468-.567)	.565 (.526-.629)	<u>.359</u> (.329-.387)
Comparison strategies	Pearson	Spearman	MSE
NAIVE BASELINE COMPARISON	.584 (.562-.604)	.607 (.587-.631)	.326 (.316-.334)
ST. CoT+QUOTES COMPARISON	<u>.555</u> (.538-.572)	<u>.596</u> (.557-.625)	.496 (.456-.549)
ST. CoT+QUOTES (2-STEP)	.538 (.505-.590)	.550 (.506-.628)	<u>.352</u> (.316-.399)

Table 3: Pearson and Spearman correlations (higher numbers are better) and MSE (lower numbers are better) for the single-discussion formulations (top) and the comparison formulations (bottom) on Dev 2. We report an average and a range of five runs. The best result is in bold, and the second-best result is underlined.

Strategies	MP (n = 36)			OF (n = 106)		
	Pearson	Spearman	MSE	Pearson	Spearman	MSE
NAIVE BASELINE	.574 (.548-.599)	.578 (.547-.611)	.493 (.481-.504)	.167 (.140-.210)	.139 (.110-.183)	.754 (.715-.788)
ST. CoT + QUOTES	.663 (.592-.730)	.607 (.541-.692)	.272 (.233-.317)	.416 (.389-.447)	.420 (.394-.451)	.405 (.379-.439)
NAIVE BASELINE COMPARISON	.708 (.686-.732)	.702 (.664-.736)	.236 (.219-.252)	.308 (.280-.341)	.328 (.295-.365)	.523 (.507-.553)

Table 4: Results on test sets; reported are the average and range of five runs. The best performance is in bold.

best or second-best performance and outperforms the NAIVE BASELINE on all metrics. Therefore, we create ST. CoT+QUOTES (2 STEP) as a 2-step comparison strategy.

The bottom pane of Table 3 shows the results for comparison strategies. The results support the effectiveness of comparing the discussion-to-be-scored with references. The comparison versions of NAIVE BASELINE and ST. CoT+QUOTES perform better than their single-discussion versions on all metrics, both in terms of average performance and stability (narrower range), apart from MSE for ST. CoT+QUOTES. The results do not support the two-step formulation. This implies that the description of a discussion does not capture the information necessary for the comparison, consistently with prior literature (Xu et al., 2024).

For the final evaluation of test data, we select NAIVE BASELINE COMPARISON, as it showed the best performance on Dev 2. We also evaluate the ST. CoT+QUOTES single-discussion formulation, since it performs best in the more resource-lean scenario without reference discussions. The NAIVE BASELINE scoring scenario will also be evaluated on test data to check whether gains over baseline are replicated in the test results.

7 Final test results

Table 4 shows the results on the test sets of MP and OF. To answer RQ1 (generalizability of strategies to classroom discussion), we compare the rows. Our results support the generalization of the strategies evaluated on short text to long classroom discussions: Both ST. CoT+QUOTES and NAIVE BASELINE COMPARISON outperformed NAIVE BASELINE on all metrics. RQ2 (generalizability across subject domains) is answered by comparing the columns. We observe that the performance on the OF data is generally much worse, across formulations and metrics, than on MP data.

We further analyze how the task formulation strategies address the limitations of LLMs in assessing instructional quality found in the literature.

Overestimation of quality One of the limitations is that LLMs tend to overestimate the instructional quality (Xu et al., 2024). To check this tendency, we plot GPT-4o predictions vs human scores in Figure 2, using the runs with the median MSE out of five. The NAIVE BASELINE exhibits overestimation, as most of the points are above the diagonal; the median scores are 2.75 (MP) and 2.5 (OF). ST. CoT+QUOTES reduces the median scores to 2.5

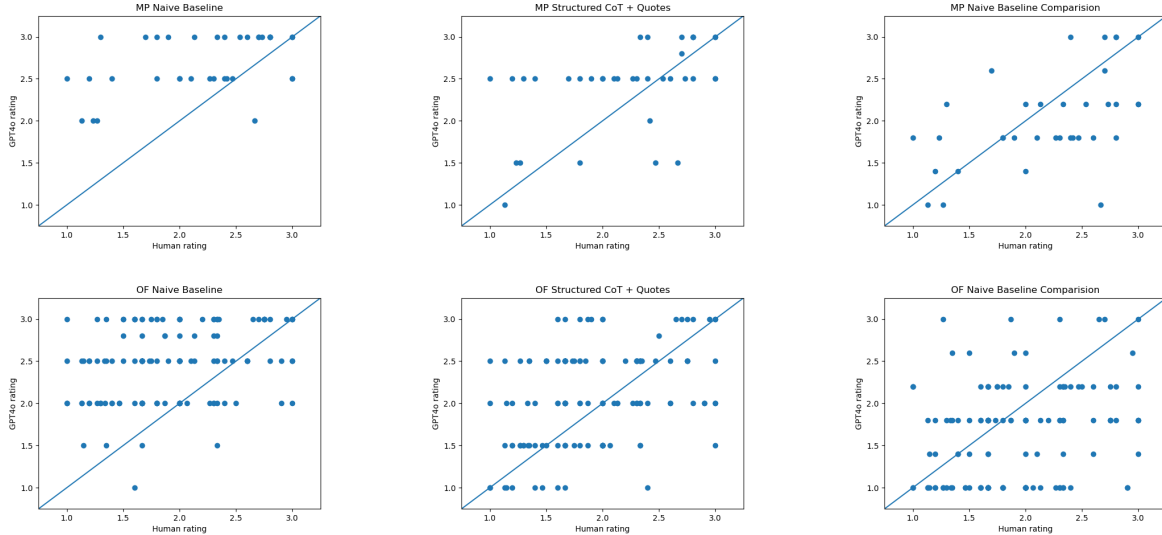


Figure 2: Scatter plots of GPT-4o predictions vs human scores. The top row is MP (Dev 2), and the bottom row is OF (discussions not in Refs or test set). The points above the diagonal are over-estimations by GPT-4o.

(MP) and 2 (OF). NAIVE BASELINE COMPARISON further pushes down the scores; >60% of the discussions receive 2 ± 0.2 points for MP, and >90% of the discussions receive 1.6 ± 0.6 points for OF. Thus, our results confirm the over-scoring by NAIVE BASELINE observed in the literature and suggest that the ST. CoT+QUOTES and NAIVE BASELINE COMPARISON formulations help reduce it.

Inability to provide relevant quotes Another limitation is that many quotes provided by LLMs are unfaithful or irrelevant (Wang and Demszky, 2023). Therefore, we investigated the quotes produced by ST. CoT+QUOTES, using the same runs as above. We sampled six discussions (three MP, three OF) with at least 50 words in their human justifications. ST. CoT+QUOTES provided more quotes than humans: 4.33 vs 1.83 per discussion, on average. All quotes given by ST. CoT+QUOTES exist in the discussions. In five out of six discussions, the system picked at least one quote also picked by a human. Generally, the system picked quotes relevant to the specific aspect; see Table 5 for example output. However, the *interpretations* of the quote were not always aligned. For example, “Do you all feel like you learned something from this?” was cited as justification for a critical evaluation of the teacher’s conclusion by a human rater (“Conclusion is minimal”), whereas the system picked the same quote to justify the evaluation “The teacher summarized important conclusions consistently throughout the discussion.” Thus, while the system’s quotes show

the grounding of the aspects in the discussion, they do not consistently support the *evaluation* of the aspect. Other five discussions are in Appendix C.

8 Discussion

Our results show the promise for generalizability of task formulation strategies for the evaluation of the coherence of short text to the evaluation of teachers’ ability to lead coherent discussions. Further analyses suggest that these strategies address previously reported challenges: overestimation bias (Xu et al., 2024) and the inability to extract relevant parts of discussions (Wang and Demszky, 2023). However, we also highlight two challenges in deploying GPT-4o in classroom discussion assessment.

The first challenge is the generalizability across subject domains (RQ2). This would be a bottleneck in real-world applications because it implies that we should tune prompts for every subject, increasing cost and time and making it difficult to generalize without a decent-sized corpus. Moreover, inconsistent scoring across subjects threatens fairness in professional development and evaluation of teachers. For instance, if we used GPT-4o’s assessment to determine the need for professional development, teachers in the more mis-scored subjects (math in our example) would suffer from more mis-allocation of resources such as coaching sessions than teachers in other subjects even when teaching practice is comparable.

The second challenge is the misalignment in the interpretation of discussions and quotes between

Human	<p>The discussion is mostly organized, purposeful, and focused on the content at hand. The teacher presents the task and sets expectations for the discussion. She systematically moves from group to group, sharing key ideas. Transitions are timely and appropriate. Conclusion is minimal (“Yeah, you did. And everyone had good methods. Do you all feel like you learned something from this? Sometimes it’s easier to learn from classmates. It’s sometimes easier to learn from your classmates than a teacher teaching and lecturing you, huh?”)</p>
GPT	<ol style="list-style-type: none"> 1. The teacher did not state a clear, detailed learning goal and expectation at the beginning. The teacher mentioned that the students would discuss their findings and answer questions, but did not explicitly state the learning goal of constructing an argument based on properties. For example, the teacher said, ‘in this discussion, it’s going to be all of you having more of a discussion, and me just listening and answering questions here and there.’ 2. The teacher moved between groups of ideas in a timely manner with apparent rationale. The teacher facilitated the discussion by asking each group to share their findings and then prompted students to ask questions, which kept the discussion focused and organized. For example, ‘All right. So, let’s just pick one person from our group... Mina and Will, go ahead and pick one person to share what your claim was, what your evidence was and what your answer was.’ 3. The teacher instructed with deliberation, ensuring that students understood the process and the importance of respectful conversation. For example, ‘I’m going to let you take the lead like I said, so we have to make sure that we don’t talk over each other.’ 4. The teacher introduced well-chosen new ideas that benefited student understanding at relevant moments. For example, the teacher clarified the importance of checking all properties but also understanding which ones are most useful, ‘Sometimes in science, and like Carlos said, Carlos did check the weight. He just realized that the weight was not as useful.’ 5. The teacher summarized important conclusions consistently throughout the discussion. For example, ‘Does everyone understand how Carlos did that?’ and ‘Do you all feel like you learned something from this?’ 6. The teacher connected a series of ideas to the learning goal by guiding students to understand which properties were most useful in identifying the mystery powder. For example, ‘Carlos, I’m reading your answer. Could you maybe tell us maybe one of the properties that aren’t that important? Maybe that would help Mina and Will.’ <p>Overall, the discussion was coherent and well-facilitated, but the initial learning goal could have been stated more clearly.</p>

Table 5: Human evaluation and STRUCTURED COT + QUOTES (GPT) output for the coherent example in Table 1. The green and red texts represent agreement and disagreement between the human and STRUCTURED COT + QUOTES, respectively. The blue texts are quotes from the discussion appearing in both evaluations.

humans and GPT-4o. This is concerning when we base feedback for teachers on GPT-4o’s outputs because it would give teachers incorrect action items or miss opportunities for improvement. In the example in Table 5, the human evaluation suggests the conclusion is the area for improvement; the teacher could have elaborated more than just saying “Do you all feel like you learned something from this? ...” However, ST. COT+QUOTES identifies it as a good conclusion. Instead, it recommends “explicitly stating the learning goal of constructing an argument based on properties,” which was al-

ready achieved according to the human evaluation. This misalignment could undermine the validity and usability of GPT-4o in generating something more than scores, including feedback. A potential remedy could be retrieving relevant pre-defined human-written feedback based on the score, but it cannot fully utilize LLMs’ advantages in flexibility and personalization. This motivates future work on in-depth analysis of human and LLM quotes and on improving the evaluation of quotes selected by GPT-4o.

9 Conclusion

We evaluated task formulation strategies to assess the coherence of classroom discussions. Our results show that strategies previously evaluated for assessing the coherence of short text, such as essays or summaries, successfully generalize to assessing much longer texts—transcripts of 20-minute-long simulated classroom discussions. We reveal that these strategies help GPT-4o tackle the limitations pointed by the literature: overestimation of instructional quality and failure to quote relevant utterances from discussions. However, they do not show cross-domain generalization even within the same simulated setting. Our study serves as a step toward supporting teachers’ development with automated personalized feedback by providing accurate automated evaluation of the target skill, though challenges still remain.

Limitations

We acknowledge the limitations of our evaluation. First, the generalizability of our findings should be explored with other LLMs and datasets. Our results demonstrate some generalizability of coherence evaluation methods from other genres (essays and summaries) to our context but also show that generalization across STEM subjects within the same simulated classroom context is not straightforward since performance is lower on OF than MP. Improving generalization across content domains is our most immediate goal. In addition, we implicitly show the generalizability across models because the prior work our prompts are based on uses models different from ours: GPT-4 for ST, CoT (Liu et al., 2023) and QUOTES (NAISMITH ET AL., 2023) and open-source LLMs, including FlanT5 and Llama2, for the comparison strategy (Liusie et al., 2024). Although our results imply that the strategies in this paper are potentially generalizable to other models, further experiments would be necessary to verify it.

Second, our implementation of the comparison formulation compares discussions only with moderately coherent reference discussions and results in excessive lowering of scores. We leave it to future work to explore strategies for selecting reference discussions that could help mitigate this excessive correction of over-scoring.

Third, human quotes are not the “gold standard” since the raters were asked to provide some examples from the discussion (see Section 3); there are

potentially other good quotes that weren’t selected. The analysis in Section 7 motivates future work to improve the evaluation of quotes selected by GPT-4o.

Finally, our experiments are done only in simulated classrooms. These are important for scaling up practice opportunities by allowing teachers to repeat the cycle of practice and reflection on their teaching without harming real students by their mistakes (Dalinger et al., 2020; Dieker et al., 2014). Generalizability to real classrooms with real students is also important. However, since our goal of scaling up feedback aligns better with the advantages of simulated classrooms, we prioritized this exploration on data from simulated discussions, leaving exploration of real-life discussions to future work.

Ethical considerations

We would like to address potential ethical concerns. First, giving student names and the whole discussions to GPT-4o is not a breach of privacy. In this work, we are not using data from real elementary students. Instead, all the data comes from responses from elementary student avatars in a simulated classroom. The student avatars are operated by an adult, called a simulation specialist, who is trained to use specialized equipment (e.g., game controllers, voice modulation software, etc.) to sound, move, and respond like upper elementary students (cf. Section 3). Each teacher participant signed a consent form that provides their written approval for the research use of the video-recorded discussion and who it can and cannot be shared with. Video recordings are only shared outside of our research team if the participant has consented to that use. For this study, no video recordings were used; we used de-identified transcripts for analyses.

Second, LLMs could be susceptible to their algorithmic biases. Our work addresses bias concerns by showing how to reduce overestimation (bias against low-performing teachers) in Section 7. The model’s explanations could be biased, too, and might not be pedagogically sound (cf. Section 7). As discussed above, instead of giving teachers the model’s explanations as they are as feedback, we plan to use the scores and outputs to provide the teachers with feedback by, for example, retrieving relevant pre-defined human-written feedback.

The costs of using GPT-4o and collecting and scoring discussion data for model development

could also be a barrier to applying our results to the real world. However, the best performing method, STRUCTURED COT + QUOTES, is zero-shot and does not require any reference discussions. Thus, it works well in resource-constrained settings. The price of GPT-4o is \$0.00250 / 1K input tokens and \$0.01000 / 1K output tokens as of writing. Since the average number of words in discussions is around 2K and the output is usually no more than 500 words (cf. Tables 2 and 5), the cost per discussion is less than \$0.1. Therefore, our method scales well at low cost.

Acknowledgments

This study used data collected on previous projects funded by the National Science Foundation (grants #2037983, #1621344, and #2032179). The opinions expressed herein are those of the authors and not the funding agency. We are grateful for the elementary pre-service teachers who participated in these earlier studies and for the contributions of practicing and retired teachers in scoring these discussion transcripts.

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.
- Jessica Boyle and Scott Crossley. 2024. [Semantic similarity of teacher and student discourse linked to quality ratings from classroom observations](#). In *Proceedings of Educational Data Mining (EDM)*, pages 797–801.
- James P Bywater, Jennifer L Chiu, James Hong, and Vidhya Sankaranarayanan. 2019. The teacher responding tool: Scaffolding the teacher practice of responding to student ideas in mathematics classrooms. *Computers & Education*, 139:16–30.
- Stacey L Carpenter, Jiwon Kim, Katherine Nilsen, Tobias Irish, Julie A Bianchini, and Alan R Berkowitz. 2020. Secondary science teachers’ use of discourse moves to work with student ideas in classroom discussions. *International Journal of Science Education*, 42(15):2513–2533.
- XI Chen and Yeping Li. 2010. Instructional coherence in chinese mathematics classroom—a case study of lessons on fraction division. *International Journal of Science and Mathematics Education*, 8:711–735.
- Tara Dalinger, Katherine B Thomas, Susan Stansberry, and Ying Xiu. 2020. A mixed reality simulation offers strategic practice for pre-service teachers. *Computers & Education*, 144:103696.
- C. Danielson. 2013. *The Framework for Teaching: Evaluation Instrument*. Danielson Group.
- Dorottya Demszky, Jing Liu, Heather C. Hill, Dan Jurafsky, and Chris Piech. 2023. Can automated feedback improve teachers’ uptake of student ideas? Evidence from a randomized controlled trial in a large-scale online course. *Educational Evaluation and Policy Analysis*.
- Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori B Hashimoto. 2021. Measuring conversational uptake: A case study on student-teacher interactions. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 1638–1653.
- Lisa A Dieker, Jacqueline A Rodriguez, Benjamin Lignugaris/Kraft, Michael C Hynes, and Charles E Hughes. 2014. The potential of simulated environments in teacher education: Current and future possibilities. *Teacher Education and Special Education*, 37(1):21–33.
- Swagata Duari and Vasudha Bhatnagar. 2022. Ffcd: A fast-and-frugal coherence detection method. *IEEE Access*, 10:85305–85314.
- Nils Feldhus, Alik Anagnostopoulou, Qianli Wang, Milad Alshomary, Henning Wachsmuth, Daniel Sonntag, and Sebastian Möller. 2024. Towards modeling and evaluating instructional explanations in teacher-student dialogues. In *Proceedings of Information Technology for Social Good*, page 225–230.
- Joan Ferrini-Mundy, Gail Burrill, and William H Schmidt. 2007. Building teacher capacity for implementing curricular coherence: Mathematics teacher professional development tasks. *Journal of Mathematics Teacher Education*, 10:311–324.
- GO Discuss Project. 2021. [Scoring](#). Qualitative Data Repository.
- Karthik Gopalakrishnan, Behnam Hedayatnia, Qianling Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. [Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations](#). In *Proc. Interspeech 2019*, pages 1891–1895.
- Deborah Hanuscin, Kelsey Lipsitz, Dante Cisterna-Alburquerque, Kathryn A Arnone, Delinda van Garderen, Zandra de Araujo, and Eun Ju Lee. 2016. Developing coherent conceptual storylines: Two elementary challenges. *Journal of Science Teacher Education*, 27:393–414.
- 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 *Proceedings of Artificial Intelligence in Education (AIED)*, pages 60–74.
- Heather Howell, Jamie Mikeska, Jessica Tierney, Benjamin Baehr, and Penny Lehman. 2021. [Conceptualization and development of a performance task for assessing and building elementary preservice teachers' ability to facilitate argumentation-focused discussions in mathematics: The ordering fractions task](#). *ETS Research Memorandum No. 21-10*.
- 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 Learning Analytics and Knowledge (LAK)*, 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 Innovative Use of NLP for Building Educational Applications (BEA)*, pages 182–198.
- 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 Human Factors in Computing Systems*, pages 1–13.
- Emily Jensen, Samuel L. Pugh, and Sidney K. D'Mello. 2021. [A deep transfer learning approach to modeling teacher discourse in the classroom](#). In *Proceedings of Learning Analytics and Knowledge (LAK)*, page 302–312.
- Sungho Jeon and Michael Strube. 2022. [Entity-based neural local coherence modeling](#). In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 7787–7805.
- Sanjit Kakarla, Danielle Thomas, Jionghao Lin, Shivang Gupta, and Ken Koedinger. 2024. Using large language models to assess tutors' performance in reacting to students making math errors. In *AI for Education: Bridging Innovation and Responsibility at the AAAI Conference on AI (AAAI)*.
- Sean Kelly, Robert Bringe, Esteban Aucejo, and Jane Cooley Fruehwirth. 2020. Using global observation protocols to inform research on teaching effectiveness and school improvement: Strengths and emerging limitations. *Education Policy Analysis Archives*, 28:62–62.
- Terry K Koo and Mae Y Li. 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2):155–163.
- Matthew A Kraft, David Blazar, and Dylan Hogan. 2018. The effect of teacher coaching on instruction and achievement: A meta-analysis of the causal evidence. *Review of Educational Research*, 88(4):547–588.
- Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Pengfei Liu, and 1 others. 2024. Generative judge for evaluating alignment. In *The Twelfth International Conference on Learning Representations*.
- 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 Empirical Methods in Natural Language Processing (EMNLP)*, pages 2511–2522.
- 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 European Chapter of the Association for Computational Linguistics (EACL)*, pages 139–151.
- Mohsen Mesgar, Leonardo F. R. Ribeiro, and Iryna Gurevych. 2021. [A neural graph-based local coherence model](#). In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2316–2321.
- Mohsen Mesgar and Michael Strube. 2018. [A neural local coherence model for text quality assessment](#). In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 4328–4339.
- Jamie N. Mikeska, Beata Beigman Klebanov, Alessia Marigo, Jessica Tierney, Tricia Maxwell, and Tanya Nazaretsky. 2024. Exploring the potential of automated and personalized feedback to support science teacher learning. In *Proceedings of Artificial Intelligence in Education (AIED)*, pages 251–258.
- Jamie N Mikeska, Dionne Cross Francis, Pamela S Lottero-Perdue, Meredith Park Rogers, Calli Shekell, Pavneet Kaur Bharaj, Heather Howell, Adam Maltese, Meredith Thompson, and Justin Reich. 2025. Promoting preservice teachers' facilitation of argumentation in mathematics and science through digital simulations. *Teaching and Teacher Education*, 154:104858.
- Jamie N Mikeska, Heather Howell, Joseph Ciofalo, Adam Devitt, Elizabeth Orlandi, Kenneth King, and G Simonelli. 2021. Conceptualization and development of a performance task for assessing and building elementary preservice teachers' ability to facilitate argumentation-focused discussions in mathematics: The mystery powder task. *Research Memorandum No. RM-21-06, Educational Testing Service*.
- Jamie N Mikeska, Heather Howell, and Devon Kinsey. 2023. 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.
- Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. [Automated evaluation of written discourse coherence](#)

- using GPT-4. In *Proceedings of Innovative Use of NLP for Building Educational Applications (BEA)*, pages 394–403.
- National Research Council. 2012. A framework for k-12 science education: Practices, crosscutting concepts, and core ideas. *National Academy of Sciences*.
- Tanya Nazaretsky, Jamie N Mikeska, and Beata Beigman Klebanov. 2023. Empowering teacher learning with ai: Automated evaluation of teacher attention to student ideas during argumentation-focused discussion. In *Proceedings of Learning Analytics and Knowledge (LAK)*, pages 122–132.
- Ani Nenkova and Rebecca J Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 145–152.
- RC Pianta. 2008. Classroom assessment scoring system™: Manual k-3. *Paul H Brookes Publishing*.
- Julia D Plummer and Arzu Tanis Ozcelik. 2015. Preservice teachers developing coherent inquiry investigations in elementary astronomy. *Science Education*, 99(5):932–957.
- John Ramsey. 1993. Developing conceptual storylines with the learning cycle. *Journal of Elementary Science Education*, 5(2):1–20.
- Kathleen J Roth, Helen E Garnier, Catherine Chen, Meike Lemmens, Kathleen Schwille, and Nicole Wickler. 2011. Videobased lesson analysis: Effective science pd for teacher and student learning. *Journal of Research in Science Teaching*, 48(2):117–148.
- Zhecheng Sheng, Tianhao Zhang, Chen Jiang, and Dongyeop Kang. 2024. Bbscore: A brownian bridge based metric for assessing text coherence. In *Proceedings of the AAAI Conference on AI (AAAI)*, pages 14937–14945.
- Patrick E Shrout and Joseph L Fleiss. 1979. Intraclass correlations: uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420.
- Mary Kay Stein, Randi A Engle, Margaret S Smith, and Elizabeth K Hughes. 2008. Orchestrating productive mathematical discussions: Five practices for helping teachers move beyond show and tell. *Mathematical Thinking and Learning*, 10(4):313–340.
- Abhijit Suresh, Jennifer Jacobs, Margaret Perkoff, James H. Martin, and Tamara Sumner. 2022. Fine-tuning transformers with additional context to classify discursive moves in mathematics classrooms. In *Proceedings of Innovative Use of NLP for Building Educational Applications (BEA)*, pages 71–81.
- Abhijit Suresh, Tamara Sumner, Jennifer Jacobs, Bill Foland, and Wayne Ward. 2019. Automating analysis and feedback to improve mathematics teachers’ classroom discourse. In *Proceedings of the AAAI conference on AI (AAAI)*, pages 9721–9728.
- Wang Tao, Cai Jinfa, and Hwang Stephen. 2015. Achieving coherence in the mathematics classroom: Toward a framework for examining instructional coherence. In *How Chinese Teach Mathematics: Perspectives From Insiders*, pages 111–148. World Scientific.
- Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2024. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. *arXiv preprint arXiv:2406.12624*.
- Nhat Tran, Benjamin Pierce, Diane Litman, Richard Correnti, and Lindsay Clare Matsumura. 2023. Utilizing natural language processing for automated assessment of classroom discussion. In *Proceedings of Artificial Intelligence in Education (AIED). Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pages 490–496.
- Nhat Tran, Benjamin Pierce, Diane Litman, Richard Correnti, and Lindsay Clare Matsumura. 2024. Analyzing large language models for classroom discussion assessment. In *Proceedings of Educational Data Mining (EDM)*, pages 500–510.
- 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 Innovative Use of NLP for Building Educational Applications (BEA)*, pages 626–667.
- 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.
- 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*.
- 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 North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4375–4389.
- Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 2023–2038.

A Rubrics

Human raters evaluated a teacher’s ability to lead a coherent discussion based on the rubrics in Table 6

Level Label	Description
1 Beginning	Discussion has a weak sense of organization, purpose, and focus.
2 Developing	Discussion is somewhat organized, purposeful, and focused on the content at hand. AND/OR Portions of the discussion are strongly variable with respect to organization, purpose, and focus.
3 Well-prepared	Discussion is mostly organized, purposeful, and focused on the content at hand.
4 Commendable	Discussion is organized, purposeful, and focused on the content at hand. AND The teacher uses the time allotted so that the learning goal is achieved.

Table 6: Rubrics for Indicator 2A (“Overall Coherence of the Discussion”) (GO Discuss Project, 2021)

and the observation notes in Table 7 (GO Discuss Project, 2021).

B LLM Prompts

B.1 Naive baseline

The prompt used as a baseline is the following:

Following is a discussion about <task information>. <task learning goal> <student information> <rating instruction> <coherence definition>

A score of 3 is characterized by <score 3 characteristics>

A score of 2 is characterized by <score 2 characteristics>

A score of 1 is characterized by <score 1 characteristics>

Please output your score and reasoning in the following JSON format: {“reason”: “...”, “score”: “a float number between 1-3”}.

<the discussion to score>

<task information> and <task learning goal> are dependent on the dataset. For MP, <task informa-

For Indicator 2a, only consider evidence of organization or planning that is connected to the intended student learning goal.
A score of 4 is characterized by a strong degree of coherence around the content and ideas that are discussed and the teacher’s successful use of the available time during the discussion to address the learning goal.
A score of 3 is characterized by a strong degree of coherence around the content and ideas that are discussed. For example: <ul style="list-style-type: none"> • Transitions between ideas and/or groups are timely and make sense. • Instruction takes place in ways that suggest deliberation on the part of the teacher. • New ideas that are introduced are well chosen and occur at relevant moments. Note that you can score a 3 even if the teacher does not achieve the learning goal by the end of the discussion.
A score of 2 is characterized by a variable degree of coherence around the content and ideas that are discussed. For example, different portions of the discussion might be scored as a 1 or 3 if viewed separately. At least some portion of the discussion is highly coherent.
A score of 1 is characterized by a lack of coherence around the content and ideas that are discussed. For example: <ul style="list-style-type: none"> • Discussion has a weak sense of purpose and trajectory. • Teacher moves between ideas abruptly and without apparent rationale. • Teacher introduces new ideas that have limited potential for benefiting student understanding. • Important conclusions may be left unstated or inconsistently summarized. • Discussion may be characterized as a series of unconnected ideas taken up one at a time.

Table 7: The observation notes provided to human raters (GO Discuss Project, 2021).

tion> is “*identifying a mystery powder in a science classroom*”, and <task learning goal> is “*The learning goal is that students will construct an argument about the identity of a mystery powder based on its properties and come to a consensus about which properties are most useful in identifying the unknown powder.*” For OP, <task informa-

tion> is “*ordering fractions in a math classroom*”, and <task learning goal> is “*The learning goal is that students will evaluate, justify, compare, and contrast strategies for ordering fractions with different numerators and denominators.*” These descriptions are taken from the handouts given to the teachers in the dataset before they facilitate discussions (Mikeska et al., 2023, 2025).

<student information> is “*Mina, Will, Emily, Jayla, and Carlos are students.*”. <rating instruction> is “*Your task is to rate the discussion based on its coherence on a scale of 1-3.*” <coherence definition> is “*To be coherent, a discussion must be organized, purposeful, and focused on the content at hand, and the teacher must use the time allotted so that the learning goal is achieved.*”

<score 3 characteristics>: a strong degree of coherence around the content and ideas that are discussed and the teacher’s successful use of the available time during the discussion to address the learning goal. For example,

- The teacher states a clear, detailed learning goal and expectation at the beginning.
- Transitions between ideas and/or groups are timely and make sense.
- Instruction takes place in ways that suggest deliberation on the part of the teacher.
- New ideas that are introduced are well chosen and occur at relevant moments.

<score 2 characteristics>: a variable degree of coherence around the content and ideas that are discussed. For example, different portions of the discussion might be scored as a 1 or 3 if viewed separately. At least some portions of the discussion are highly coherent.

<score 1 characteristics>: a lack of coherence around the content and ideas that are discussed. For example:

- Discussion has a weak sense of purpose and trajectory.
- Teacher moves between ideas abruptly and without apparent rationale.
- Teacher introduces new ideas that have limited potential for benefiting student understanding.

- Important conclusions may be left unstated or inconsistently summarized.
- Discussion may be characterized as a series of unconnected ideas taken up one at a time.

These score characteristics are adopted from the observation notes in Table 7. Only the discussion is sent as a user input to GPT-4o, and the rest is sent as a system input.

B.2 Prompts for the single discussion strategies

Only the discussion is sent as a user input to GPT-4o, the rest is sent as a system input.

STRUCTURED CoT

Following is a discussion about <task information>. <task learning goal> <student information> <rating instruction> To do so, first, read the discussion carefully. Then, describe whether the teacher succeeded in doing or failed to do each of the following:

<aspects of coherence>

In the end, rate the discussion on a scale of 1-3.

Please output your description and score in the following JSON format: {“description”: “1. The teacher ...”, “score”: “a float number between 1-3”}.

<the discussion to score>

<aspects of coherence>:

1. state a clear, detailed learning goal and expectation at the beginning,
2. move between (groups of) ideas timely with apparent rationale,
3. instruct with deliberation,
4. introduce well-chosen new ideas that benefit student understanding at relevant moments,
5. summarize important conclusions consistently throughout the discussion, and
6. connect a series of ideas to the learning goal.

QUOTES The prompts is the same as NAIVE BASELINE, with the addition of the following right before "Please output your score...":

When you rate the discussion, provide quotes from it in your reasoning to support your score.

NEGATIVE FOCUS The prompts is the same as NAIVE BASELINE, with the addition of the following right before "Please output your score...":

When you rate the discussion, provide your reasoning and conclude to what extent (mostly, somewhat or seldom) the teacher failed to shape a coherent discussion and build ideas toward the learning goal.

STRUCTURED CoT + QUOTES prompt The prompts is the same as STRUCTURED CoT, with the addition of the following right before "In the end, rate...":

When you describe each of the above aspects, provide quotes from the discussion in your reasoning to support your score.

B.3 Prompts for the comparison strategies

Only the part starting from "Here's the first discussion;" is sent as a user input, the rest is sent as a system input. We optimized the ordering of the discussions for each prompts using Dev 1 because it impacts the decisions (Liusie et al., 2024).

NAIVE COMPARISON BASELINE

Following is a discussion about <task information> <task learning goal> <student information> <comparison instruction> <coherence definition>

A highly coherent discussion is characterized by <score 3 characteristics>

A moderately coherent discussion is characterized by <score 2 characteristics>

An incoherent discussion is characterized by <score 1 characteristics>

You may say that the first discussion has a similar coherence to the second one.

Please output your decision and reasoning in the following JSON format: {"reason": "...", "The first discussion is": "similar/better/worse"}.

Here's the first discussion;

<a reference discussion>

Here's the second discussion

<the discussion to score>

<comparison instruction> is "Your task is to determine whether the first discussion is better or worse than the second one based on their coherence."

ST. CoT+QUOTES COMPARISON

Following are two discussions about <task information> <task learning goal> <student information> <comparison instruction> To do so, first, read both discussions carefully. Then, for each discussion, describe whether the teachers succeeded in doing or failed to do each of the following:

<aspects of coherence>

When you describe each of the above aspects, provide quotes from the discussions in your reasoning to support your comparison.

In the end, decide whether the first discussion is better or worse than the second one. You may say that the first discussion has a similar coherence to the second one.

Please output your decision and reasoning in the following JSON format: {"reason": "1. In the first discussion, ...", "The first discussion is": "similar/better/worse"}.

Here's the first discussion;

<the discussion to score>

Here's the second discussion

<a reference discussion>

ST. CoT+QUOTES COMPARISON - TWO STEP

Step 1 prompt is the following:

Following is a discussion about <task information>. <task learning goal> <student information> Your task is to describe the coherence of the discussion. To do so, first, read the discussion carefully. Then, describe whether the teacher succeeded in doing or failed to do each of the following:

<aspects of coherence>

When you describe each of the above aspects, provide quotes from the discussion to support your description.

Step 2 prompt is the following:

Following are the evaluations of two different classroom discussions about <task information> <comparison instruction> To do so, first, read each aspect in both evaluations carefully, including the quotes in the evaluations. In the end, decide whether the first discussion is better or worse than the second one. You may say that the first discussion has a similar coherence to the second one.

Please output your decision and reasoning in the following JSON format: {"reason": "...", "The first discussion is": "similar/better/worse"}.

Here's the evaluation of the first discussion;

<the output of step 1 for a reference discussion>

Here's the evaluation of the second discussion;

<the output of step 1 for the discussion to score>

C Additional comparisons between human and ST. COT+QUOTES

Tables 8-12 have additional comparisons between justifications by human raters and ST. COT+QUOTES.

Human	<p>The discussion was mostly organized with portions that were harder to follow. At the beginning of the lesson, the teacher said (e.g., today we're going to look at everyone's claim from our experiment. And, I want us all to work on constructing, defending and critiquing ours and others' claims) and then during the discussion, references the properties chart to help guide the students toward the learning goal.</p>
GPT	<ol style="list-style-type: none"> 1. The teacher did not state a clear, detailed learning goal and expectation at the beginning. The teacher mentioned, 'I want us all to work on constructing, defending and critiquing ours and others' claims,' but did not provide a detailed learning goal or clear expectations. 2. The teacher moved between groups of ideas timely with apparent rationale. For example, the teacher transitioned from discussing the importance of weight to other properties like texture and color, and then to the reactions with vinegar and water. 3. The teacher instructed with deliberation, asking students to turn and talk, and then come back together to share their thoughts. 4. The teacher introduced well-chosen new ideas that benefited student understanding at relevant moments, such as explaining why weight might not be as important and guiding students to see that color was not a distinguishing property. 5. The teacher summarized important conclusions consistently throughout the discussion, such as reiterating that weight and color were not as useful in identifying the mystery powder. 6. The teacher connected a series of ideas to the learning goal by guiding students to use the most relevant properties (texture, reaction with vinegar, and reaction with water) to identify the mystery powder as baking soda.

Table 8: Human evaluation and STRUCTURED COT + QUOTES (GPT) output. The green and red texts represent agreement and disagreement between the human and STRUCTURED COT + QUOTES, respectively. The blue texts are quotes from the discussion appearing in both evaluations.

Human	<p>The T made a small attempt to frame the purpose of the discussion at the start by asking the students if they knew what was meant by “properties”. The students did try to provide information, but the T wound up taking over and giving a weak example of what properties were. The T had each group share their claim and evidence. After each group shared, the T critiqued their evidence while reading the groups’ prework section containing their rationale about the most important properties (Key Idea 3). There was a weak review of the properties at the end of the discussion—again with the T providing the information. There was a short “consensus” discussion where the T polled the students to see if they thought the mystery powder was baking soda. The T had a very weak ending.</p>
GPT	<ol style="list-style-type: none"> 1. The teacher did not state a clear, detailed learning goal and expectation at the beginning. The teacher mentioned, ‘Today we’re going to review what we’ve been doing for the last couple of classes. We are going to be working on identifying a substance based on its properties,’ but did not elaborate on the specific learning goal or expectations. 2. The teacher moved between ideas timely with apparent rationale, such as transitioning from discussing properties to having students share their answers and reasoning. For example, ‘Talk among yourselves for a couple of minutes about what the mystery powder is and how you can support your answer. You guys ready to share your answers with the class?’ 3. The teacher instructed with deliberation, providing clear explanations and feedback. For instance, ‘That’s a good explanation. You guys said that baking soda is white, smooth, bubbles when mixed with vinegar, looks clear when is mixed with water and weighs 24 grams.’ 4. The teacher introduced well-chosen new ideas that benefited student understanding at relevant moments, such as explaining why weight might not be a useful property in this context: ‘Weight just tells you how much you have, so I didn’t bother with it.’ 5. The teacher summarized important conclusions consistently throughout the discussion, such as reiterating the properties to consider: ‘When we do look at properties, you want to consider everything, because in this chart, something can’t be smooth and rough.’ 6. The teacher connected a series of ideas to the learning goal by discussing the properties and how they help identify the mystery powder, but the connection could have been stronger with a clearer initial statement of the learning goal.

Table 9: Human evaluation and STRUCTURED COT + QUOTES (GPT) output. The green and red texts represent agreement and disagreement between the human and STRUCTURED COT + QUOTES, respectively. The blue texts are quotes from the discussion appearing in both evaluations.

Human 1	<p>The discussion overall was coherent and focused on the learning goal. The T began the discussion by explaining that they will be talking about the fractions they ordered. The T then shares that they will begin by explaining the strategy they used and why it works for any set of fractions. <i>After the students finished sharing their strategies and why they used them, the T then shifted to having the students think about other groups strategies.</i> Once Mina came to the realization that her strategy did not work, the T then transitions the discussion to think about whether the other strategies used could be used with any set of fractions. <i>To end the discussion, the T asks the student what strategy they would use if she gave them the same problem again based on the discussion.</i> By the end the learning goal was achieved. Mina realized that her strategy would not work and the other groups realized their strategy could not be generalized to all fraction sets.</p>
Human 2	<p>The T opens the discussion by saying, <i>“Today we’re going to be talking about ordering fractions and the worksheet that you guys did</i> with three-ninths, nine-tenths, oh three-tenths, nine-tenths and three-fourths. So first I want everyone to go around the room and I will call on you. And you’re going to explain your strategy and why you think it works for any set of fractions. First, we’re going to start off with Will and Jayla. I want you to explain to the group what you did for your strategy.” <i>She then asks each group to share their strategy. Then, she focuses on one strategy at a time and asks others what they think about it.</i> After that (and convincing M she is incorrect), she discusses the answers from one group at a time to the last question. <i>Finally she asks them what they would do “If I were to give you this problem again, after listening to everyone’s strategy and what the limitations were and what they did and why they did it, would you do a, would you use a different strategy for the next time? I want Mina to go first.”</i> So overall, the discussion was pretty organized. However, <i>there were times when the T transitioned to a different idea rather than continuing with a S’s thought</i>; for example, when C brought up that M’s strategy would work if the fractions had the same denominator, the T said, “So now we’re going to jump back to Will and Jayla. So now we’re going to talk about the second question.” Also, <i>the T introduced a new representation (rectangles) instead of using W and J’s number line to convince M her answer was incorrect.</i></p>
GPT	<ol style="list-style-type: none"> 1. The teacher did not state a clear, detailed learning goal and expectation at the beginning. The teacher mentioned the topic but did not articulate the specific learning objectives. For example, the teacher said, <i>‘Today we’re going to be talking about ordering fractions and the worksheet that you guys did,’</i> but did not elaborate on the learning goals. 2. The teacher moved between groups of ideas timely with apparent rationale, ensuring each student had a chance to explain their strategy and then facilitating a discussion on the strengths and limitations of each method. For instance, <i>the teacher asked each group to explain their strategy and then prompted other students to comment on it.</i> 3. The teacher instructed with deliberation, asking follow-up questions to ensure students explained their reasoning. For example, the teacher asked Mina to explain more about her strategy and why she chose it. 4. <i>The teacher introduced well-chosen new ideas that benefited student understanding at relevant moments, such as when the teacher provided a visual explanation to help Mina understand why her method did not work.</i> 5. The teacher did not consistently summarize important conclusions throughout the discussion. While the teacher did clarify points at times, there was no consistent summarization of key takeaways after each strategy discussion. 6. The teacher connected a series of ideas to the learning goal by discussing the limitations and strengths of each strategy, but this connection could have been more explicit. <i>The teacher asked students to reflect on whether they would use a different strategy next time, which helped tie the discussion back to the learning goal.</i>

Table 10: Human evaluation and STRUCTURED CoT + QUOTES (GPT) output. The green and red texts represent agreement and disagreement between the human and STRUCTURED CoT + QUOTES, respectively. The blue texts are quotes from the discussion appearing in both evaluations.

Human	<p>No discussion guidelines or introduction provided. Introduced unproductive or confusing tangents (asked Ss to define numerator and denominator but accepted simplistic description of the top number vs. the bottom number; started drawing an incorrect area model to help M demonstrate her strategy; suggested that E/C use a picture or time line to demonstrate their strategy.) T provided some clear and timely transitions between discussions of different strategies.</p>
GPT	<ol style="list-style-type: none"> 1. The teacher clearly stated the learning goal at the beginning: ‘So today students, we are going to evaluate, justify, compare and contrast strategies for ordering fractions with different numerators and different denominators.’* This sets a clear expectation for the lesson. 2. The teacher moved between ideas timely and with apparent rationale, such as transitioning from discussing Jayla and Will’s number line strategy to practicing with new fractions: ‘I think this might be easier if we practice on some new numbers, okay?’ 3. The teacher instructed with deliberation, ensuring students understood each step before moving on, as seen when the teacher asked students to place dots on the number line and confirmed their understanding: ‘Yes. Good job. So now from this number line, we know that five-eighths is greater than three-eighth, but we still don’t know where one fourth lands.’ 4. The teacher introduced well-chosen new ideas at relevant moments, such as comparing fractions to one-half to determine their relative sizes: ‘The next strategy is Mina’s strategy. Do you want to describe to the class what you did?’ 5. The teacher summarized important conclusions consistently, such as confirming the order of fractions after each exercise: ‘So, now we can see that the correct order three-tenths is the least greatest then it is three-fourths and the greatest is nine-tenths.’ 6. The teacher connected a series of ideas to the learning goal by revisiting different strategies and ensuring students understood the rationale behind each: ‘Now to finish up this discussion, I’m going to give you guys another set of fractions and whatever strategy that you want you can decide what one is the greatest and what one’s the smallest, okay?’ <p>Overall, the discussion was coherent and well-structured, effectively meeting the learning goal.</p>

Table 11: Human evaluation and STRUCTURED CoT + QUOTES (GPT) output. The green and red texts represent agreement and disagreement between the human and STRUCTURED CoT + QUOTES, respectively. The blue texts are quotes from the discussion appearing in both evaluations. *The human rater annotated “So today students, ...” as the evidence for introduction in a separate document.

Human	<p>The teacher provides hardly any introduction at the beginning, but then provides a better introduction at 0:24 after “testing the waters.”*¹ The teacher focuses on addressing Mina’s misconception, but does so with what I think are area models that she drew before the interaction rather than using the number lines, which could do the same thing. The teacher does try to make it clear that the number line strategy is not always a good option, but most of the ideas are her own.</p>
GPT	<ol style="list-style-type: none"> 1. The teacher did not state a clear, detailed learning goal and expectation at the beginning. The teacher mentioned discussing strategies but did not clearly outline the learning goal: ‘we’re going to talk about what strategies are helpful when we are comparing fractions, and what strategies might be better used on a different kind of problem.’ 2. The teacher moved between ideas timely with apparent rationale, focusing on different students’ work and strategies: ‘So we’re going to start our discussion by looking at Will and Jayla’s work with their ordering fractions.’ 3. The teacher instructed with deliberation, asking students to explain their thought processes and addressing misconceptions: ‘If you could just explain to the class, your thought process for putting together the strategy, I think that might be really helpful to the other students in the class.’ 4. The teacher introduced well-chosen new ideas that benefited student understanding at relevant moments, such as using visual aids: ‘I actually drew a picture that maybe we could look at together.’*² 5. The teacher summarized important conclusions consistently throughout the discussion, reinforcing key points: ‘So when we look at this picture, can we see which one’s biggest?’ 6. The teacher connected a series of ideas to the learning goal by discussing different strategies and their effectiveness: ‘All of the strategies we’re going to talk about today do work at some point, but we want to talk about the benefits and maybe some of the things that are a little bit harder about those strategies as well.’

Table 12: Human evaluation and STRUCTURED COT + QUOTES (GPT) output. The green and red texts represent agreement and disagreement between the human and STRUCTURED COT + QUOTES, respectively. The blue texts are quotes from the discussion appearing in both evaluations. *¹at 0:24, the teacher said “we’re going to talk ...” *²the human rater annotated “I actually drew ...” as the evidence for the introduction of new ideas in a separate document.

A Bayesian Approach to Inferring Prerequisite Structures and Topic Difficulty in Language Learning

Anh-Duc Vu,^{†‡} Jue Hou,^{†‡} Anisia Katinskaia,^{†‡} Ching-Fan Sheu,[◇] Roman Yangarber[‡]

[†]Department of Computer Science, University of Helsinki, Finland

[‡]Department of Digital Humanities, University of Helsinki, Finland

[◇] National Cheng Kung University, Taiwan

first.last@helsinki.fi

Abstract

Understanding how linguistic topics are related to each another is essential for designing effective and adaptive second-language (L2) instruction. We present a data-driven framework to model topic dependencies and their difficulty within a L2 learning curriculum. First, we estimate topic difficulty and student ability using a three-parameter Item Response Theory (IRT) model. Second, we construct topic-level knowledge graphs—as directed acyclic graphs (DAGs)—to capture the prerequisite relations among the topics, comparing a threshold-based method with the statistical Grow-Shrink Markov Blanket algorithm. Third, we evaluate the alignment between IRT-inferred topic difficulty and the structure of the graphs using edge-level and global ordering metrics. Finally, we compare the IRT-based estimates of learner ability with assessments of the learners provided by teachers to validate the model’s effectiveness in capturing learner proficiency. Our results show a promising agreement between the inferred graphs, IRT estimates, and human teachers’ assessments, highlighting the framework’s potential to support personalized learning and adaptive curriculum design in intelligent tutoring systems.

1 Introduction

A key goal of Intelligent Tutoring Systems (ITS) is to support *personalized* learning by answering key questions: What does the student know? How are they performing? What should they learn next? Achieving this requires three components: a Domain Model (to represent subject knowledge), the Student Model (to represent learner proficiency), and the Instruction Model (to implement pedagogical strategy). Of these, the domain model is foundational, as it informs both the student assessment and the instructional choices. Prior work has explored domain modeling in many learning domains, such as mathematics (Ritter et al., 2007; Arroyo et al., 2014; Klinkenberg et al., 2011).

Beyond proficiency estimation, recent work emphasizes domain models that offer pedagogical insights—such as relative topic difficulty and efficient or optimal learning paths (Swamy et al., 2022; Cohausz, 2022; Weidlich et al., 2022). These can help teachers adapt instruction and improve learning outcomes. In this paper, we focus on modeling relationships among *topics* in language learning using two approaches: predictive modeling and causal modeling. The causal model aims to provide an interpretable domain structure, while the predictive model offers empirical estimates of learning outcomes.

We collect data from real-world learners in our language learning system, Revita (Katinskaia et al., 2018; Katinskaia and Yangarber, 2018; Katinskaia et al., 2017).¹ In Revita’s learning setting, learners complete exercises related to grammar topics in the target language. These exercises are automatically generated from texts that learners upload themselves or select from a shared library of materials. The exercises are presented in the form of multiple-choice or fill-in-the-blank (“cloze”) questions. Each question is associated with one or more *learning topics*—a.k.a. linguistic constructs (Katinskaia et al., 2023)—and the learner’s answer is graded according to its correctness in terms of each topic. We collaborate with language teachers from several universities and collect real data from language learners.

The main goal of this paper is to explore the domain model—using data from learners of Russian, one of several languages offered by the Revita learning platform—which is based on the Russian topics and their relationships. We highlight the following contributions of this paper:

1. We present a simple causal modeling scheme for the domain model and model topics with a directed acyclic graph (DAG). The nodes

¹revitaai.github.io

in the graph represent topics, and the edges represent the relationships between them.

2. We verify our graph structure with predictive analysis: Bayesian network and hierarchical item response theory (IRT) model.

The paper is organized as follows. In section 2 we outline relevant prior work. Section 3 describes our topic inventory, the process of data collection and performance aggregation by topic. Section 4 describes our approach to build the prerequisite graph structures and the statistical models we use to verify the graph structures. Section 5 shows the experiment results. Section 6 concludes with current directions of research.

2 Related Work

Several approaches for modeling learning have been proposed. We briefly review two types of models: (1) predictive models and (2) causal models.

Predictive models focus on predicting with a set of independent latent variables. When modeling learning, these latent variables refer to the levels of the student’s proficiency on various learning topics. One approach is Item Response Theory (van der Linden and Hambleton, 2013). ITS is not the only application of IRT—it can be applied in many settings, including stress testing, psychological and medical testing, etc. Depending on the application domain, the latent trait can be level of anxiety, neurosis, authoritarian personality, etc. IRT has an information-theoretic basis similar to “Elo” ratings (Elo, 1978). The Elo formulas, originally developed for rating chess players, have been adapted in the context of ITS (Pelánek, 2016; Hou et al., 2019). The language-learning domain is more complex than other domains where IRT is used, since the learning topics to be mastered are relatively much more numerous, and have complex relationships among them.

With the rise of deep learning in recent years, *deep knowledge tracing* (DKT) was proposed (Piech et al., 2015), modeling the state of learner knowledge with a recurrent neural network—RNN (Hochreiter and Schmidhuber, 1997). Researchers have proposed several neural network-based approaches (Zhang et al., 2017; Abdelrahman and Wang, 2019; Su et al., 2018; Liu et al., 2019; Pandey and Srivastava, 2020; Song et al., 2021). The benefit of applying neural networks is that they do not require human-engineered

features; despite the success of deep learning, they suffer from a lack of interpretability (Jiang et al., 2024).

Causal models describe the causal relationships in a system. In our case, we consider the causal relationship to be the *prerequisite* relationship among topics or the learner’s knowledge states. The benefit of using causal models is that they can provide a more directly interpretable representation of the domain knowledge (Jiang et al., 2024). Some causal models describe the domain as a directed acyclic graph (DAG), which provides direct value from the perspective of pedagogy. Researchers have explored the use of causal models in education with Bayesian networks (Pardos and Heffernan, 2010) or Markov Blanket (Jiang et al., 2024).

In the field of education, Knowledge Space Theory (KST) (Doignon and Falmagne, 2012) can also be considered as a special graphical causal model. KST is a mathematical framework for modeling the learner’s knowledge, and represents the learner’s current proficiency as a set of mastered skills, which is referred as a *knowledge state*. Each state contains a subset of the skills in the domain. The student has mastered the domain when she reaches the state containing all skills. KST models not only the learner knowledge, but also learning paths, starting from the empty set toward the full set of topics. Various approaches are used to build a knowledge space, from explicit elicitation of knowledge from human experts to data-driven methods, such as Formal Concept Analysis (FCA) (Ganter and Wille, 2012).

3 Data

This work uses learner data collected in collaboration with language teachers at several universities. The dataset covers university students learning Russian as a second language (L2), whose levels range from A1 to C2 on the CEFR scale (Little, 2007), both as part of their university courses and as independent study. Learners upload texts of personal interest, or, if participating in a university course, practice with texts selected or adapted for them by their teachers. Based on the selected texts, the Revita intelligent language tutoring system automatically generates interactive exercises (Katinskaia et al., 2023).²

Revita supports a variety of exercise types for each language, including grammar, vocabulary, lis-

²revita.helsinki.fi

Topics	Examples
(1) Verb: II conjugation	Мы скоро увидим <u>восход</u> . (We will see the sunrise soon.)
(2) Complex pronoun:	Нам нужно кое о <u>чем</u> поговорить. (We need to talk about something)
(3) Perfective vs. imperfective aspect	Страны <u>согласовали</u> проект о будущих отношениях. (The countries agreed on a draft on future relations.)
(4) Dative subject with predicative adjective, or with impersonal verb	Мне <u>необходимо поговорить</u> с врачом. (I need to talk to a doctor. Literally: [it is] necessary for me to talk to a doctor.)

Table 1: Examples of instances of *topics* found in text (underlined). *Candidates* are words that will be chosen for exercises about the topics (marked in **bold**).

tening comprehension, etc. It also provides continual diagnostic assessment. It assists the learners with contextualized feedback and hints depending on their answers. Exercise creation and hint generation are built upon a linguistically-informed domain model, which drives the personalized selection and generation of exercises based on each learner’s proficiency level. In this study, we focus on learner data from grammar exercises in Russian, which serve as the foundation for modeling topic dependencies and estimating topic difficulty.³

3.1 Data collection

Topics: In this paper, we use the term *topic* to refer to specific language learning targets (also known as “skills” in ITS and education literature)—for example, particular patterns of nominal case usage, verb conjugation classes, syntactic constructions involving negation and tense, etc. These are not simply individual grammatical features, such as *past tense* or *plural number*, but rather combinations that reflect in a meaningful fashion how language is taught and learned. For instance, learners may work on mastering topics such as *past tense of a certain verbal paradigm*, rather than *past tense* in general.

To define these topics, we consulted with experts in language pedagogy and textbooks, to align with real-life instructional goals. Table 1 shows examples of topics and exercises that target them.

Exercises: All exercises are automatically generated by the Revita system, based on authentic texts chosen by the teachers and learners from arbitrary sources. The system creates a number of exercise types; here we focus on fill-in-the-blank (“cloze”) and multiple-choice exercises. In a cloze exercise, the system hides certain words or phrases, and shows the learner a hint—the lemma (dictionary form) of the hidden word or phrase. The learner’s

³All learner data was anonymized prior to analysis in accordance with ethical research requirements and standards.

task is to enter the correct surface forms, based on the context of the cloze. In a multiple-choice exercise, the learner is given several options to choose from, with the options generated automatically.

Learners are allowed multiple attempts for each exercise. When an answer is incorrect, the system provides hints on subsequent attempts to support the learner. These hints *gradually* guide the learner toward the correct answer—starting with general guidance and becoming increasingly specific with each additional attempt.

The exercise sequencing strategy follows a hybrid adaptive design. The system is designed to model the learner’s state to select those exercises that optimally match each learner’s current proficiency—targeting an expected success rate of 50%, to keep the exercises appropriately challenging. This is in keeping with Vygotsky’s theory of the Zone of Proximal Development, which states that for optimal learning, the exercises must not be too difficult too often (to avoid frustrating the learner) and not too easy too often (to avoid boring the learner) (Poehner, 2008). Alternatively, learners can manually select their own study paths using a predefined lesson structure, organized from easier to more difficult topics.

Assignment of credit and penalty: Each exercise is associated with one or more topics. The system evaluates the learner’s response to estimate performance on each topic individually. A response may be correct with respect to some topics but incorrect with respect to others—for instance, a learner might use the correct verb tense but the wrong grammatical person. If the learner answers correctly only after receiving hints, we apply a slight penalty, proportionally distributed across the topics linked to those hints. To assign credit and penalty, the system uses several NLP components, including a morphological analyzer, dependency parser, and rule-based pattern matcher. These tools compare the learner’s response with the correct an-

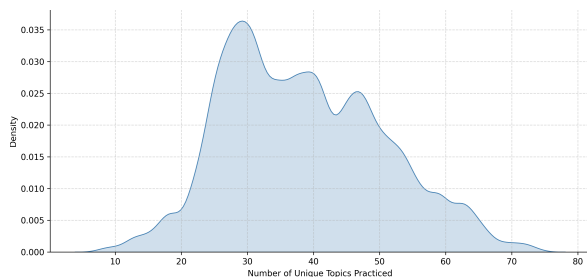


Figure 1: Distribution of the number of unique topics practiced by each student.

swer to determine the topic-level performance for each exercise.

Assignment of credits and penalties is one of the main challenges in our work on assessment. Most statistical approaches, such as IRT, have a clear definition of an *item*, and a clear credit standard—right/wrong answer given by the learner in response to the item. The classic example of an item in IRT is a test question, e.g., in mathematics: it is dichotomous and unambiguous, with a clear judgment of the answer—correct or incorrect. Our major challenge is that our topics are not judged directly, as test items are in other learning domains. It is challenging to determine the credit and penalty for each topic based on the learner’s answer, because the link from exercise to topic is *one-to-many*. This one-to-many nature of the link makes the standard of credit less clear. To tackle this problem, a more sophisticated approach is required to assign credit and penalty. We also face another common problem in language learning and assessment: ambiguity. A substantial proportion of exercises admit *more than one* possibly correct answer, leading to the problem of determining grammatical correctness (Katinskaia and Yangarber, 2021, 2023, 2024). The quality of our NLP components directly impacts the accuracy of the assessment, and therefore the quality of our learning data.

3.2 Data pre-processing

We have collected over 470K student exercise attempts, each with credit and penalty assigned. These exercises were completed by 1,639 unique students. These exercises span over 200 detailed grammatical constructs (Katinskaia et al., 2023), which we group into a smaller set of learning topics that align with pedagogical learning targets, as described above. From this, we derive over 80 distinct topics to be used for modeling and construction of prerequisite graphs.

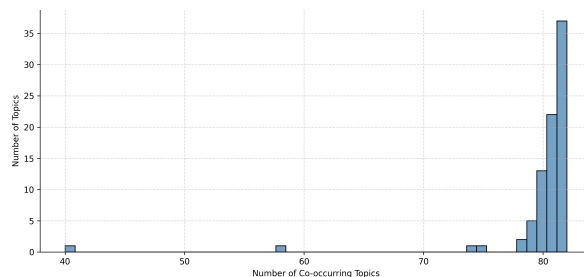


Figure 2: Distribution of counts of co-practiced topics, based on shared student activity. Each value on the X-axis indicates with how many other topics each topic was co-practiced. The Y-axis shows how many topics have the given co-practiced count.

Each exercise is associated with one or more linguistic topics. To enable topic-level analysis, we “explode” (i.e., multiply out) the data, so that each exercise attempt is represented *multiple* times—once per each topic linked to the exercise. This allows us to track student performance separately for each topic. The number of “exploded” data points—pair-wise records linking between student and topic—is approximately 990K.

The histogram in Figure 1 shows the distribution of unique topics practiced per student. Most students engage with 25 to 50 distinct topics, with a concentration around 30. Since learners tend to focus on topics appropriate to their proficiency level, we expect considerable overlap in practiced topics among students of similar levels. This local overlap is useful for constructing prerequisite graphs, as it provides aligned performance patterns across comparable learners without requiring complete topic coverage by each individual.

We next check what topics are *co-practiced* with other topics—i.e., which topics have been practiced together with other topics by at least one student. Figure 2 shows how many topics are co-practiced with other topics. In fact, most topics are co-practiced with 80 or more other topics, indicating a highly interconnected curriculum, where students tend to practice multiple topic combinations. This highlights the dense overlap in student exposure across topics, which is a useful signal for data-driven construction of dependency graphs.

Figure 3 shows the distribution of students that have engaged with each topic. While some topics are widely practiced by hundreds of learners, others are encountered by only a few students, indicating potential variation in topic popularity, curriculum coverage, or personalized learning paths. This vari-

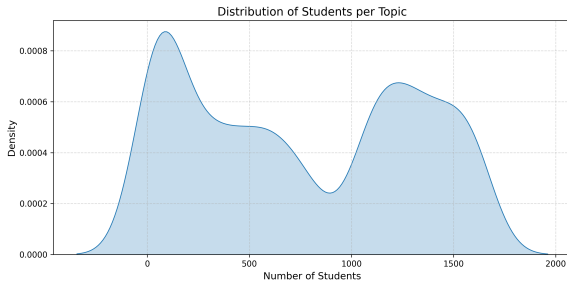


Figure 3: Distribution of the number of students per topic.

ability may impact both topic-level estimation and the structure of the prerequisite graph.

We have also collected data from over 50 students who completed 100 or more exercises each, and who have teacher-assigned CEFR levels. This subset provides a valuable reference for evaluating student ability and topic difficulty. Overall, the dataset’s size and structure support both robust probabilistic modeling of learner proficiency and detailed analysis of topic interdependencies.

4 Methodology

4.1 Prerequisite graph construction

To represent the prerequisite structure of the topics, we construct a DAG (Chickering, 2002) over learning topics, where each node represents a topic. Directed edges in this graph indicate prerequisite relationships inferred from empirical student performance patterns. Specifically, a directed edge from topic A to topic B , denoted $A \rightarrow B$, means that mastery of topic A is likely a prerequisite for success on topic B .

We explore multiple methods for constructing the prerequisite graph. The first is a threshold-based approach, in which a directed edge $A \rightarrow B$ is added if a statistically significant fraction of students consistently perform better on topic A than *the same students* perform on topic B . This approach focuses solely on relative performance outcomes across topics. By aggregating student-specific accuracy rates, the method infers likely learning dependencies under the assumption that prerequisite topics are easier for students to master than their dependents.

The second method is Grow-Shrink Markov Blanket (GS-MB) approach to learn topic dependencies based on statistical conditional independence tests (Margaritis and Thrun, 2000). We first identify potential neighbors of a target topic by

evaluating unconditional correlations (grow phase), then we remove far neighbors by testing for conditional independence given the remaining set (shrink phase) until reaching the actual Markov blanket of the topic. The resulting undirected dependencies are then converted into directed edges using edge orientation heuristics.

To ensure that the resulting prerequisite graph is a valid directed acyclic graph (DAG), we apply data-driven postprocessing to eliminate cycles and resolve bidirectional edges. If a cycle is detected, we iteratively remove the weakest edge within the cycle—where “weakness” is determined using statistical evidence such as a low agreement ratio or minimal co-occurrence frequency across student performance data. Unlike the traditional Grow-Shrink approach proposed by Margaritis and Thrun (2000), which attempts to reverse and reinsert removed edges followed by directional propagation heuristics, our method permanently removes low-confidence edges without reorientation. This simplification focuses on preserving only the most statistically supported links while enforcing global acyclicity. For bidirectional dependencies (i.e., both $A \rightarrow B$ and $B \rightarrow A$), we retain only the edge with the stronger statistical support, ensuring a consistent and interpretable prerequisite structure.

4.2 IRT Modeling of Student Performance

We use a probabilistic model to estimate student ability and topic difficulty based on their exercise-performance data. Specifically, we apply the three-parameter logistic (3PL) Item Response Theory (IRT) model (Baker, 2001). In 3PL, each student has an ability parameter θ_s , and each topic has two parameters: difficulty β_t , and discrimination α_t . We also take into account the factor of luck as guessing parameter g . The probability that a student s answers topic t correctly is modeled as:

$$c_{u,t} \sim \text{Bernoulli}(g + (1 - g) \cdot \sigma(\alpha_t(\theta_s - \beta_t)))$$

where $\sigma(\cdot)$ is the sigmoid function.

We assume a fixed guessing parameter $g = 0.01$ for cloze-style exercises, which approximates the probability of answering correctly by chance. For multiple-choice exercises, g is determined dynamically based on the number of answer options.

To estimate the posterior distributions of the model parameters, we perform fully Bayesian inference via Markov Chain Monte Carlo (MCMC) (Gilks et al., 1995), using the No-U-Turn Sampler

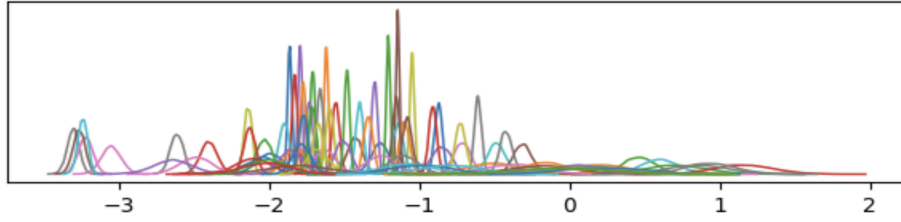


Figure 4: Posterior distributions of topic difficulty (β) estimated by the IRT 3PL model. X-axis is estimated topic difficulty. Each curve represents the density for a different topic.

(NUTS) (Hoffman and Gelman, 2014) as implemented in PyMC (Patil et al., 2010; Salvatier et al., 2016).

NUTS is a gradient-based sampling algorithm that extends Hamiltonian Monte Carlo (HMC) by adaptively deciding how many steps to simulate on each iteration, based on the gradients of the log-posterior. It dynamically simulates forward and backward trajectories in the parameter space, stopping when a “U-turn” is detected, and then selects a new sample from the visited states using a probability distribution. The posterior samples of β and α for each topic capture both the parameter estimates (mean) and their associated *uncertainty* (standard deviation), enabling more detailed downstream analysis and validation of the graph structure.

4.3 Comparing Graph Structure with IRT Difficulty

We assess the extent to which the structure of the prerequisite graph agrees with IRT-inferred topic difficulty. Intuitively, if topic A is a prerequisite for topic B , then A should be easier (i.e., have lower β) than B . To evaluate this alignment, we use three complementary metrics.

Edge Agreement Score (EAS) measures the proportion of edges in the graph that follow the expected difficulty order. For each edge $A \rightarrow B$, we check whether $\beta_A < \beta_B$. The EAS is calculated as the fraction of such edges over all edges in the graph. A perfect score of 1.0 indicates that *all* edges point from an easier to a harder topic.

Weighted Direction Score (WDS) refines this idea by incorporating the size of the difficulty gap. Rather than using a hard threshold, we score each edge $A \rightarrow B$ using a sigmoid-transformed difference between the difficulties of topics A and B :

$$\sigma(\beta_A, \beta_B) = 1/(1 + e^{-(\beta_B - \beta_A)})$$

This yields higher scores when β_B is much

greater than β_A , and values near 0.5 when the difference is small or uncertain. WDS offers a smoother estimate that rewards clear hierarchical structure.

Kendall’s Tau, originally introduced by Kendall (1938), is designed to measure the ordinal association between two ranked variables. We use it to compare the global ordering of topics implied by the graph with the ranking induced by the IRT-inferred difficulty estimates. This is done by computing a topological sort of the graph to obtain a linear topic ordering, which is then correlated with IRT’s β values using Kendall’s Tau. A high Tau value indicates strong agreement: topics that appear earlier in the graph tend to be easier than those ranked later.

Together, these metrics offer both local and global perspectives on how well the learned DAG structure matches the IRT-inferred difficulty landscape.

5 Experiments and Results

5.1 IRT Estimations

Figure 4 shows the posterior density of all topic difficulty estimates. The IRT model estimates topic difficulty values ranging from -3.31 to 1.16, giving a total range of 4.47 on the X-axis. Of 83 topics, 38 have standard deviations below 0.05, and 55 are below 0.10, meaning that their difficulty estimates are quite stable. For 95% confidence intervals (CI), 50 topics (~60% of all topics) have interval widths under 0.30, which is about 6.7% of the full difficulty range. For 17 topics (20% of all topics), the estimated CI width is under 0.10—only 2.2% of the full range. These statistics suggest that many topics are estimated with high confidence, and they are reliable enough to be used for comparison with the topic graph.

Figure 5 illustrates the relationship between uncertainty in topic difficulty (standard deviation of β) and topic discrimination (mean of α). Topics with higher discrimination α tend to show lower

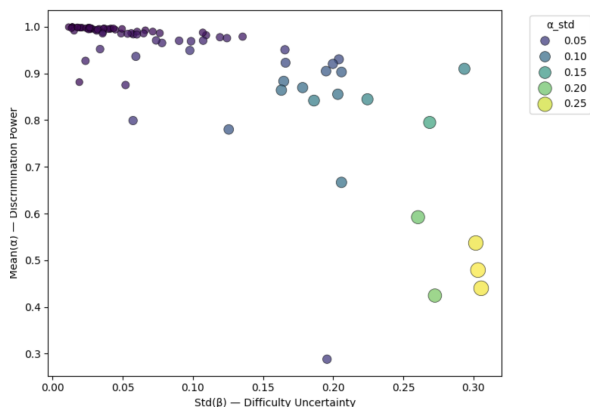


Figure 5: Correlation between uncertainty in the difficulty of a topic— $\text{std}(\beta)$ —and discrimination parameter of the topic— $\text{mean } \alpha$ —for each topic. Each point represents a topic. Marker color and size indicate the uncertainty in α .

uncertainty in their estimated difficulty, indicating more stable and informative estimates. In contrast, topics with lower discrimination show greater uncertainty in β , as well as higher variability in α , as indicated by larger, lighter-colored markers. This pattern suggests that strongly discriminative topics provide more reliable signals for modeling.

A detailed heatmap of the correctness of the student responses by topic and student ability quantiles (Q1 = lowest, Q5 = highest) is shown in Appendix Figure 9 and 10. In the heatmap, topics are ordered by their IRT-estimated mean difficulty β . The color of each cell shows the average correctness rate, and the number of student-topic interactions. As expected, higher-ability students (Q4–Q5) perform better, particularly on the more difficult topics, reinforcing the validity of the estimated difficulty scores.

Figure 6 shows how the estimates of student ability θ vary across CEFR levels assigned by the teachers. The correlation between CEFR grade and IRT-estimated ability is moderate, with a Spearman coefficient of $r = 0.473$, indicating that as CEFR level increases, IRT-based ability estimates also tend to rise.

Figure 7 shows the relationship between the number of exercises completed by each student and the uncertainty in their estimated ability, measured as the posterior standard deviation of θ . There is a strong negative correlation ($r = -0.758$, $p < 0.001$), indicating that students who complete more exercises tend to have more confident (lower-variance) ability estimates. This supports the intuitive notion that additional observations reduce

posterior uncertainty in the IRT model.

Figure 8 shows a strong negative relationship between the number of students who practiced a topic and the uncertainty in that topic’s IRT difficulty estimate. Topics attempted by more students tend to have significantly lower standard deviation in their β values, suggesting higher confidence in the estimated difficulty. This trend is quantitatively supported by a Spearman correlation of $r = -0.899$ ($p < 0.001$), confirming that broader student coverage leads to more stable parameter estimates.

5.2 Graph construction

We construct two types of topic prerequisite graphs to capture learning dependencies. The first, a threshold-based graph, connects topics where a consistent performance advantage suggests one precedes the other. The second, built using the Grow-Shrink Markov Blanket algorithm, identifies conditional dependencies between topics based on statistical independence tests.

The threshold-based graph includes 83 nodes and 173 edges, resulting in a wide and dense structure with many inferred prerequisite links. In contrast, the GS-MB graph is sparser, with 80 nodes and 86 edges, forming a deeper and narrower hierarchy. Both graphs are processed to remove cycles and bidirectional edges, ensuring they are valid directed acyclic graphs (DAGs). Visualizations of both graphs can be found in the Appendix (Figures 11 and 12).⁴

Both graphs offer useful perspectives. When we manually examine their qualitative plausibility from a linguistic standpoint, we find that the threshold-based graph often aligns more intuitively with expected topic relationships in Russian, suggesting that threshold-based edges may capture pedagogically meaningful dependencies more effectively than the GS-MB structure. We will explore this in further depth in future work.

5.3 Graph vs. IRT estimations

Two approaches are evaluated for constructing topic prerequisite graphs: a threshold-based method and the Grow-Shrink Markov Blanket algorithm. Both produce DAGs, which are evaluated for alignment with the IRT-inferred topic difficulties using three metrics: Edge Agreement Score (EAS),

⁴Both of these graphs are too large to fit into the paper; please see the complete graph of threshold-based approach [here](#) and GS-MS approach [here](#).

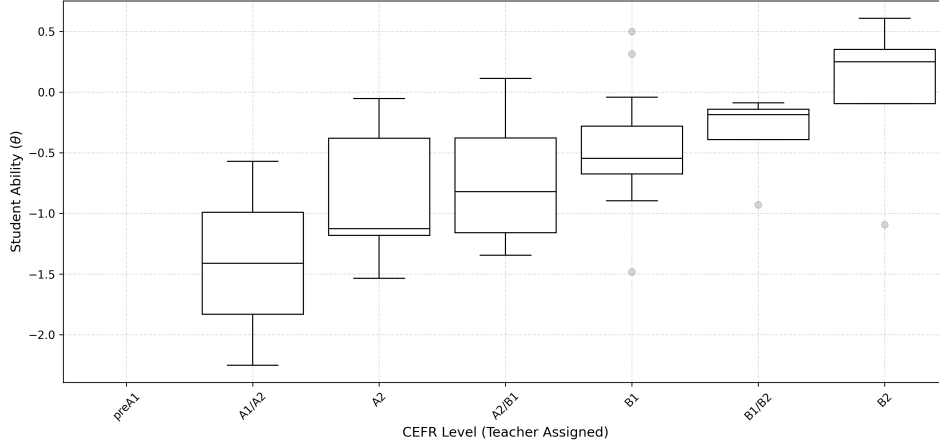


Figure 6: IRT ability estimates θ across teacher-assigned CEFR levels. Boxplot shows distribution of student abilities per CEFR level.

Graph	Uncertainty Filter	EAS	WDS	Kendall's Tau
Threshold-based (G_{dag1})	$\sigma_\beta < 0.05$	1.000	0.869	0.240
	$\sigma_\beta < 0.10$	0.604	0.609	0.220
GS-Markov Blanket (G_{dag2})	$\sigma_\beta < 0.05$	0.523	0.514	0.050
	$\sigma_\beta < 0.10$	0.505	0.502	0.014

Table 2: Agreement between graph structure and IRT-estimated topic difficulty. EAS: Edge Agreement Score. WDS: Weighted Direction Score. Kendall's Tau compares topological sort with IRT difficulty rank.

Weighted Direction Score (WDS), and Kendall's Tau.

Table 2 summarizes the results under two topic uncertainty thresholds— $\sigma_\beta < 0.05$ and $\sigma_\beta < 0.1$ —which correspond to subsets of 30 and 50 topics, respectively. The threshold-based graph consistently shows stronger alignment with IRT difficulty estimates, achieving perfect edge agreement (EAS = 1.00), high directional consistency (WDS = 0.869), and a moderate Kendall's Tau of 0.240 under stricter filtering. Even with relaxed thresholds, it maintains relatively strong scores across all three metrics. In contrast, the GS-MB graph produces lower EAS, WDS, and notably near-zero Kendall's Tau values (i.e., 0.050 and 0.014), indicating that its topological structure does not match the global difficulty ranking well.

These results suggest that while GS-MB could be effective at capturing local conditional dependencies, it falls short in representing an overall difficulty hierarchy—a strength more consistently captured by the threshold-based method.

6 Conclusion

In this work, we present a unified framework for modeling topic difficulty and learning dependencies in second-language acquisition, leveraging

large real-world learner data from thousands of students. Using probabilistic modeling and graph-based structure learning, we analyze over 470K student exercise attempts spanning more than 80 topics. Our aim is twofold: (1) to estimate topic-level difficulty and learner ability using a Bayesian IRT model, and (2) to construct interpretable prerequisite graphs that reveal topic hierarchies potentially useful for improving learning.

We compare two graph construction methods: a threshold-based approach that aggregates relative performance gaps across students, and a Grow-Shrink Markov Blanket (GS-MB) method based on statistical conditional independence tests. Three evaluations using Edge Agreement Score (EAS), Weighted Direction Score (WDS), and Kendall's Tau show that the threshold-based method aligns more closely with the IRT-inferred topic difficulties. This supports the hypothesis that prerequisite topics tend to be easier than their dependents, and suggests that simple, data-driven heuristics can reveal meaningful pedagogical structures.

Our findings also demonstrate that model confidence is strongly influenced by the *volume* and *diversity* of learner data. Students who have completed more exercises tend to have lower uncertainty in their ability estimates; topics practiced by

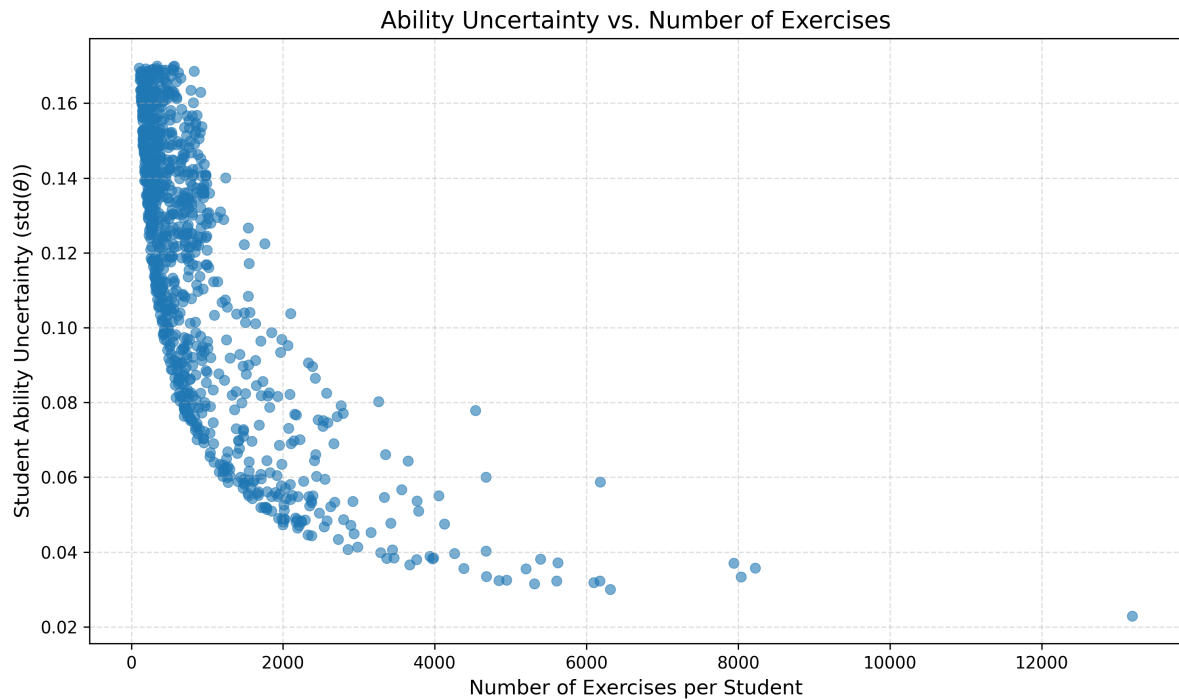


Figure 7: Relationship between number of exercises completed and ability uncertainty (standard deviation of θ). Each point represents a student.

more learners show lower variability in their difficulty estimates. These patterns highlight the value of large-scale learner data in stabilizing the parameter estimates and guiding curriculum analysis.

Moreover, the estimated IRT abilities exhibit a correlation with teacher-assigned CEFR levels, providing external validation for the model and supporting its use in real-world learner assessment. We further explore several aggregate statistics, including topic-topic co-occurrence and student-topic interaction distributions, to explore coverage patterns and the implications for curriculum design.

In summary, this study contributes a robust methodology for combining statistical modeling and graph structure learning in an educational setting. The approach offers practical tools for curriculum designers and language educators to identify learning gaps, and to evaluate learner proficiency. In future work, we will explore extending the model to dynamic learning sequences, fine-grained topic representations, and multilingual adaptation, to further enhance intelligent language tutoring systems.

Limitations

Our results at present have several limitations that may affect the generalizability and precision of the

results.

The dataset primarily consists of learners at the A2, B1, and B2 levels, with relatively few samples from C-level students and very limited representation of pre-A1 and A1 learners. As a result, the inferred difficulty hierarchy and student ability estimates may not fully reflect the learning needs or patterns of beginners and advanced learners.

The distribution of labeled performance data is imbalanced: 78.4% of responses are correct, while only 21.6% are incorrect. This skew may reduce the model’s sensitivity to detecting subtle topic-level challenges, and can introduce bias in estimating both the topic difficulty and discrimination parameters.

Addressing these gaps—through more diverse learner sampling and more balanced task evaluation—would improve the robustness of future modeling efforts.

References

- Ghodai Abdelrahman and Qing Wang, 2019. Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184.
- Ivon Arroyo, Beverly Park Woolf, Winslow Burelson,

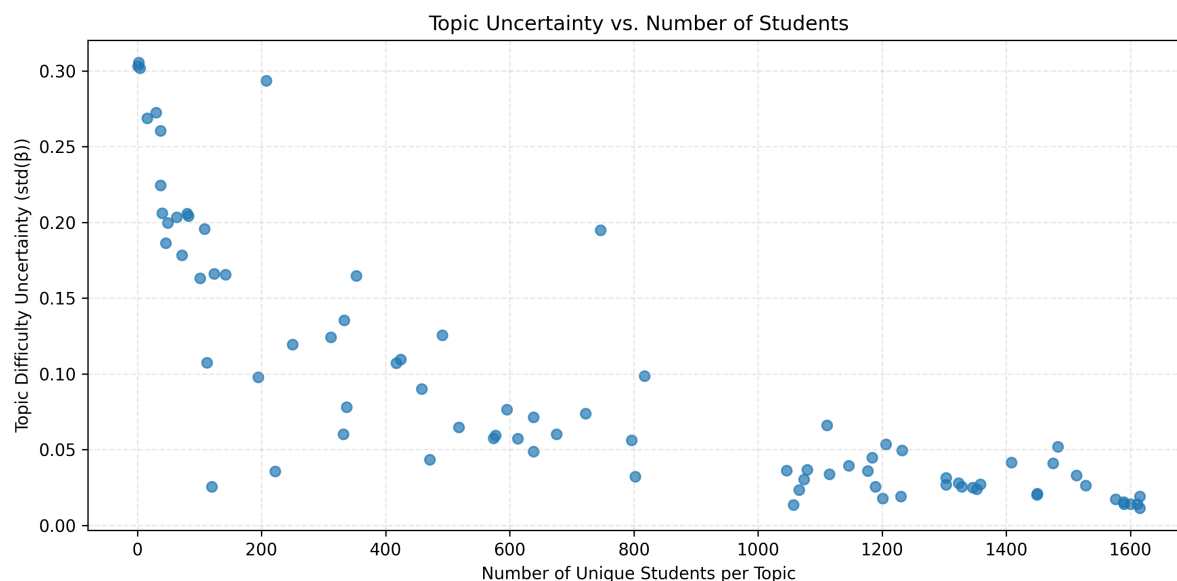


Figure 8: Topic difficulty uncertainty ($\text{std}(\beta)$) vs. number of unique students per topic. Each point represents a topic.

- Kasia Muldner, Dovan Rai, and Minghui Tai. 2014. [A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect](#). *International Journal of Artificial Intelligence in Education*, 24(4):387–426.
- Frank B. Baker. 2001. *The Basics of Item Response Theory*. ERIC Clearinghouse on Assessment and Evaluation.
- David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554.
- Lea Cohausz. 2022. Towards real interpretability of student success prediction combining methods of xai and social science. *International Educational Data Mining Society*.
- Jean-Paul Doignon and Jean-Claude Falmagne. 2012. *Knowledge spaces*. Springer Science & Business Media, New York, NY.
- Arpad E. Elo. 1978. *The rating of chessplayers, past and present*. Arco Pub., New York.
- Bernhard Ganter and Rudolf Wille. 2012. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, New York, NY.
- Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. 1995. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Matthew D Hoffman and Andrew Gelman. 2014. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. In *Journal of Machine Learning Research*, volume 15, pages 1593–1623.
- Jue Hou, Maximilian W Koppatz, José Maria Hoya Quecedo, Nataliya Stoyanova, Mikhail Kopotev, and Roman Yangarber. 2019. Modeling language learning using specialized Elo ratings. In *BEA: 14th Workshop on Innovative Use of NLP for Building Educational Applications, ACL: 56th annual meeting of Association for Computational Linguistics*.
- Bo Jiang, Yuang Wei, Ting Zhang, and Wei Zhang. 2024. [Improving the performance and explainability of knowledge tracing via markov blanket](#). *Information Processing and Management*, 61(3):103620.
- Anisia Katinskaia, Jue Hou, Anh-Duc Vu, and Roman Yangarber. 2023. [Linguistic constructs represent the domain model in intelligent language tutoring](#). In *EACL: 17th Conference of European Chapter of Association for Computational Linguistics*.
- Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2017. Revita: a system for language learning and supporting endangered languages. In *6th Workshop on NLP for CALL and 2nd Workshop on NLP for Research on Language Acquisition, at NoDaLiDa, Gothenburg, Sweden*.
- Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2018. Revita: a language-learning platform at the intersection of ITS and CALL. In *Proceedings of LREC: 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan*.
- Anisia Katinskaia and Roman Yangarber. 2018. Digital cultural heritage and revitalization of endangered Finno-Ugric languages. In *Proceedings of the 3rd Conference on Digital Humanities in the Nordic Countries, Helsinki, Finland*.

- 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.
- Anisia Katinskaia and Roman Yangarber. 2023. Grammatical error correction for sentence-level assessment in language learning. In *18th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 488–502.
- Anisia Katinskaia and Roman Yangarber. 2024. Gpt-3.5 for grammatical error correction. In *Proceedings of COLING-LREC: Joint International Conference on Computational Linguistics and Language Resources and Evaluation*.
- Maurice G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.
- Sharon Klinkenberg, Marthe Straatemeier, and Han LJ van der Maas. 2011. Computer adaptive practice of maths ability using a new item response model for on-the-fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824.
- David Little. 2007. The common European framework of reference for languages: Perspectives on the making of supranational language education policy. *The Modern Language Journal*, 91(4):645–655.
- Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115.
- Dimitris Margaritis and Sebastian Thrun. 2000. [Bayesian network induction via local neighborhoods](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 12.
- Shalini Pandey and Jaideep Srivastava. 2020. Rkt: Relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1205–1214.
- Zachary A. Pardos and Neil T. Heffernan. 2010. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Anand Patil, David Huard, and Christopher J Fonesbeck. 2010. Pymc: Bayesian stochastic modelling in python. *Journal of Statistical Software*, 35(4):1–81.
- Radek Pelánek. 2016. Applications of the Elo rating system in adaptive educational systems. *Computers & Education*, 98:169–179.
- 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.
- Matthew E Poehner. 2008. *Dynamic assessment: A Vygotskian approach to understanding and promoting L2 development*, volume 9. Springer Science & Business Media, New York, NY.
- Steven Ritter, John R. Anderson, Kenneth R. Koedinger, and Albert Corbett. 2007. [Cognitive tutor: Applied research in mathematics education](#). *Psychonomic Bulletin & Review*, 14(2):249–255.
- John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. 2016. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55.
- Xiangyu Song, Jianxin Li, Yifu Tang, Taige Zhao, Yunliang Chen, and Ziyu Guan. 2021. Jkt: A joint graph convolutional network based deep knowledge tracing. *Information Sciences*, 580:510–523.
- Yu Su, Qingwen Liu, Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Chris Ding, Si Wei, and Guoping Hu. 2018. Exercise-enhanced sequential modeling for student performance prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Vinitra Swamy, Bahar Radmehr, Natasa Krco, Mirko Marras, and Tanja Käser. 2022. [Evaluating the explainers: Black-box explainable machine learning for student success prediction in moocs](#). In *EDM*.
- Wim J van der Linden and Ronald K Hambleton. 2013. *Handbook of modern item response theory*. Springer Science & Business Media, New York, NY.
- Joshua Weidlich, Dragan Gašević, and Hendrik Drachler. 2022. Causal inference and bias in learning analytics: A primer on pitfalls using directed acyclic graphs. *Journal of Learning Analytics*, 9(3):183–199.
- Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774.

A Appendix

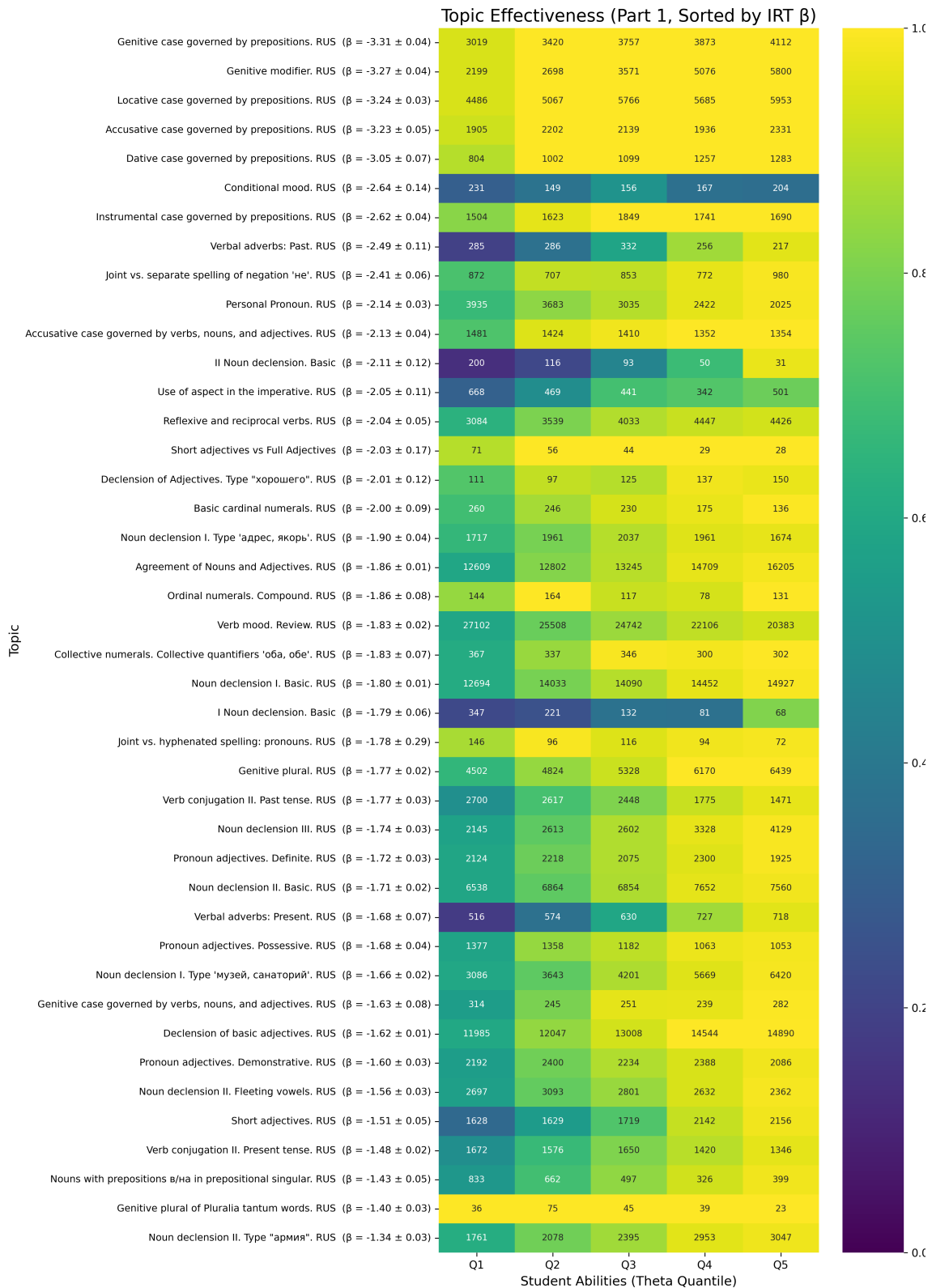


Figure 9: Heatmap of *rate of correct answers* per topic (Part 1). Correctness rates shown per topic and student ability quantile (Q1 = lowest ability, Q5 = highest). Color shows average correctness rate. Number in box indicates support: the number of student-topic interactions. Topics are ordered by their IRT-estimated difficulty β .

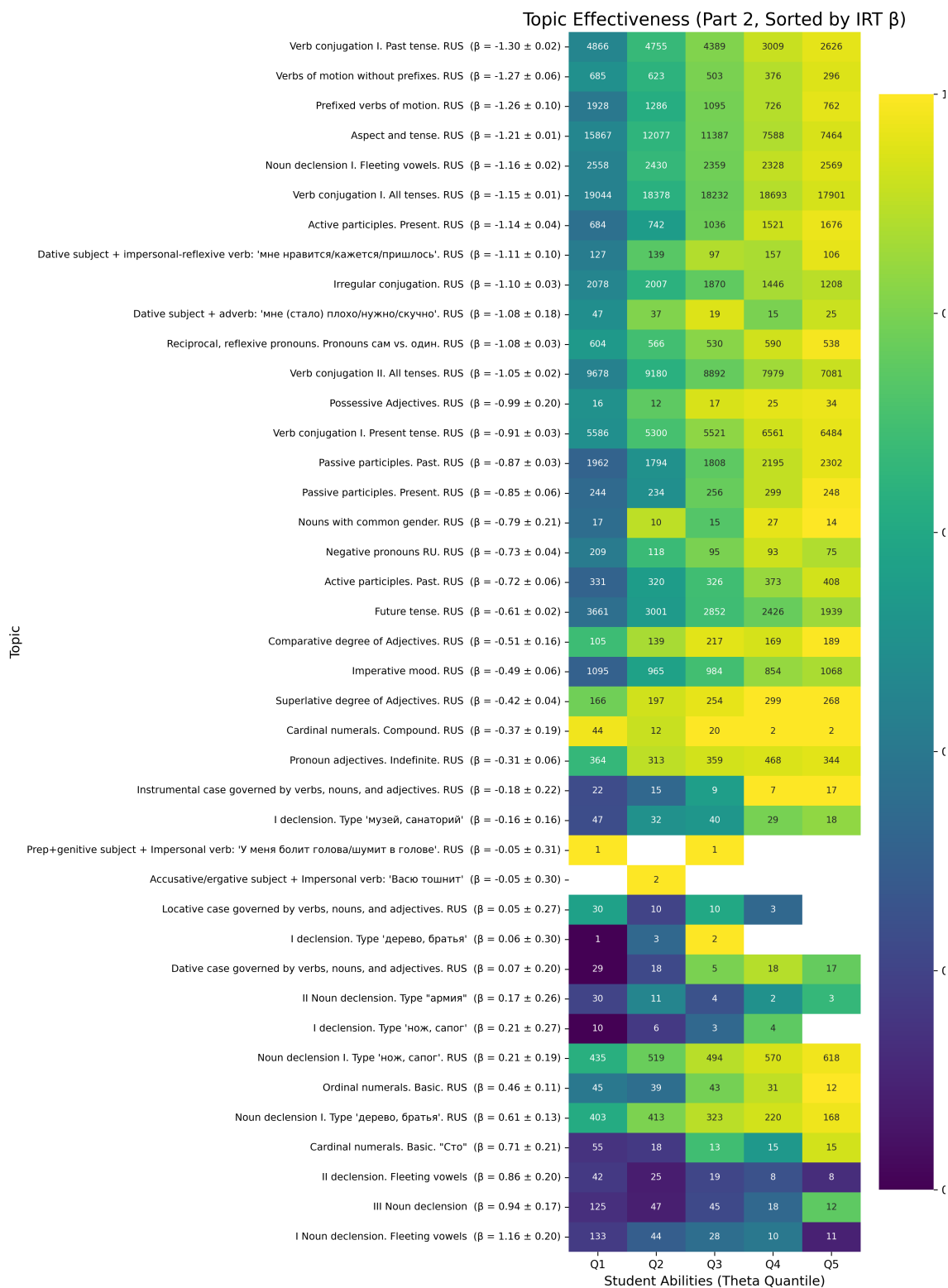


Figure 10: Heatmap of correctness per topic (Part 2); continuation of the heatmap showing remaining topics.

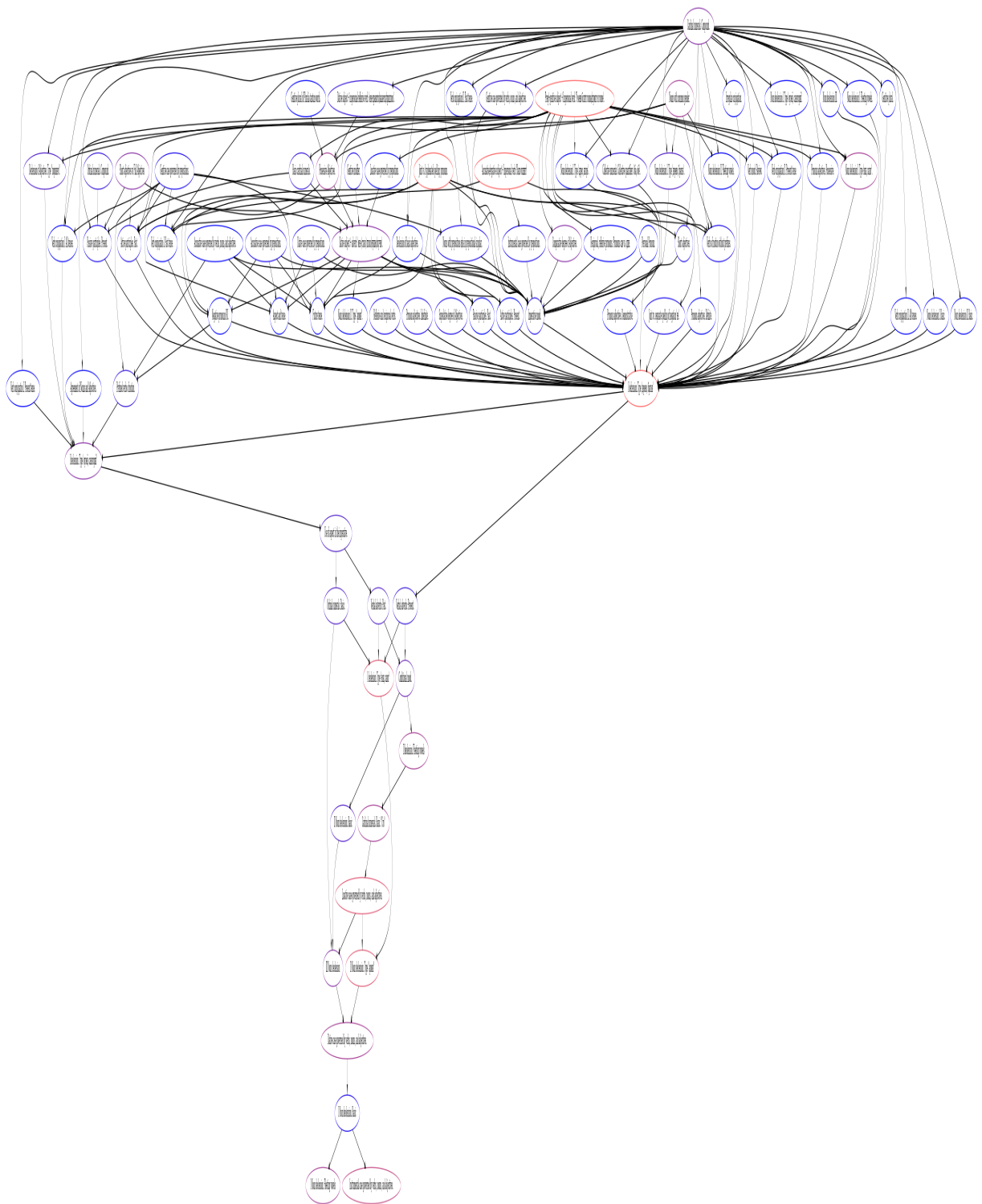


Figure 11: Prerequisite graph constructed using the threshold-based method.

Improving In-context Learning Example Retrieval for Classroom Discussion Assessment with Re-ranking and Label Ratio Regulation

Nhat Tran

Diane Litman

Benjamin Pierce

Richard Correnti

Lindsay Clare Matsumura

University of Pittsburgh

Pittsburgh, PA, USA

{nlt26, dlitman, bep51, rcorrent, lclare}@pitt.edu

Abstract

Recent advancements in natural language processing, particularly large language models (LLMs), are making the automated evaluation of classroom discussions more achievable. In this work, we propose a method to improve the performance of LLMs on classroom discussion quality assessment by utilizing in-context learning (ICL) example retrieval. Specifically, we leverage example re-ranking and label ratio regulation, which forces a specific ratio of different types of examples on the ICL examples. While a standard ICL example retrieval approach shows inferior performance compared to using a predetermined set of examples, our approach improves performance in all tested dimensions. We also conducted experiments to examine the ineffectiveness of the generic ICL example retrieval approach and found that the lack of positive and hard negative examples can be a potential cause. Our analyses emphasize the importance of maintaining a balanced distribution of classes (positive, non-hard negative, and hard negative examples) in creating a good set of ICL examples, especially when we can utilize educational knowledge to identify instances of hard negative examples.

1 Introduction

The automatic evaluation of classroom discussion quality has emerged as a significant area of interest within educational research. A wide range of studies have established that the quality of classroom discourse plays a pivotal role in facilitating student learning and cognitive development (Desimone and and, 2017; Wilkinson et al., 2015; Suresh et al., 2019; Jacobs et al., 2022). Nevertheless, large-scale assessment of classroom discussions remains prohibitively resource-intensive and logistically challenging. The development of automated scoring systems offers a promising solution, enabling the generation of extensive datasets to investigate the mechanisms through which discourse

shapes student reasoning and understanding. Furthermore, such systems hold the potential for integration into formative assessment practices, providing educators with actionable feedback to enhance the effectiveness of classroom discussions.

Compared to pre-trained language models (PLMs) such as BERT (Devlin et al., 2019), large language models (LLMs) have been shown to be more reliable in scoring different dimensions of classroom discussion quality, based on the *Instructional Quality Assessment (IQA)* (Tran et al., 2024a). Prior LLM approaches for classroom discussion assessment have ranged from using zero-shot prompts (Wang and Demszky, 2023; Whitehill and LoCasale-Crouch, 2024) that do not exploit the few-shot learning capability of LLMs (Brown et al., 2020), to utilizing few-shot prompts but with a fixed set of examples for every input (Tran et al., 2024a,b). Inspired by the advancement of in-context learning (ICL) *example retrieval* (Wang et al., 2024; Zhang et al., 2023), we attempt to automatically select few-shot examples based on a given input.

Our work thus aims to improve the automated scoring of classroom discussion quality with ICL example retrieval. Utilizing LLMs for binary prediction with a ‘target’ label (e.g., if we are identifying if a label y is present in the current turn, the target now is y), we define the types of examples as follows. If an example has the same label as the target label, it is a *positive* example, otherwise, it is a *negative* example. A *hard negative* example is a negative example that we expect will be difficult for a model to distinguish from positive examples, i.e., positive and hard negative examples are semantically similar in the input space but represent different classes in the output space. From a retrieval perspective, the hard negative examples are often selected based on some quantitative metrics such as their distance in the embedding space or their ranking from a reward model (Wang et al.,

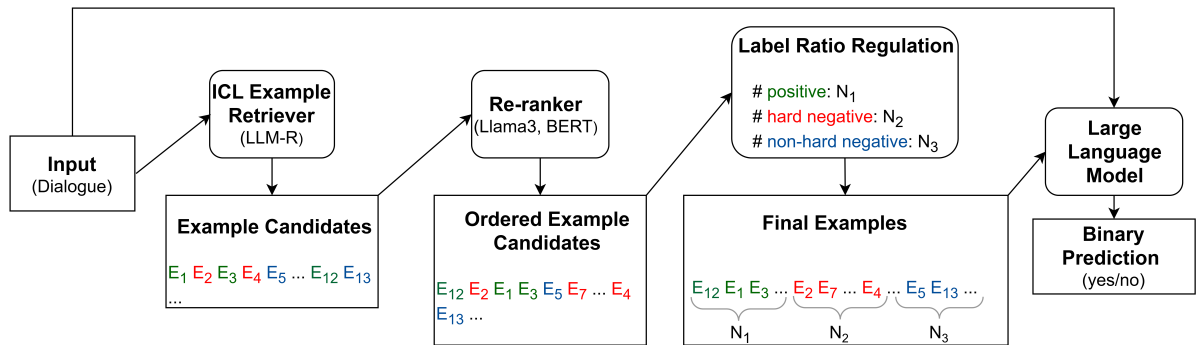


Figure 1: Overview of the proposed method.

2024; Zhang et al., 2023). However, in the context of classroom discussion, since we can leverage a qualitative metric (i.e., domain knowledge about the definition of the labels), we can identify hard negative examples for the target label more reliably. Specifically, based on the definitions of the labels, for a target label A, we know that label A' is closer to A compared to other labels from a human perspective. Therefore, when finding hard negative examples for A, we can quickly select instances with label A' as candidates without needing to calculate any kind of 'distance' between them.

After experimenting with a generic ICL example retrieval approach (LLM-R by Wang et al., 2024), we found it is ineffective for classroom discussion. We hypothesized that the problem is from 1) the imbalance of positive/negative examples and 2) the lack of hard negative examples, as we cannot control the retrieval process. The first hypothesis is well-known in ICL learning as we need both positive and negative examples to learn effectively (Min et al., 2022). The second hypothesis is from the observation that hard negative examples play a crucial role in getting good prediction performance (Tran et al., 2024a; Robinson et al., 2021). Moreover, although we have domain knowledge about hard negative examples based on the annotations of the labels, the ICL retriever only relies on quantitative metrics (e.g., higher-ranking examples) to identify them. To address these issues, we proposed a 2-step approach. First, we train a BERT-based re-ranker to re-order the retrieved examples from LLM-R (Wang et al., 2024). Second, we employ label ratio regulation (LRR), which selects examples from the sorted list while maintaining a specific ratio of positive, non-hard negative, and hard negative examples in the 10-example set used in the prompt (Figure 1).

Our goal is to answer these research questions:

- RQ_1 Does the proposed method help improve performance?
- RQ_2 Does ICL example retrieval have good coverage of the label space (type of examples)?
- RQ_3 How does the ratio of the ICL examples used in the label ratio regulation influence the performance?

Our contributions are three-fold. First, we show that a standard ICL example retrieval approach, despite being useful for other natural language processing (NLP) tasks, is ineffective for classroom discussion assessment. Further analyses suggest that the lack of positive examples and hard negative examples can be causes for this poor performance. Second, we propose an approach utilizing re-ranking and label ratio regulation to complement the standard ICL example retrieval. It helps improve performance and yields comparable results to a finetuned retriever without finetuning the retriever. Third, we demonstrate that even with re-ranking, the retrieval process fails to effectively select hard negative examples, which emphasizes the importance of label ratio regulation when the domain knowledge of the classes (e.g., which class is a hard negative example) is available.

2 Related Work

2.1 LLMs for Classroom Discussion Scoring

As generative LLMs such as GPT-4 (OpenAI et al., 2024), Llama (Grattafiori et al., 2024), and Mistral (Jiang et al., 2023) have outperformed PLMs in many NLP tasks, there has been growing interest in leveraging these LLMs for classroom discussions.

When predicting accountable talk moves in classroom discussions, a finetuned LLM such as GPT-3 has consistently surpassed RoBERTa (Liu et al., 2019) in precision (Kupor et al., 2023). However,

since finetuning LLMs requires significant expertise, extensive data, and substantial computational resources, researchers have increasingly focused on zero-shot and few-shot approaches that do not require additional training. For instance, one study examined the zero-shot capabilities of ChatGPT in three tasks: scoring transcript segments using classroom observation instruments, identifying key strengths and missed opportunities in instructional strategies, and providing actionable suggestions for fostering student reasoning (Wang and Demszky, 2023). The findings revealed that ChatGPT struggled to score classroom transcripts using instruments like the Classroom Assessment Scoring System (CLASS) or the Mathematical Quality of Instruction (MQI) and offered repetitive feedback.

Research has also explored the application of LLMs at more granular levels, such as sentence-level or utterance-level analysis. While zero-shot ChatGPT provided clear and detailed explanations for its predictions, it performed significantly worse than the smaller BERT model in three out of four student talk move categories for the classification task (Wang et al., 2023). Tran et al. (2024b) analyzed three prompt-based factors: task formulations, context length, and the presence of few-shot examples and found that all of them can have impacts on the final performance. Although the importance of few-shot examples has been shown and some prior work utilized few-shot prompting, they had a fixed set of examples, which might not be representative enough and might not have examples relevant to the current input (Tran et al., 2024a,b). *Our work focuses on automatically retrieving a set of examples based on the input to cover dynamic scenarios in a classroom discussion.*

2.2 ICL Example Retrieval for LLMs

In-context learning, the emergent capability of LLMs that allows them to execute diverse tasks by conditioning on a limited set of input-output examples without requiring parameter updates or finetuning, has been demonstrated in many LLMs such as GPT-3 (Brown et al., 2020) or Llama (Touvron et al., 2023). Various approaches have been made to create better LLM prompts (Li and Liang, 2021; Le Scao and Rush, 2021; Hao et al., 2022). Different from the standard retrieval-augmented generation by using a dense retriever such as ColBERT (Khattab and Zaharia, 2020) to get additional information for LLMs, there is an area of research focused specifically on finding better ICL

examples to boost LLMs' performance (Ye et al., 2023; Li and Qiu, 2023; Li et al., 2023; Zhang et al., 2023). Liu et al. (2022) demonstrated that ICL performance can be enhanced by either using the BM25 algorithm or by finetuning dense retrievers with feedback from LLMs to retrieve relevant examples from a training set. Wang et al. (2024) proposed an iterative training framework (LLM-R) to retrieve ICL examples in 3 steps: 1) rank an initial set of retrieved candidates based on the conditional LLM log probabilities of the ground-truth outputs; 2) train a cross-encoder reward model to capture the fine-grained ranking signals from LLMs; and 3) train a bi-encoder dense retriever using knowledge distillation. Our work falls into this area by proposing a method to dynamically retrieve ICL examples. However, we focus on the label ratio of the example set, which has not been studied in prior work.

3 Dataset

We use videos of English Language Arts classes in a Texas district to create our corpus. The videos were recorded during the course of an online instructional coaching program (Correnti et al., 2021). They were collected from 18 fourth-grade and 13 fifth-grade classes, whose teachers on average had 13 years of teaching experience. 61% of the student population was considered low income, with the following racial proportion: Latinx (73%), Caucasian (15%), African American (7%), multiracial (4%), and Asian or Pacific Islander (1%).

Annotators manually scored videos holistically, on a scale from 1 to 4, using the IQA on 11 dimensions (Matsumura et al., 2008) for both teacher and student contributions. They were also scored using more fine-grained talk moves annotated at the sentence level using the *Analyzing Teaching Moves (ATM)* discourse measure (Correnti et al., 2021). The final corpus consists of **112** discussion transcripts that have been scored using both the IQA and the ATM (see Appendix A for the statistics of the scores). Thirty-two videos (29%) were double-scored, showing good to excellent reliability for the IQA (the Intraclass Correlation Coefficients (ICC) range from .89-.98) and moderate to good reliability for the ATM (ICC range from .57 to .85). Below is an excerpt with annotated ATM codes:

Teacher: [Justin.]Repeat [Tell me who's Justin?]
Press

Student: [Justin is... Well, Via's boyfriend who stands up for August and

is very nice to him. Even though he saw him for the first time, he was kind of shocked, but he kind of got used to him.]**Strong Explanation**

IQA dimension scores. To compare with prior work (Tran et al., 2024a), we focused on 4 of the 11 IQA dimensions, in which 2 of them focus on teaching moves and 2 focus on student contributions. They were previously chosen because of their relevance to dialogic teaching principles that emphasize collaboration and active participation in meaning-making. Furthermore, all four scores are calculated based on the frequency of their related ATM codes. The four dimensions include: *Teacher links Student’s contributions* (T-Link), *Teacher presses for information* (T-Press), *Student links other’s contributions* (S-Link), *Student supports claims with evidence and explanation* (S-Evid). We define S-Evid as the higher score of *Student provides text-based evidence* and *Student provides explanation*. Descriptions of these dimensions can be found in Appendix B.

Based on the definitions of the ATM codes (Appendix C), 2 IQA dimensions have hard negative examples (e.g., have examples that are semantically similar to the positive examples but have a different label based on a notion of strength). For S-Link, a positive example has *Strong Link* as the ATM label while a hard negative example has *Weak Link*. A similar rule applies to S-Evid (*Strong Text-based Evidence* vs *Weak Text-based Evidence*; *Strong Explanation* vs *Weak Explanation*).

Due to the small size of the data, we follow Tran et al. (2024a) and use 2-fold cross-validation. In each fold, half of the data (56 transcripts) is considered as training data and the remaining data (56 transcripts) is used for evaluation. We also make sure that transcripts of the same teacher are in the same fold to prevent data leakage.

4 Methods

4.1 ICL Example Retrieval for LLMs

We adopt the prompts from prior work (Tran et al., 2024a) for our LLM. We utilize the predictive approach, which is the approach that yields the best results in all 4 IQA dimensions (Predictive-llm). It is the BC-5turns-10s strategy described by Tran et al. (2024a), utilizing the LLM as a binary classifier by prompting it to determine whether an observation related to an IQA dimension is present in a single turn (yes or no) (see Appendix D).

For our ICL example retriever, we use LLM-R (Wang et al., 2024)¹. It uses LLMs to rank the candidates based on the log-likelihood of the ground-truth output, then trains a cross-encoder as a reward model to mimic the preferences of LLMs, and finally distills that knowledge to a bi-encoder for efficient inference. For a given input (a 5-turn dialogue excerpt), we retrieve the top 10 examples from the training data and use them as few-shot examples in the LLM prompt. We use separate retrievers (LLM-R) for teachers’ and students’ turns. In other words, when predicting a teacher or student’s turn, we will only try to retrieve examples from a pool consisting of examples from the same speaker role (student or teacher). For example, if we are predicting if the last turn (given its 4 previous turns) is T-Press, the retriever will only try to find examples (5-turn dialogue windows) by looking at ones that end with a teacher’s turn.

Although LLM-R specializes in ICL example retrieval, it was trained on tasks different from classroom discussions (e.g., sentiment, reading comprehension, closed-domain QA). Besides using off-the-shelf LLM-R, we also fine-tune it on classroom discussions. However, because our dataset is small, finetuning an ICL example retriever on the training set is ineffective. We instead use another classroom discussion dataset, TalkMoves (Suresh et al., 2022), to finetune LLM-R.

The TalkMoves dataset contains K-12 math classroom transcripts, annotated for talk moves based on accountable talk theory and dialog acts. The dataset includes 567 transcripts, comprising 174,186 annotated teacher utterances, 59,874 annotated student utterances, and 1.8 million words (15,830 unique). All of the transcripts are annotated for 6 teacher talk moves (Keeping everyone together, Getting students to relate to another’s ideas, Restating, Pressing for accuracy, Revoicing, and Pressing for reasoning) and 4 student talk moves (Relating to another student, Asking for more info, Making a claim, and Providing evidence or reasoning). For finetuning the retriever, we use the same binary prediction task as Predictive-llm. However, we perform multiple binary predictions (yes/no) for all possible talk moves in each turn and use the definitions of these talk moves from the dataset (Suresh et al., 2022). While these moves differ from ATM codes, they share similarities and reflect

¹https://github.com/microsoft/LMOps/tree/main/llm_retriever

a theoretical approach closely related to the one behind ATM.

4.2 Re-ranking and Label Ratio Regulation

In this section, we propose a method that uses re-ranking and forces a specific label ratio in the example set to improve ICL performance for classroom discussion quality assessment.

Re-ranking. Re-ranking is a popular approach in retrieval tasks. The initial retrieval process is generally designed to be fast, often prioritizing speed over perfect accuracy. As a result, in ICL example retrieval, the first batch of examples retrieved can be broad, including both highly relevant and somewhat irrelevant information. Re-ranking addresses this by filtering and reordering these examples according to refined relevance scores, reducing noise and irrelevant information. In the first step, we re-rank the top-100 retrieved examples to get a set of examples ordered by their usefulness. We experiment with 2 re-ranking methods.

LLM as a re-ranker: We use a Llama3 model as the scorer. Specifically, for a given input and a retrieved example, we ask a yes/no question if the example can help answer the given question and use the probability of “yes” as the score.

BERT as a re-ranker: We train a BERT-based model as a cross-encoder reward model that gives higher scores to good ICL examples. We first create the necessary training data to train the BERT model. To do this, from our available training data (Section 3), for each instance (a turn), we retrieve the top-K using the LLM-R retriever (either trained or not trained). We then employ Llama3 to obtain the rankings. The ranking score is calculated as $\log p(y|x, x_i, y_i)$ where x is the given input, y is the gold answer, x_i and y_i are an in-context learning example retrieved and its label. For a training example (x, y) , we first sample one positive example (x^+, y^+) from the top-ranked candidates and N_{neg} negative examples $(x_i^-, y_i^-)_{i=1}^{N_{neg}}$ from the bottom-ranked candidates. The reward model takes as input the concatenation of (x, y, x^+, y^+) and produces a score $s(x, y, x^+, y^+)$ for the positive example, and $s(x, y, x_i^-, y_i^-)$ for the negatives. The training objective is to minimize the cross-entropy loss:

$$\mathcal{L}_{\text{reward}} = -\log \frac{e^{s(x,y,x^+,y^+)}}{e^{s(x,y,x^+,y^+)} + \sum_{i=1}^{N_{\text{neg}}} e^{s(x,y,x_i^-,y_i^-)}}$$

Label Ratio Regulation (LRR). Thinking that the lack of hard negative examples and the imbalance of positive/negative examples can be potential

causes for the poor performance of the off-the-shelf retrieval setting, we want to ensure that this will not happen. To do so, we make sure the 10-example set follows a specific label ratio of positive, negative, and hard negative (if applicable) examples. For a fair comparison, we force the ratio to mimic the ratio from the fixed setting (defined in Tran et al. (2024a)). Although the ordering of few-shot examples is also a non-trivial factor (Ye et al., 2022), it is not what we focus on. Therefore, we fix the order of the chosen examples. For T-Press and T-Link, from top to bottom, we want 5 positive and then 5 negative examples. Similarly, for S-Link and S-Evid, we will see 4 positive, 4 easy negative, and 3 hard negative examples, respectively, from top to bottom of the example set. To do so, given the ranked list of examples, we pick from top to bottom until the predetermined label ratio is satisfied and skip examples that violate the label ratio if added. For instance, if we already have 5 positive examples for T-press, we will ignore the remaining positive examples in the list and only pick an example if it is a negative one as we go down the list.

5 Experimental Setup

To make it comparable to prior work without ICL example retrieval from Tran et al. (2024a), we use LLama3-8B (Grattafiori et al., 2024) as the LLM for classroom discussion assessment ².

We use 3 baselines to test the effectiveness of the proposed method:

fixed: In this setting, we use a set of 10 fixed examples for each fold in the cross-validation. We follow prior work to pick those 10 examples for the LLM prompts (Tran et al., 2024a). This setting is also used as a baseline for comparison with approaches that utilize ICL example retrieval. One thing to note is that using this sampling method, we will have a fixed ratio of positive, easy negative, and hard negative examples in the 10-shot example set.

retrieved: In this setting, we use LLM-R (Wang et al., 2024) to find the top-10 examples from the training data. Then, we use those 10 examples for few-shot prompting.

mixed: In this setting, we construct a set of top-5 retrieved examples and 5 examples from the fixed set. For the 5 examples from the fixed set, we pri-

²<https://huggingface.co/meta-llama/Llama-3.1-8B>

ortize hard-negative examples first. In addition, prior work has shown that lacking hard negative examples is detrimental to the performance (Tran et al., 2024b). Therefore, we decide to select harder negative examples from the fixed set, as we cannot guarantee that they exist in the retrieved set. Specifically, for S-Link and S-Evid, we pick 3 hard negative examples, 1 positive example, and 1 non-hard negative example. For T-Press and T-Link, we pick 3 positive and 2 negative examples. Then, we choose the remaining 5 examples from the retrieved ones based on the descending order of cosine similarity between them and the input embedded by LLM-R.

To test the performance of the proposed method, we experimented with 2 re-ranking methods: using LLama3 as the *LLM re-ranker* and using a finetuned *BERT re-ranker*. To highlight the importance of each component (LRR and re-ranking), we report the performance from utilizing both re-ranking and label ratio regulation in combination and from using each component separately.

To compare the performances between non-finetuned and finetuned retrievers, we finetune a new LLM-R on another classroom discussion dataset (TalkMoves from Suresh et al., 2022) and repeat the experiments.

Quadratic Weighted Kappa (QWK) is used as the main evaluation metric. It is a common metric for quantifying inter-rater reliability that penalizes disagreements proportional to the degree of disagreement, which is vital in contexts where a greater distance between scores is meaningful.

6 Results and Discussion

RQ₁: Effectiveness of the proposed method. Table 1 shows the macro average over 2-fold cross-validation of QWK scores in various settings, including the 3 baselines and the proposed method for both non-finetuned and finetuned retrievers.

The standard ICL example retrieval is not effective. When using a non-finetuned LLM-R, we observe that relying solely on retrieved examples (row 2) is worse than the *fixed* baseline (row 1). This implies that using ICL retrieval is ineffective in this case, despite helping to improve performance in previous work on other domains (Wang et al., 2024; Zhang et al., 2023). On the other hand, the *mixed* settings (row 3), where we combine examples from the retriever with the fixed set, are the baselines that achieve the best performance in all

IQA dimensions. This suggests that the retrieved examples are still useful to some extent.

Our proposed method with BERT as the re-ranker achieves the best performance in all 4 IQA dimensions (row 7) for both non-finetuned and finetuned retrievers. Although finetuning the LLM-R boosts the performance of the *retrieved* setting (row 2), the proposed method performs comparably for both non-finetuned and finetuned settings of the LLM-R retriever (row 7), suggesting that finetuning the retriever on a new domain, which is computationally expensive, is not necessary. Our hypothesis for this minimal gain is that the TalkMoves data consists of math discussions, which contain math-specific lexicons not present in English Language Art discussions from our dataset. Additionally, the TalkMoves dataset is skewed towards sixth-grade to eighth-grade students, while our data only has discussions from fourth-grade and fifth-grade students.

As a re-ranker, although LLama3 shows equal or better performance over the *retrieved* setting in T-Link and T-Press (row 5 vs 2), it is inferior to the *fixed* setting in S-Link and S-Evid (row 5 vs row 1). On the other hand, using BERT as a re-ranker with label ratio regulation achieved the best results in all dimensions. With this combination, we are now able to outperform the mixed setting despite using only retrieved examples. This implies that for this task, using an LLM such as LLama3 as a judge for re-ranking is not a reliable method in comparison with finetuning a PLM such as BERT.

The LRR is shown to be essential for improved performance as removing it leads to decreases in QWK (rows 6 and 8 compared to the previous rows). The drop in performance in S-Link and S-Evid is larger than the drop in T-Link and T-Press. The former 2 dimensions (S-Link and S-Evid) have hard negative examples based on the coding manual, which suggests that LRR is more important when hard negative examples are available for the target dimension. With only re-ranking, we can perform similarly or worse than the *retrieved* setting. For instance, using a LLama3 re-ranker without LRR is worse than vanilla retrieval (row 6 versus 2). On the other hand, with LRR, we consistently outperform the *retrieved* setting, with or without using a re-ranker (row 4, 5, 7 versus 2)⁴. Moreover, when the retriever is finetuned, if we have to pick

³Two-tailed t-test on 2-fold cross-validation.

⁴Except for S-Link with finetuned retriever.

ID	Setting	Non-finetuned Retriever				Finetuned Retriever			
		T-Link	T-Press	S-Link	S-Evid	T-Link	T-Press	S-Link	S-Evid
1	fixed	0.65	0.73	0.64	0.79	0.65	0.73	0.64	0.79
2	retrieved	0.62	0.71	0.62	0.75	0.66	0.73	0.66	0.80
3	mixed	<i>0.68</i>	<i>0.76</i>	<i>0.67</i>	<i>0.81</i>	<i>0.72</i>	<i>0.77</i>	<i>0.71</i>	<i>0.82</i>
4	LRR only	0.63	0.72	0.65	0.79	0.68	0.76	0.65	0.81
5	Llama3 + LRR	0.65	0.72	0.62	0.76	0.68	0.75	0.63	0.77
6	w/o LRR	0.61	0.70	0.56	0.68	0.65	0.72	0.60	0.70
7	BERT + LRR	0.72	0.80	0.73	0.83	0.73	0.81	0.73	0.83
8	w/o LRR	0.66	0.78	0.64	0.77	0.66	0.78	0.65	0.77

Table 1: Quadratic Weighted Kappa (QWK) scores of the two retrievers. For each IQA dimension (T-Link, T-Press, S-Link, S-Evid), italic numbers represent the best baseline results. Bold numbers highlight the best retriever results. All numbers are statistically significant compared to their counterparts in the *mixed* baseline ($p < 0.05$).³

IQA	Non-finetuned LLM-R		Finetuned LLM-R	
	Avg	% w/o hard negative	Avg	% w/o hard negative
S-Link	3 / 1.2 / 3.2 / 1.5	0 / 27.2 / 0 / 24.3	3 / 1.5 / 3.3 / 1.7	0 / 23.3 / 0 / 20.7
S-Evid	3 / 1.9 / 3.1 / 2.1	0 / 22.7 / 0 / 20.8	3 / 1.7 / 3.4 / 2.0	0 / 20.5 / 0 / 19.1

Table 2: Presence of hard negative examples in the fixed, retrieved, mixed setting and an approach utilizing BERT re-ranking without LRR. We report the average number of hard negative examples included in the 10 examples (Avg) and the percentage of test instances where the few-shot examples in the prompt do not have any hard negative example. In each cell, from left to right, the 4 numbers represent the statistics for *fixed*, *retrieved*, *mixed* settings, and from an approach utilizing BERT re-ranking without LRR.

only one component, using LRR is usually better than using a re-ranker (row 4 versus rows 6 and 8). This suggests that we should always enforce the label ratio in the example set.

RQ₂: Issues in the label ratio of retrieved examples from automatic ICL example retrieval.

The lack of hard negative examples and skew in the ratio of positive and negative examples can be potential causes for the low performance of example retrieval. Noticing that directly using the retrieved examples is not an effective way to improve the performance of LLM-based classroom discussion quality assessment, we hypothesize the potential causes and do analyses to test them. Compared to the *fixed* and *mixed* settings, one thing we could not control in the *retrieved* setting is the distribution of the examples. We can think of two causes for the poor performance using the *retrieved* setting: 1) the lack of hard negative examples and 2) the lack of positive examples.

Missing hard negative examples in the few-shot example set will have a negative influence (Tran et al., 2024a). Table 2 shows the presence of hard

negative examples in the *fixed*, *retrieved*, *mixed* settings, and an approach using BERT ranking without LRR for S-Link and S-Evid (the only two dimensions that have hard negative examples according to the definitions in the coding manual). We can see that the *retrieved* setting has fewer hard negative examples on average compared to the *fixed* and *mixed* settings. We also only witness cases in which the example set has no hard negative examples in the retrieved setting. With only a BERT re-ranker, these numbers barely change as we only see small increases in the average number of hard negative examples and decreases in the number of cases without any hard negative example compared to the retrieved setting (4th number versus 2nd number) in each cell. This aligns with one of our previous observations from Section 6 that removing LRR results in bigger decreases in QWK for S-Link and S-Evid compared to the other two dimensions. This implies that re-ranking alone still does not guarantee the presence of hard negative examples in the set of 10 few-shot examples for prompting. However, with domain knowledge of

IQA	Non-finetuned LLM-R		Finetuned LLM-R	
	Avg	% without positive	Avg	% without positive
T-Link	5 / 3.7 / 4.7 / 4.2	0 / 6.8 / 0 / 1.2	5 / 3.3 / 3.5 / 3.5	0 / 5.3 / 0 / 2.3
T-Press	5 / 7.3 / 5.4 / 6.1	0 / 0.0 / 0 / 0.0	5 / 6.8 / 5.2 / 5.5	0 / 0.0 / 0 / 0.0
S-Link	4 / 3.2 / 3.8 / 3.5	0 / 6.2 / 0 / 0.0	4 / 3.8 / 4.3 / 4.1	0 / 0.0 / 0 / 0.0
S-Evid	4 / 5.9 / 5.1 / 5.5	0 / 2.3 / 0 / 0.0	4 / 5.6 / 4.9 / 4.3	0 / 3.1 / 0 / 1.2

Table 3: Presence of positive examples in the fixed, retrieved, mixed setting, and an approach with only BERT re-ranking (no LRR), with the same notations as Table 2.

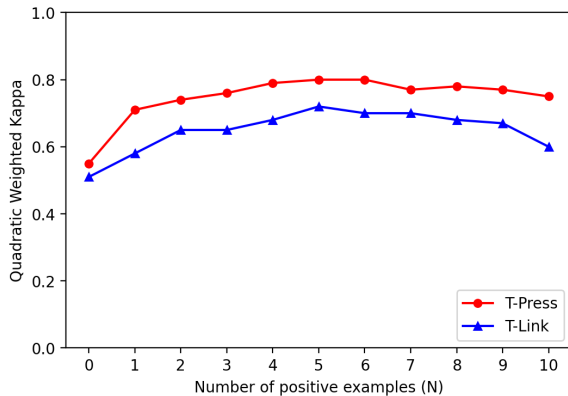


Figure 2: Results of T-Press and T-Link from different label ratios with N positive examples for BERT+LRR.

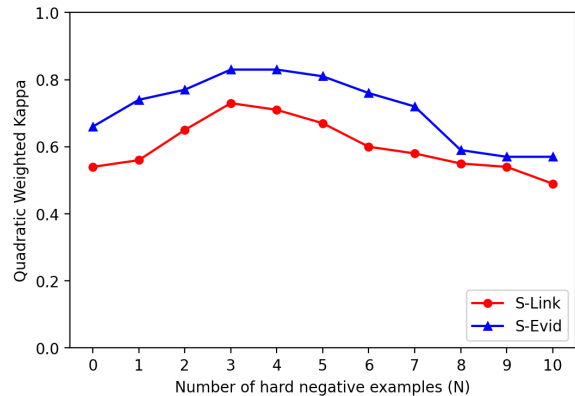


Figure 3: Results of S-Link and S-Evid from different label ratios with N hard negative examples for BERT+LRR.

hard negative examples (i.e., knowing that Weak Link is a hard negative example label for Strong Link, which represents S-Link), even with automated retrieval, we can ensure that hard negative examples are in the set.

Looking at the presence of positive examples ($x_i, y_i = y$) in Table 3, we see that the *retrieved* settings (2nd numbers of the cell) tend to include more positive examples in the 10-example set for T-Press and S-Evid while having fewer positive examples for T-Link and S-Link. Although they are rare, there are still cases in which we have no positive examples in the 10-example set for the *retrieved* setting, which never happens for the *fixed* and *mixed* settings. The BERT re-ranking (4th number) helps decrease the number of cases without any positive examples, and it makes the average number of positive examples retrieved in each IQA dimension closer to the *fixed* and *mixed* settings.

RQ₃: The ratio of different types of examples does matter. We conduct additional experiments on our best model (BERT+LRR) by varying the ratios by changing the number of positive examples, negative examples, and hard negative examples (if they are available) to see if certain label ratios yield

better results. Specifically, for T-Press and T-Link, because there is no hard negative example for these two dimensions, we record the performance with the N positive examples (N = 0 to 10) and 10-N negative examples. For S-Link and S-Evid, we pick N (N = 0 to 10) hard negative examples and equally split the remaining 10-N examples into positive examples and non-hard negative examples (if 10-N is odd, we have 1 more positive example).

Figure 2 shows the results for T-Press and T-Link with various positive/negative ratios. We observe that increasing the number of positive examples helps improve the performance to a certain point. Specifically, we see noticeable improvements until N=3, then it starts to slow down. However, after N=5 positive examples, the QWK begins to go down as N increases. This suggests that we should have a balance between positive and negative examples, which is reasonable, as having too many examples of a certain perspective (positive or negative) can create biases in that direction for the LLMs. Similarly, for S-Link and S-Evid, the performances are boosted until N=3 hard negative examples are selected and then they quickly drop (Figure 3). This

time, the downward trend after $N=4$ is more noticeable compared to Figure 2, implying that having too many hard negative examples causes more harm than good. In other words, although hard negative examples are essential for high performance, we should keep room for positive and non-hard negative examples. Overall, these observations suggest that we should be cautious when selecting a label ratio for the example set, and having a balanced split between the possible labels (positive, non-hard negative, and hard negative) is a safer choice, as the dominance of a label tends to result in decreased performance.

7 Conclusions

This work proposes a simple but effective ICL example retrieval method that utilizes example re-ranking and label ratio regulation (LRR) to improve few-shot LLM performance in automated classroom discussion assessment. The results show that our fully automated example retrieval and selection approach outperforms the baselines in all tested IQA dimensions. Additionally, the performance of a non-finetuned example retriever (LLM-R) is comparable to that of a retriever finetuned on a similar domain dataset, suggesting that skipping the finetuning process of the retriever is viable. Further analyses show that the lack of positive and hard negative examples can be the reason for the poor performance of the traditional ICL example retrieval approaches. We also observe that even with re-ranking, both finetuned and non-finetuned retrievers fail to select enough hard negative examples to make the few-shot prompting effective, which highlights the importance of label ratio regulation in maintaining the presence of hard negative examples. Finally, we investigate the influence of the ratio of positive, non-hard negative, and hard negative examples, demonstrating that having an excessive number of any category hurts performance. We would like to explore the proposed method with a more advanced prompting method, such as Chain-of-thought (Wei et al., 2022), in the future.

Limitations

While our method uses Label Ratio Regulation (LRR) to maintain a specific ratio, it treats each potential example separately when selecting the top-10. This independent selection might not be ideal, as the chosen examples can interact with each

other. Exploring combinatorial optimization and sequential decision-making techniques could lead to improvements.

Another limitation of our study is the lack of analysis on the influence of the size of the example pool on the performance. Because our training set is small, the relevance and diversity of the candidate examples can be a hindrance to the generic ICL example retrieval baseline. If we have a bigger example pool with more diversity, the LRR might become unnecessary.

The proposed approach involves several components, which people might find too complex and counterintuitive, potentially hindering the ease of LLM usage for downstream tasks. Additionally, the experiments are conducted on a single and small dataset. As a result, the generalizability of the findings is weakened.

Furthermore, we only use the smallest version of LLama3. Utilizing a bigger LLM (e.g., LLama3-70B) might yield higher results and undermine the effectiveness of ICL example retrieval.

Last but not least, our experiments are grounded on a binary classification task and the assumption that hard negative examples can be identified based on the definitions of the labels.

Acknowledgments

We thank the Learning Research and Development Center for their student financial support, as well as the Pitt PETAL group and the anonymous reviewers for their useful comments for improving this work.

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, 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.
- Richard Correnti, Lindsay Clare Matsumura, Marguerite Walsh, Dena Zook-Howell, Donna DiPrima Bickel, and Baeksan Yu. 2021. Effects of online content-focused coaching on discussion quality and reading achievement: Building theory for how coaching develops teachers' adaptive expertise. *Reading Research Quarterly*, 56(3):519–558.

- Laura M. Desimone and Katie Pak and. 2017. [Instructional coaching as high-quality professional development](#). *Theory Into Practice*, 56(1):3–12.
- 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.
- 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.
- Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. [Structured prompting: Scaling in-context learning to 1,000 examples](#). *Preprint*, arXiv:2212.06713.
- Jennifer Jacobs, Karla Scornavacco, Charis Harty, Abhijit Suresh, Vivian Lai, and Tamara Sumner. 2022. [Promoting rich discussions in mathematics classrooms: Using personalized, automated feedback to support reflection and instructional change](#). *Teaching and Teacher Education*, 112:103631.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Omar Khattab and Matei Zaharia. 2020. [Colbert: Efficient and effective passage search via contextualized late interaction over bert](#). In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’20*, page 39–48, New York, NY, USA. Association for Computing Machinery.
- Ashlee Kupor, Candice Morgan, and Dorottya Demszky. 2023. [Measuring five accountable talk moves to improve instruction at scale](#). *arXiv preprint*.
- Teven Le Scao and Alexander Rush. 2021. [How many data points is a prompt worth?](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). 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 4582–4597, Online. Association for Computational Linguistics.
- Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. [Unified demonstration retriever for in-context learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.
- Xiaonan Li and Xipeng Qiu. 2023. [Finding support examples for in-context learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. 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.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Lindsay Clare Matsumura, Helen E. Garnier, Sharon Cadman Slater, and Melissa D. Boston. 2008. [Toward measuring instructional interactions “at-scale”](#). *Educational Assessment*, 13(4):267–300.
- Sewon Min, Xinxin Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. [Rethinking the role of demonstrations: What makes in-context learning work?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. [Contrastive learning with hard negative samples](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

- 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.
- Abhijit Suresh, Tamara Sumner, Jennifer Jacobs, Bill Foland, and Wayne Ward. 2019. [Automating analysis and feedback to improve mathematics teachers’ classroom discourse](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9721–9728.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Nhat Tran, Benjamin Pierce, Diane Litman, Richard Correnti, and Lindsay Clare Matsumura. 2024a. [Analyzing large language models for classroom discussion assessment](#). In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 500–510, Atlanta, Georgia, USA. International Educational Data Mining Society.
- Nhat Tran, Benjamin Pierce, Diane Litman, Richard Correnti, and Lindsay Clare Matsumura. 2024b. [Multi-dimensional performance analysis of large language models for classroom discussion assessment](#). *Journal of Educational Data Mining*, 16(2):304–335.
- Deliang Wang, Dapeng Shan, Yaqian Zheng, Kai Guo, Gaowei Chen, and Yu Lu. 2023. [Can chatgpt detect student talk moves in classroom discourse? a preliminary comparison with bert](#). In *Proceedings of the 16th International Conference on Educational Data Mining*, pages 515–519, Bengaluru, India. International Educational Data Mining Society.
- 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, St. Julian’s, Malta. 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.
- 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.
- 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*, 16(1):34–60.
- Ian A. G. Wilkinson, P. Karen Murphy, and Sevda Binici. 2015. *Dialogue-Intensive Pedagogies for Promoting Reading Comprehension: What We Know, What We Need to Know*, pages 37–50. American Educational Research Association.
- Jiacheng Ye, Jiahui Gao, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2022. [ProGen: Progressive zero-shot dataset generation via in-context feedback](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3671–3683, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. [Retrieve anything to augment large language models](#). *Preprint*, arXiv:2310.07554.

A Data statistics

The transcripts have an average length of 3,421 tokens, with a median length of 3,537 tokens. The shortest transcript contains 1,986 tokens, while the longest reaches 6,393 tokens. Table 4 presents the statistics for the four key IQA dimensions highlighted in this work.

B Description of IQA Dimensions

Descriptions of the 4 focused IQA dimensions can be found in Table 5.

C Description of ATM codes

Descriptions of the relevant ATM codes can be found in Table 6.

D Prompts

Figure 4 contains the prompt used for the binary prediction of a target IQA adopted from Tran et al. (2024a), where {IQA description} is from the second column in Table 5.

IQA	Distribution	Avg Score	Relevant ATM code	Hard negative ATM Code
T-Link	[69, 23, 9, 11]	1.66	Recap or Synthesize S Ideas	n/a
T-Press	[8, 13, 11, 80]	3.46	Press	n/a
S-Link	[84, 7, 10, 11]	1.54	Strong Link	Weak Link
S-Evid	[38, 17, 9, 48]	2.60	Strong Text-based Evidence	Weak Text-based Evidence
			Strong Explanation	Weak Explanation

Table 4: Data distribution and mean (**Avg**) of 4 focused *IQA* rubrics for Teacher (*T*) and Student (*S*) with their relevant *ATM* codes and hard negative *ATM* code (if available). An *IQA* rubric's distribution is represented as the counts of each score (1 to 4 from left to right) (n=112 discussions).

IQA Dimension	IQA Dimension's Description
T-Link: Teacher links Student's contribution	<i>Did Teacher support Students in connecting ideas and positions to build coherence in the discussion about a text?</i> 4: 3+ times during the lesson, Teacher connects Students' contributions to each other and shows how ideas/ positions shared during the discussion relate to each other. 3: Twice... 2: Once... OR The Teacher links contributions to each other, but does not show how ideas/positions relate to each other (re-stating). 1: The Teacher does not make any effort to link or revoice contributions.
T-Press: Teacher presses Students	<i>Did Teacher press Students to support their contributions with evidence and/or reasoning?</i> 4: 3+ times, Teacher asks Students academically relevant Questions, which may include asking Students to provide evidence for their contributions, pressing Students for accuracy, or to explain their reasoning. 3: Twice... 2: Once... OR There are superficial, trivial, or formulaic efforts to ask Students to provide evidence for their contributions or to explain their reasoning. 1: There are no efforts to ask Students to provide evidence for their contributions or to ask Ss to explain their reasoning.
S-Link: Student links other's contributions	<i>Did Students' contributions link to and build on each other during the discussion about a text?</i> 4: 3+ times during the lesson, Students connect their contributions to each other and show how ideas/positions shared during the discussion relate to each other. 3: Twice... 2: Once... OR the Students link contributions to each other, but do not show how ideas/positions relate to each other (re-stating). 1: The Students do not make any effort to link or revoice contributions.
S-Evid(a): Student provides text-based evidence	<i>Did Students support their contributions with text-based evidence?</i> 4: 3+ times, Students provide specific, accurate, and appropriate evidence for their claims in the form of references to the text. 3: Twice... 2: Once... OR There are superficial or trivial efforts to provide evidence. 1: Students do not back up their claims.
S-Evid(b): Student provides explanation	<i>Did Students support their contributions with reasoning?</i> 4: 3+ times, Students offer extended and clear explanation of their thinking. 3: Twice... 2: Once... OR There are superficial or trivial efforts to provide explanation. 1: Students do not explain their thinking or reasoning.

Table 5: *IQA* dimensions and their definitions. For each *IQA* dimension, the italic line is {*IQA* description} used in the prompt in Appendix D.

Code	Definition	Example
Press	T asks the same S follow-up Questions (i.e., uptake/push-back Q's, request for text-based evidence and explanation).	Why did you say that? Where is the evidence? How else might Salva feel?
Recap or Synthesize S Ideas	T links multiple Ss' ideas or positions. T synthesizes multiple responses.	What I hear you saying is that the character has changed from the beginning of the book which is similar to Ana's idea that the character has matured.
Weak Link	Ss attempt to link contributions to each other, but do not show how ideas/positions relate to each other. The S might simply be revoicing or repeating another S's contribution.	"I disagree with Ana"... without explaining why or which aspect of Ana's statement S disagrees with.
Strong Link	Ss connect their contributions to each other and show how ideas/positions shared during the discussion relate to each other. Ss elaborate, challenge, or build on each other's ideas.	I'm not sure what Ana says is right because I don't see where in the text it says that. . .
Weak Text-Based Evidence	Ss provide inaccurate, incomplete, inappropriate, vague, or trivial evidence from/reference to text	Naya lived a hard life because in the chapters about her, we learn that she has to do a lot of things for her family.
Strong Text-Based Evidence	Ss provide accurate, appropriate, specific evidence from/reference to text that supports claim	On page 59, in the last paragraph it says, "I have talked to the others here," uncle Jake said. "We believe that the village of Loun-Ariik was attacked and probably burned your family." Uncle paused and looked away."
Weak Explanation	S provides a brief or circular explanation that basically repeats or restates the response or relies on evidence to speak for itself.	I think that they didn't catch the fish because, , Tim hasn't caught any fish and Tim and Tom haven't caught any fish lately.
Strong Explanation	Ss provide an elaboration/justification of their answer or of the evidence they selected to support their answer.	Yeah, it is. The cause is, he didn't get the little girl's advice so, the effect of that is the calabash broke.

Table 6: ATM codes and their definitions

Prompt for binary prediction

Given a dialogue between a teacher and students in a classroom, in the last turn, {IQA description}?

Example 1

Dialogue: {Example Excerpt 1(5-turn)}

Answer (yes or no): {Example Answer 1}

...

Example 10

Dialogue: {Example Excerpt 10 (5-turn)}

Answer (yes or no): {Example Answer 10}

Input

Dialogue: {Dialogue}

Answer (yes or no):

Figure 4: Prompt templates for binary prediction.

Exploring LLMs for Predicting Tutor Strategy and Student Outcomes in Dialogues

Fareya Ikram, Alexander Scarlatos, Andrew Lan

University of Massachusetts Amherst
{fikram, ajscarlatos, andrewlan}@umass.edu

Abstract

Tutoring dialogues have gained significant attention in recent years, given the prominence of online learning and the emerging tutoring abilities of artificial intelligence (AI) agents powered by large language models (LLMs). Recent studies have shown that the strategies used by tutors can have significant effects on student outcomes, necessitating methods to predict how tutors will behave and how their actions impact students. However, few works have studied predicting tutor strategy in dialogues. Therefore, in this work we investigate the ability of modern LLMs, particularly Llama 3 and GPT-4o, to predict both future tutor moves and student outcomes in dialogues, using two math tutoring dialogue datasets. We find that even state-of-the-art LLMs struggle to predict future tutor strategy while tutor strategy is highly indicative of student outcomes, outlining a need for more powerful methods to approach this task.

1 Introduction

Tutoring has been shown to be highly effective in increasing student learning, both when administered by human tutors or intelligent tutoring systems (Nickow et al., 2020; Nye et al., 2014). Recently, several automated tutors, powered by large language models (LLMs), have been deployed in educational settings, such as Khan Academy’s Khanmigo (Khan Academy, 2023) or Carnegie Learning’s LiveHint (Carnegie Learning, 2024). To ensure that students benefit from these tools, it is important to study the strategies used by tutors and how they impact student learning outcomes.

Tutor strategy is commonly formalized using “moves”, or high-level pedagogical actions taken in any given dialogue turn to support student learning (Demszky and Hill, 2023; Macina et al., 2023; Suresh et al., 2022). Recent studies have shown that explicit move and strategy information can be used to improve tutor effectiveness (Wang et al.,

2024a,b). Others that train LLMs to be effective tutors (Tack et al., 2023; Team et al., 2024; Sonkar et al., 2024; Huber et al., 2023; Vasselli et al., 2023; Scarlatos et al., 2025b) have also highlighted the importance of pedagogical strategy. Several prior works have used tutor moves to predict student outcomes (Lin et al., 2022; Borchers et al., 2024; Abdelshiheed et al., 2024; Yin et al., 2025), though text alone processed by LLMs is often sufficient (Scarlatos et al., 2025a; Chen et al., 2024). In this work, we explicitly study the effect of move annotations, compared to text alone, on predicting tutor strategy and student outcomes.

While many works have studied how to *identify* tutor moves in dialogues (Demszky et al., 2021; Wang and Demszky, 2024; Moreau-Pernet et al., 2024; Wang et al., 2023; McNichols and Lan, 2025), there are few works studying how to predict *future* tutor moves. One prior work does so using GRUs and RoBERTa (Ganesh et al., 2021), though to the best of our knowledge, none have studied how generative language models, such as Meta’s Llama 3 (Dubey et al., 2024) or OpenAI’s GPT-4 (OpenAI, 2024b), perform on this task.

In this work, to address the needs of understanding tutor strategy and its effect on student outcomes, we seek to answer the following research questions: **RQ1**: Can LLMs predict *tutor strategy* using tutor moves and dialogue history?, **RQ2**: Can LLMs predict *student outcomes* using tutor moves and dialogue history?, and **RQ3**: Which tutoring strategies have the highest impact on student outcomes? To the best of our knowledge, our study is the first to jointly address these questions using modern generative LLMs. Overall, we find that tutor strategy prediction is highly challenging, with student outcome prediction being easier and facilitated by tutor move annotations. Our findings indicate that tutor strategy prediction is an important and challenging task worth further study in future work.

Speaker	Utterance
Tutor	Hi Ayisha, please talk me through your solution (generic)
Student	I started by noting I concluded that the desk cost her \$350, since that was her winning bid.
Student	Yes, you're right...Carmen's bid added an additional \$150, making the total \$200 + \$150 + \$150 = \$350.
Tutor	Check your calculation of \$200 + \$150 + \$150 = \$350. Your total is not correct (probing)

Table 1: A tutor-student dialogue from MathDial, showing annotated moves for tutor turns.

2 Methodology

In this section, we detail the four primary tasks we use to study tutor strategy and student outcome prediction, as well as the various LLM and non-LLM methods we evaluate on these tasks. First, we define our notation. Each tutor-student dialogue, $D = \{T_i, S_i, M_i, E_i, C\}_{i=1}^N$, contains a sequence of alternating textual student utterances S_i and tutor utterances T_i . Each tutor turn is associated with one or more categorical moves, i.e., granular pedagogical actions, M_i , and each student turn is optionally associated with a binary measure of success E_i . Each dialogue is also labeled with a final binary measure of success, C . As our focus is to study tutor strategy and student outcomes, rather than student behavior, we do not use student move labels. We show an example dialogue in Table 1.

2.1 Tasks

We now detail our four primary tasks. First, we examine future tutor move prediction to investigate if models can predict tutoring strategy. Second, we examine tutor move classification to investigate if models can infer moves from tutor utterances. Third, we examine future dialogue success prediction to investigate if models can infer the outcome of a dialogue from limited context. Finally, we examine next student turn success prediction to investigate if models can predict short-term student outcomes. We formalize these tasks as follows:

$$\hat{M}_{i+1} = f_{\theta}(\{T_j, S_j, M_j\}_{j=1}^i) \quad (1)$$

$$\hat{M}_i = f_{\theta}(\{T_j, S_j, M_j\}_{j=1}^{i-1}, T_i, S_i) \quad (2)$$

$$\hat{C}_i = f_{\theta}(\{T_j, S_j, M_j\}_{j=1}^i) \quad (3)$$

$$\hat{E}_{i+1} = f_{\theta}(\{T_j, S_j, M_j\}_{j=1}^i) \quad (4)$$

2.2 LLM-Based Methods

We evaluate tutor move and student outcome prediction using two large language model (LLM)-based methods. First, we fine-tune the **Llama 3.2 3B** model using Low-Rank Adaptation (LoRA) (Saari et al., 2018), where we train the model to predict the labels as text tokens following the input, using comma separation for multi-label turns. Second, we use zero-shot prompting with **GPT-4o**, where we prompt the model to follow the annotation schema in each particular dataset to identify and predict tutor moves. We do not use GPT-4o for student outcome prediction because pre-trained LLMs do not show high alignment with student behavior (Liu et al., 2025). We show an example prompt in Figure 7, and provide further implementation details in Appendix A. In order to determine how tutor move information impacts LLM predictions, for both methods, we experiment with including only the dialogue as input, as well as both the dialogue and previous turn move labels as input.

2.3 Baselines

We additionally experiment with three traditional baselines, where the input space only uses the move labels of previous turns. First, we employ a second-order **Markov Chain** (Boyer et al., 2009), which estimates the probability of a tutor move or student success given the two preceding moves. Next, we employ **Logistic Regression**, using the frequency distribution of moves up to the current turn as input features. Finally, in order to capture temporal dependencies beyond adjacent moves, we employ an **LSTM** model on sequences of tutor move types encoded as multi-hot vectors.

We do not use these baselines for current move identification since they do not process text. Additionally, we do not use Markov Chain and Logistic Regression for predicting future moves for AlgebraNation because it is a multi-label task; the input space for Markov Chain would be exponentially large, and Logistic Regression would suffer from an overwhelming amount negative labels per class.

3 Experiments

3.1 Datasets

We experiment with two math tutoring datasets, MathDial (Macina et al., 2023) and AlgebraNation (Lyu et al., 2024), to answer our research questions. MathDial contains one-on-one dialogues where a tutor guides a student through solving multi-step

Model	MathDial				AlgebraNation			
	Tutor Move		Future Tutor Move		Tutor Move		Future Tutor Move	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Markov Chain	–	–	42.27	<u>42.81</u>	–	–	–	–
Logistic Regression	–	–	44.69	34.58	–	–	–	–
LSTM	–	–	<u>46.19</u>	31.13	–	–	3.36	23.17
GPT-4o - Dialogue	49.56	48.76	36.48	32.49	<u>79.22</u>	54.30	<u>65.72</u>	27.32
GPT-4o - Dialogue & Moves	50.07	<u>49.03</u>	31.72	29.00	83.82	57.40	69.86	<u>26.33</u>
Llama 3 - Dialogue	<u>52.58</u>	45.88	42.74	32.92	58.82	<u>62.78</u>	24.53	20.20
Llama 3 - Dialogue & Moves	59.69	57.26	50.35	49.33	63.42	68.90	25.69	21.09

Table 2: Results for identifying tutor moves and predicting future tutor moves. Llama 3 performs best on MathDial, while GPT-4o generally performs best on AlgebraNation.

math reasoning problems. The student utterances in this dataset are simulated by an LLM, prompted to mimic common student misconceptions. Each tutor turn is labeled with one of four moves: *probing*, *focus*, *telling*, and *generic*. We leverage turn-level student correctness labels from (Scarlatos et al., 2025a). Our final dataset contains 2,484 dialogues with a 1,947/537 train/test split.

AlgebraNation contains discourse from an online forum where students pose questions and discuss with both tutors and peers. Each tutor turn is labeled with any number of 16 move types. Each post on the forum is marked with success if the responses resolve the original student question. The dataset contains 2,318 forum posts, which we split into a 1,854/464 train/test split. We show label distributions for both datasets in Appendix C.

3.2 Evaluation Metrics

We employ two widely used metrics: i) accuracy (**Acc.**), the portion of predicted labels that match the ground truth, and ii) weighted **F1**, the harmonic mean of precision and recall, weighted by label frequency to account for imbalanced label distributions. For AlgebraNation, where each tutor turn may have multiple moves, we use exact match across all moves in a turn to compute accuracy.

3.3 Tutor Move Prediction

Quantitative Analysis We show the results for tutor move prediction Table 2. Across all methods, predicting the next tutor move proves to be more difficult than classifying the current move, particularly for AlgebraNation, which contains real-world interactions and more granular move definitions. LLMs improve over baselines for future move prediction, showing the importance of textual context and powerful models for this task. However, the F1 for future move prediction is low overall, only

reaching 49% for MathDial and 27% for AlgebraNation. These results indicate that tutor behavior is highly unpredictable, and that even state-of-the-art LLMs struggle to predict future tutor moves.

Additionally, the results across models and datasets are inconsistent; Llama 3 performs better on MathDial while GPT-4o performs better on AlgebraNation. Notably, unlike the other methods, GPT-4o was not trained on AlgebraNation, which exhibits a highly skewed label distribution (Figure 4). This imbalance may help explain why Llama 3 tends to default to predicting the majority class in the future move prediction task (Figure 2), a pattern not observed with GPT-4o. On the other hand, GPT-4o’s move prediction on MathDial likely suffers from confusion between label definitions, as we discuss in the qualitative analysis.

For the move identification task, Llama 3 performs best on MathDial, with the inclusion of annotated move labels significantly increasing performance. Similar to future move prediction, we attribute Llama 3’s higher performance to GPT-4o’s confusion between labels. For AlgebraNation, GPT-4o outperforms Llama 3 in accuracy but underperforms it in F1. This disparity can be explained by observing that Llama 3’s output distribution more closely resembles the ground truth distribution, as seen in Figure 1.

Qualitative Analysis We examine label misclassifications to investigate error patterns in model predictions, revealing dataset-specific challenges. For move classification, in MathDial, confusion frequently arises between *probing* and *focus* moves, while in AlgebraNation, *giving instruction* is often mistaken for *giving explanation* (Table 4). These misclassifications are likely attributable to conceptual overlap in the definitions of these categories, underscoring the nuanced nature of interpreting tu-

Model	MathDial				AlgebraNation	
	Turn Success		Dialogue Success		Dialogue Success	
	Acc.	F1	Acc.	F1	Acc.	F1
Markov Chain	53.75	52.86	53.27	49.01	62.99	63.55
Logistic Regression	52.80	<u>52.54</u>	64.05	66.43	69.30	63.24
LSTM	52.20	<u>52.27</u>	79.13	69.90	<u>77.98</u>	<u>77.76</u>
Llama - Dialogue	<u>63.56</u>	50.54	79.21	70.11	75.64	75.52
Llama - Dialogue & Moves	64.11	49.27	<u>79.16</u>	<u>69.96</u>	81.00	80.84

Table 3: Results for predicting student outcomes from dialogues. Baselines are competitive on MathDial, while Llama 3 performs best on AlgebraNation.

tor intentions. The MathDial authors also note that annotators had difficulty differentiating between *probing* and *focus* moves (Macina et al., 2023). On the other hand, the AlgebraNation annotation guidelines are more specific and instruction-driven, likely helping GPT-4o’s performance due to its ability to generalize well under clear, directive annotation schemes (OpenAI, 2024a). Notably, misclassifications decrease when observing previous move labels, reflected in Table 2, with these labels likely acting as informative in-context examples.

3.4 Student Outcome Prediction

Quantitative Analysis We show the results for student outcome prediction in Table 3. Notably, both Llama 3 and LSTM are able to achieve high performance on dialogue success prediction for both datasets, indicating the tractability of predicting near-term student outcomes in dialogues. However, we note that the MathDial results are inflated as they reflect majority class prediction, as shown in Figure 3. On the other hand, the distribution is more balanced for AlgebraNation, indicating that the outcomes of real students are more reliably predicted than the outcomes of simulated ones. We also see that previous move labels improve performance for Llama 3, showing that tutor moves complement dialogue text to infer student outcomes.

Predicting student outcomes at the turn-level proves to be more difficult than at the dialogue-level in MathDial, with baselines performing close to random chance. However, using LLMs improves performance on this task, capturing nuanced details in the dialogue text to help predict student behavior, as noted in (Scarlatos et al., 2025a).

Regression Analysis To investigate the impact of tutor moves on student outcomes, we examine the learned coefficients of our logistic regression model when predicting dialogue-level success and perform a Chi-squared analysis, shown in Tables

8 and 9. For AlgebraNation, *confirmatory feedback*, *giving instruction*, and *giving explanation* have the greatest positive impact on success. These tutor moves share a common thread: they are all instructionally supportive behaviors that actively guide the student’s understanding or progress. Each move either reinforces correct reasoning (confirmatory feedback), clarifies procedural steps (giving instruction), or deepens conceptual understanding (giving explanation). This correlation suggests that successful dialogues are those in which tutors take an active and supportive role in scaffolding the student’s learning process. For MathDial, *generic* and *probing* have the strongest positive impact on success, whereas *telling* has a negative impact on success. This finding aligns with prior work (Berghmans et al., 2014) showing that facilitative peer tutoring is more effective than directive tutoring.

4 Conclusion

In this work, we investigate the abilities and limitations of LLMs in classifying and predicting tutoring strategies and student outcomes in dialogues. We find that while results vary across models and datasets, LLMs outperform traditional baselines while still struggling at the task of tutor strategy prediction overall. Additionally, we find that student outcome prediction is tractable for LLMs, with tutor move information improving accuracy. Our findings emphasize the importance and challenges of studying tutor strategy in dialogues, given the impact that such strategies can have on student outcomes. Future work should explore how to improve tutor strategy prediction, potentially using in-context learning (Lee et al., 2024) or reinforcement learning (Li et al., 2024). Additionally, future work should explore how to suggest optimal tutor moves, potentially using reinforcement learning guided by student outcomes (Scarlatos et al., 2025b).

Limitations

We identify several technical and practical limitations of our work. First, the generalizability of our results is constrained by the scope and nature of the datasets. MathDial involves synthetic student responses generated by LLMs, which may not reflect the complexity and variability of authentic student behavior. Conversely, AlgebraNation, while comprising real-world interactions, has a highly imbalanced label distribution that poses challenges for model evaluation. Additionally, our evaluation methodology predominantly relies on exact match accuracy and weighted F1 scores. These standard metrics may not fully capture the nuanced characteristics of our models. Finally, the absence of student move tracking in our current modeling approach may affect the results, as sequential modeling of student behavior could potentially enhance predictive performance.

Acknowledgements

This work is partially supported by Renaissance Philanthropy via the learning engineering virtual institute (LEVI) and NSF grants 2118706, 2237676, and 2341948.

References

- Mark Abdelshiheed, Jennifer K. Jacobs, and Sidney K. D’Mello. 2024. Aligning tutor discourse supporting rigorous thinking with tutee content mastery for predicting math achievement. In *Artificial Intelligence in Education*, pages 150–164, Cham. Springer Nature Switzerland.
- Inneke Berghmans, Lotte Michiels, Sara Salmon, Filip Dochy, and Katrien Struyven. 2014. Directive versus facilitative peer tutoring? a view on students’ appraisal, reported learning gains and experiences within two differently-tutored learning environments. *Learning Environments Research*, 17:437–459.
- 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.
- Kristy Elizabeth Boyer, Eun Young Ha, Michael D Wallis, Robert Phillips, Mladen A Vouk, and James C Lester. 2009. Discovering tutorial dialogue strategies with hidden markov models. In *Artificial Intelligence in Education*, pages 141–148. IOS Press.
- Carnegie Learning. 2024. Livehint overview. Online: <https://support.carnegielearning.com/help-center/math/livehint/article/livehint-overview/>.
- 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 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.
- Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori B 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.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Ananya Ganesh, Martha Palmer, and Katharina Kann. 2021. [What would a teacher do? Predicting future talk moves.](#) In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4739–4751, Online. Association for Computational Linguistics.
- Thomas Huber, Christina Niklaus, and Siegfried Handschuh. 2023. [Enhancing educational dialogues: A reinforcement learning approach for generating AI teacher responses.](#) In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 736–744, Toronto, Canada. Association for Computational Linguistics.
- Khan Academy. 2023. Supercharge your teaching experience with khanmigo. Online: <https://www.khanmigo.ai/>.
- Seanie Lee, Jianpeng Cheng, Joris Driesen, Alexandru Coca, and Anders Johannsen. 2024. [Effective and efficient conversation retrieval for dialogue state tracking with implicit text summaries.](#) 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 96–111, Mexico City, Mexico. Association for Computational Linguistics.

- Hang Li, Tianlong Xu, Jiliang Tang, and Qingsong Wen. 2024. [Knowledge tagging system on math questions via llms with flexible demonstration retriever](#). *Preprint*, arXiv:2406.13885.
- Jionghao Lin, Shaveen Singh, Lele Sha, Wei Tan, David Lang, Dragan Gašević, and Guanliang Chen. 2022. Is it a good move? mining effective tutoring strategies from human–human tutorial dialogues. *Future Generation Computer Systems*, 127:194–207.
- Naiming Liu, Shashank Sonkar, and Richard G. Baraniuk. 2025. [Do llms make mistakes like students? exploring natural alignment between language models and human error patterns](#). *Preprint*, arXiv:2502.15140.
- Bailing Lyu, Chenglu Li, Hai Li, Wangda Zhu, and Wanli Xing. 2024. Explaining technical, social, and discursive participation in online mathematical discussions. *Distance Education*, pages 1–24.
- 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.
- Hunter McNichols and Andrew Lan. 2025. [The studychat dataset: Student dialogues with chatgpt in an artificial intelligence course](#). *Preprint*, arXiv:2503.07928.
- Baptiste Moreau-Pernet, Yu Tian, Sandra Sawaya, Peter Foltz, Jie Cao, Brent Milne, and Thomas Christie. 2024. [Classifying tutor discursive moves at scale in mathematics classrooms with large language models](#). In *Proceedings of the Eleventh ACM Conference on Learning @ Scale, L@S '24*, page 361–365, New York, NY, USA. Association for Computing Machinery.
- 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. Working Paper 27476, National Bureau of Economic Research.
- 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.
- OpenAI. 2024a. [Gpt-4.1 prompting guide](#). https://cookbook.openai.com/examples/gpt4-1_prompting_guide. Accessed: 2025-04-21.
- OpenAI. 2024b. [Hello gpt-4o](#). Accessed: 2025-02-19.
- Mika Saari, A Muzaffar bin Baharudin, Pekka Sillberg, Sami Hyrynsalmi, and Wanglin Yan. 2018. Lora—a survey of recent research trends. In *2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0872–0877. IEEE.
- Alexander Scarlatos, Ryan S. Baker, and Andrew Lan. 2025a. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th Learning Analytics and Knowledge Conference, LAK 2025, Dublin, Ireland, March 3-7, 2025*. ACM.
- Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025b. [Training llm-based tutors to improve student learning outcomes in dialogues](#). *Preprint*, arXiv:2503.06424.
- 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, Miami, Florida, USA. Association for Computational Linguistics.
- 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.
- 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.
- LearnLM Team, Abhinit Modi, Aditya Srikanth Veerubhotla, Aliya Rysbek, Andrea Huber, Brett Wiltshire, Brian Veprek, Daniel Gillick, Daniel Kasenberg, Derek Ahmed, Irina Jurenka, James Cohan, Jennifer She, Julia Wilkowski, Kaiz Alarakyia, Kevin R. McKee, Lisa Wang, Markus Kunesch, Mike Schaeckermann, Miruna Pîslar, Nikhil Joshi, Parsa Mahmoudieh, Paul Jhun, Sara Wiltberger, Shakir Mohamed, Shashank Agarwal, Shubham Milind Phal, Sun Jae Lee, Theofilos Strinopoulos, Wei-Jen Ko, Amy Wang, Ankit Anand, Avishkar Bhoopchand, Dan Wild, Divya Pandya, Filip Bar, Garth Graham, Holger Winnemoeller, Mahvish Nagda, Prateek Kohar, Renee Schneider, Shaojian Zhu, Stephanie Chan, Steve Yadlowsky, Viknesh Sounderajah, and Yanniss Assael. 2024. [Learnlm: Improving gemini for learning](#). *Preprint*, arXiv:2412.16429.
- 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.

Deliang Wang, Dapeng Shan, Yaqian Zheng, and Gaowei Chen. 2023. Teacher talk moves in k12 mathematics lessons: Automatic identification, prediction explanation, and characteristic exploration. In *Artificial Intelligence in Education*, pages 651–664, Cham. Springer Nature Switzerland.

Rose Wang and Dorottya Demszky. 2024. [Edu-ConvoKit: An open-source library for education conversation data](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 61–69, Mexico City, Mexico. Association for Computational Linguistics.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024a. 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. 2024b. Tutor copilot: A human-ai approach for scaling real-time expertise. *arXiv preprint arXiv:2410.03017*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Stella Xin Yin, Zhengyuan Liu, Dion Hoe-Lian Goh, Choon Lang Quek, and Nancy F. Chen. 2025. [Scaling up collaborative dialogue analysis: An ai-driven approach to understanding dialogue patterns in computational thinking education](#). In *Proceedings of the 15th International Learning Analytics and Knowledge Conference, LAK '25*, page 47–57, New York, NY, USA. Association for Computing Machinery.

A Details on Experimental Setup

For finetuning Llama 3, we perform a hyperparameter search over learning rates [5e-5, 1e-4, 2e-4, 3e-4] and LoRA ranks [4, 8, 16, 32]. For the final training, we set LoRA’s α to 16, LoRA’s rank to 8, batch size to 64 using gradient accumulation, gradient norm clipping to 1.0, and learning rate to 1e-4. We train for 5 epochs using the AdamW optimizer. We use a random 20% percent of dialogues in the train set to use as a validation set for early stopping. We use the meta-llama/Llama-3.2-3B-Instruct model from the Huggingface Transformers library (Wolf et al., 2019) and run all experiments on NVIDIA L40 GPUs.

We prompt GPT-4o with the OpenAI API using a temperature of 0 while setting the maximum tokens to 1000 and response format to JSON.

Logistic regression is implemented using the sklearn library. The model input is the frequency distribution of moves up to the target tutor move.

For the second-order Markov chain, we compute transition matrices by mapping state pairs based on frequency and normalizing them to yield valid probability distributions.

For the LSTM, we encode the sequence of moves as multi-hot vectors. For multi-label prediction, we use positive weighting for each class, calculated using the proportion of instances in each class. We perform a hyperparameter search with hidden dimensions [64, 128, 256, 512], number of layers [2, 3], dropout rates [0.1, 0.3, 0.5], and learning rates [1e-3, 5e-3, 1e-2]. For multi-label classification, we also search for the optimal probability threshold. Our final models were trained with hidden dimensions 128, 2 layers, dropout of 0.3 and learning rate of 0.001. The learned threshold for multi-label classification is 0.85. We implement the LSTM using Pytorch.

B Misclassification Analysis

Llama 3 Finetuned on Tutor Move Classification					
Without Previous Tutor Moves			With Previous Tutor Moves		
Ground Truth	Predicted	Count	Ground Truth	Predicted	Count
MathDial					
probing	focus	572	probing	focus	339
telling	focus	143	telling	focus	136
focus	probing	81	focus	probing	124
focus	telling	78	generic	focus	69
generic	focus	46	focus	telling	57
AlgebraNation					
giving_explanation	giving_instruction	73	giving_explanation	giving_instruction	74
giving_instruction	giving_explanation	40	giving_instruction	questioning	29
managing_discussions	asking_for_elaboration	34	asking_for_elaboration	questioning	24
questioning	giving_instruction	34	managing_discussions	questioning	23
giving_instruction	questioning	27	giving_instruction	giving_explanation	23

Table 4: Top 5 misclassifications for MathDial and AlgebraNation with Llama 3, comparing inputs with vs. without previous tutor moves. The most common confusion in MathDial is between *focus* and *probing*. The most common confusion in AlgebraNation is between *giving instruction* and *giving explanation*. Total misclassifications decrease by including previous tutor moves.

GPT-4o on Tutor Move Classification					
Without Previous Tutor Moves			With Previous Tutor Moves		
Ground Truth	Predicted	Count	Ground Truth	Predicted	Count
MathDial					
probing	focus	400	probing	focus	431
focus	probing	274	focus	probing	218
focus	telling	169	focus	telling	180
telling	focus	69	telling	focus	63
telling	probing	60	probing	telling	61
AlgebraNation					
giving_instruction	giving_explanation	95	giving_instruction	giving_explanation	79
giving_explanation	giving_instruction	50	giving_explanation	giving_instruction	44
giving_instruction	correcting	37	encouraging_peer_tutoring	praising_and_encouraging	44
encouraging_peer_tutoring	praising_and_encouraging	37	confirmatory_feedback	praising_and_encouraging	40
confirmatory_feedback	praising_and_encouraging	36	asking_for_elaboration	questioning	38

Table 5: Top 5 misclassifications for MathDial and AlgebraNation with GPT-4o, comparing inputs with vs. without previous tutor moves. The most common confusion in MathDial is between *focus* and *probing*. The most common confusion in AlgebraNation is between *giving instruction* and *giving explanation*.

Llama 3 on Future Tutor Move Prediction					
Without Previous Labels			With Previous Labels		
Ground Truth	Predicted	Count	Ground Truth	Predicted	Count
MathDial					
probing	focus	736	probing	focus	378
telling	focus	335	telling	focus	175
generic	focus	252	generic	focus	125
focus	telling	78	focus	telling	121
probing	telling	64	focus	probing	119
AlgebraNation					
questioning	giving_instruction	105	giving_explanation	giving_instruction	162
giving_explanation	giving_instruction	88	questioning	giving_instruction	131
giving_instruction	giving_explanation	82	confirmatory_feedback	giving_instruction	95
confirmatory_feedback	giving_instruction	65	providing_further_references	giving_instruction	79
managing_discussions	giving_instruction	65	managing_discussions	giving_instruction	78

Table 6: Top 5 misclassifications for MathDial and AlgebraNation with Llama 3, comparing inputs with vs. without previous tutor moves. The most common confusion in MathDial is between focus and probing. Previous labels decrease the total top five misclassifications for MathDial. The most common misclassifications in AlgebraNation all occur when *giving instruction* is predicted.

GPT-4o in Future Tutor Move Prediction					
Without Previous Labels			With Previous Labels		
Ground Truth	Predicted	Count	Ground Truth	Predicted	Count
MathDial					
focus	probing	551	focus	probing	553
probing	focus	311	probing	focus	309
telling	probing	288	telling	probing	283
generic	focus	176	telling	focus	178
telling	focus	173	generic	focus	170
AlgebraNation					
giving_instruction	giving_explanation	144	giving_instruction	giving_explanation	144
giving_explanation	giving_instruction	66	confirmatory_feedback	giving_explanation	62
confirmatory_feedback	giving_explanation	66	questioning	giving_explanation	56
questioning	giving_instruction	65	giving_instruction	confirmatory_feedback	54
questioning	giving_explanation	61	questioning	giving_instruction	54

Table 7: Top 5 misclassifications for MathDial and AlgebraNation with GPT-4o, comparing inputs with vs. without previous tutor moves. The most common confusion in MathDial is between *focus* and *probing*. The most common confusion in AlgebraNation is between *giving instruction* and *giving explanation*. Overall, including previous tutor moves decreases the top 4 misclassifications rates.

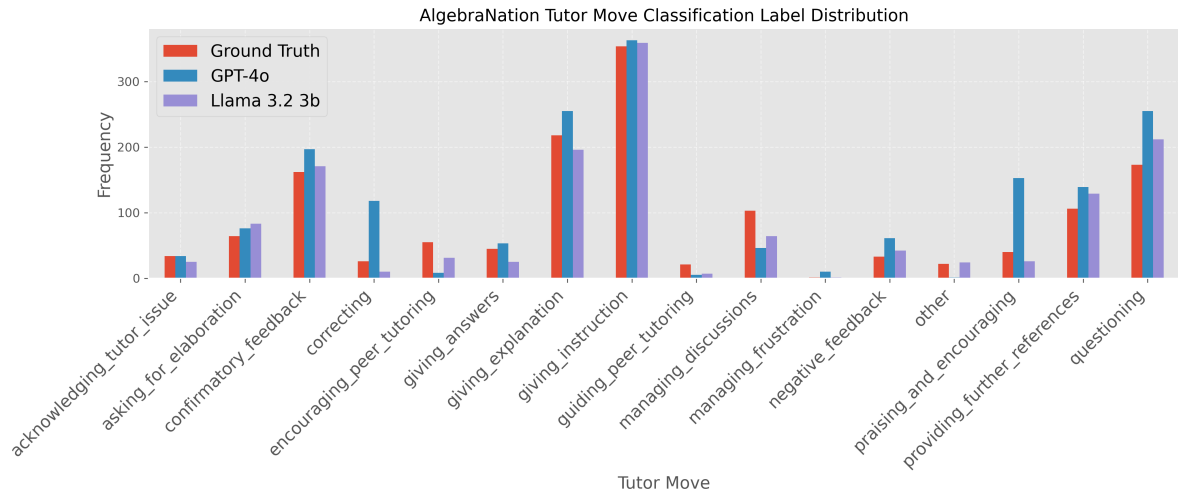


Figure 1: Label distribution of tutor move classification for GPT-4o and Llama 3 trained with dialogue and tutor moves.

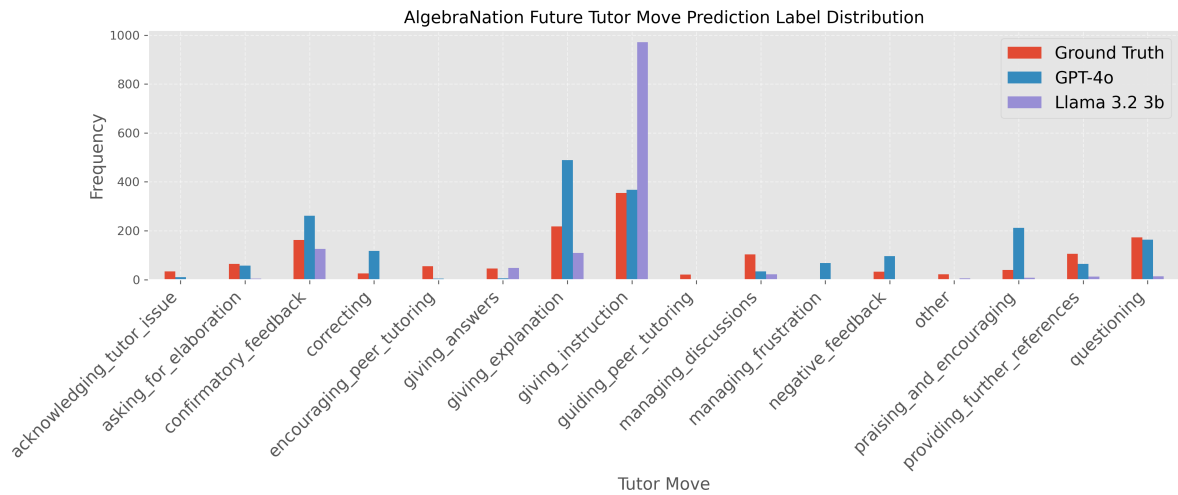


Figure 2: Label distribution of future tutor move prediction for GPT-4o and Llama 3 trained with dialogue and tutor moves. Llama 3 predictions are heavily skewed towards *giving instruction*.

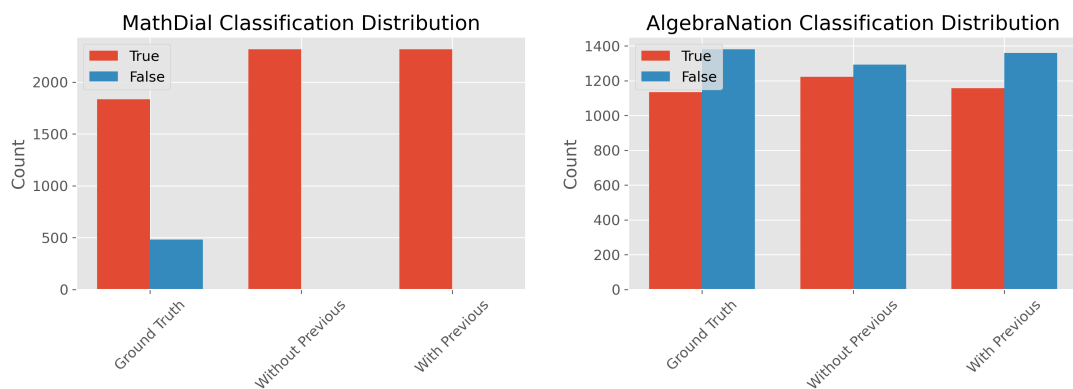


Figure 3: Left: Distribution of dialogue success classification in MathDial using Llama 3. Right: Distribution of dialogue success classification in AlgebraNation using Llama 3.

C Label Distributions

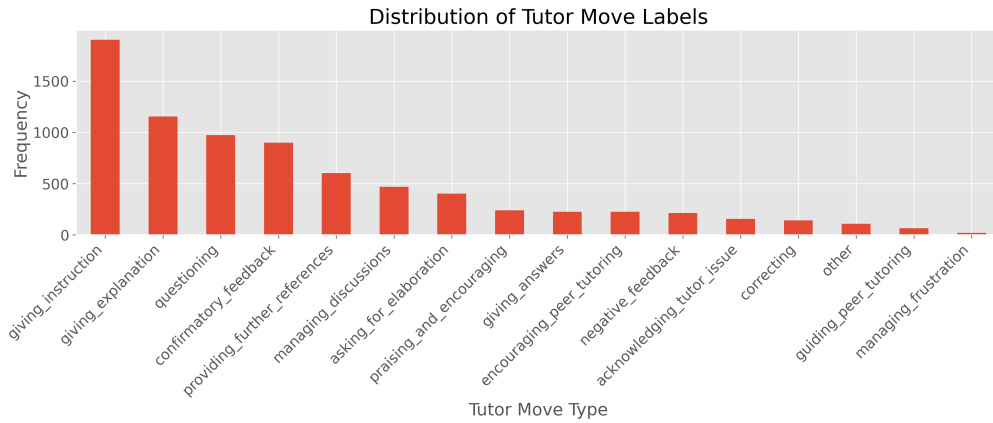


Figure 4: Tutor move distribution of AlgebraNation dataset. Few classes make up the majority of the distribution.

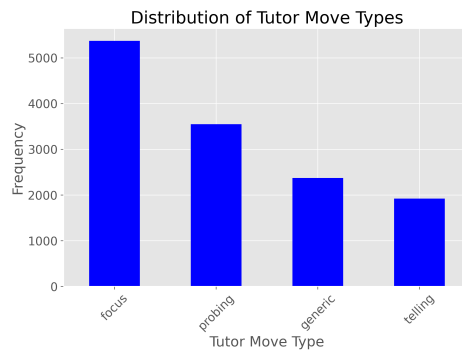


Figure 5: Tutor move distribution of MathDial dataset.

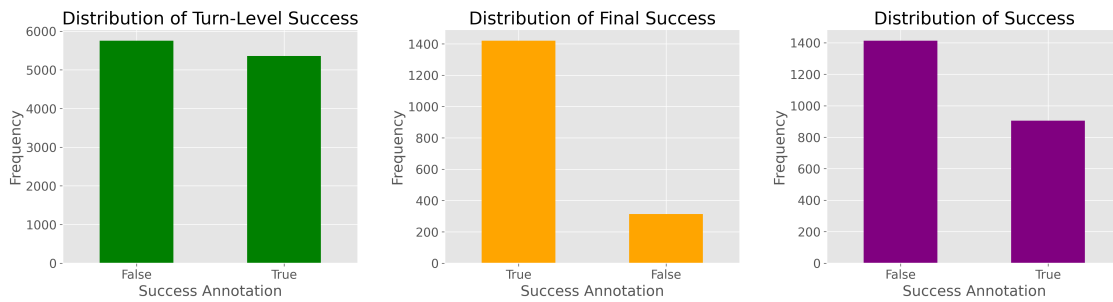


Figure 6: Left: Distribution of turn-level student success in MathDial. Center: Distribution of final turn student success in MathDial. Right: Distribution of dialogue success in AlgebraNation.

D Logistic Regression Coefficients and Chi-Squared Analysis

Feature	Coefficient	χ^2	p-value	Significant
confirmatory feedback	0.168682	684.413557	7.330488e-151	Yes
giving instruction	0.144908	505.315750	6.628000e-112	Yes
giving explanation	0.130397	408.828643	6.593282e-91	Yes
praising and encouraging	0.089068	189.507614	4.072012e-43	Yes
giving answers	0.072106	123.889276	8.907816e-29	Yes

Table 8: Logistic regression coefficients and Chi-squared analysis conducted to evaluate the impact of tutor moves on final dialogue correctness, with corresponding p-values for the top 5 significant features for AlgebraNation.

Feature	Coefficient	χ^2	p-value	Significant
generic	1.063463	18.7393	1.4986e-05	Yes
focus	0.175187	2.2684	1.3203e-01	No
probing	-0.186663	6.7967	9.1324e-03	Yes
telling	-1.062664	19.3447	1.0912e-05	Yes

Table 9: Logistic regression coefficients and Chi-squared analysis conducted to evaluate the impact of tutor moves on final dialogue correctness, with corresponding p-values for all features for MathDial.

E Prompt Examples for Tutor Move Classification

Prompt Used for Classification Task (MathDial)

System: You are a math teacher who tutors student on a variety of problems.

Dialogue ID: 14

Task: Classify the last teacher move into one of the following four categories: *Focus, Probing, Telling, Generic*. These four categories are defined below:

- **Focus**

- Seek Strategy: *So what should you do next?*
- Guiding Student Focus: *Can you calculate ... ?*
- Recall Relevant Information: *Can you reread the question and tell me what is ... ?*

- **Probing**

- Asking for Explanation: *Why do you think you need to add these numbers?*
- Seeking Self Correction: *Are you sure you need to add here?*
- Perturbing the Question: *How would things change if they had ... items instead?*
- Seeking World Knowledge: *How do you calculate the perimeter of a square?*

- **Telling**

- Revealing Strategy: *You need to add ... to ... to get your answer.*
- Revealing Answer: *No, he had ... items.*

- **Generic**

- Greeting/Farewell: *Hi ..., how are you doing with the word problem? Good Job! Is there anything else I can help with?*
- General Inquiry: *Can you walk me through your solution?*

Instructions: Given the student-teacher dialogue below, classify the final teacher move. Return a JSON object with the dialogue ID as the key and the teacher move type as the value. For example:

```
{"1234": "focus"}
```

Dialogue:

[BEGIN DIALOGUE]

Teacher Turn 0: Hi Ayisha, please talk me through your solution (generic)

Student Turn 0: I started by noting that the first bid was \$200 and each subsequent bid was \$50 more, so the second bid was $\$200 + \$50 = \$250$, the third bid was $\$250 + \$50 = \$300$, and the fourth bid was $\$300 + \$50 = \$350$. Then I realized that Carmen bid after each of the other 3 people, so she made 3 bids in total. Therefore, I concluded that the desk cost her \$350, since that was her winning bid.

Teacher Turn 1: Okay, let's start again. If each bid adds \$50 to the price and there were 3 bids - how much is added to the overall cost? (focus)

Student Turn 1: The overall cost is increased by \$150, since each bid adds \$50 and there were 3 bids.

Teacher Turn 2: Yes \$150. Carmen adds a \$50 bid after each of the three bids. How much more does this add to the cost? (focus)

Student Turn 2: This adds an additional \$150, since each bid is \$50 and there were 3 bids. Therefore, the total cost of the desk is \$350.

Teacher Turn 3: 3×50 is 150. So it adds a further \$150 making the total so far \$300 (telling)

Student Turn 3: Yes, that's correct. The total cost of the desk is \$350, since the first bid was \$200 and each subsequent bid was \$50 more, so the final bid was $\$300 + \$50 = \$350$.

Teacher Turn 4: No, go back and check what we said. The 3 bids \$150 + Carmen's Bid \$150 + the opening bid \$200 (focus)

Student Turn 4: Yes, you're right. The total cost of the desk is \$350, since the opening bid was \$200, the 3 bids added \$150, and Carmen's bid added an additional \$150, making the total $\$200 + \$150 + \$150 = \350 .

Teacher Turn 5: Check your calculation of $\$200 + \$150 + \$150 = \350 . Your total is not correct

[END DIALOGUE]

Figure 7: Full prompt used to classify the final tutor move in a MathDial dialogue.

Prompt Used for Classification Task (AlgebraNation)

System: You are a math teacher who tutors student on a variety of problems.

Dialogue ID: 10520899

Task: Classify the last tutor move into one or more of the following categories: *confirmatory_feedback*, *negative_feedback*, *correcting*, *giving_instruction*, *giving_explanation*, *providing_further_references*, *questioning*, *asking_for_elaboration*, *praising_and_encouraging*, *managing_frustration*, *managing_discussions*, *giving_answers*, *encouraging_peer_tutoring*, *guiding_peer_tutoring*, *acknowledging_tutor_issue*, and *other*. These categories are described below:

- **confirmatory_feedback:** Whether a reply provides confirmatory feedback about an answer's correctness.
- **negative_feedback:** Whether a reply states that an answer is incorrect.
- **correcting:** Whether a reply addresses errors in the student's problem-solving approach.
- **giving_instruction:** Whether a reply breaks down a task, performs a part, or initiates a task for the student to complete.
- **giving_explanation:** Whether a reply explains concepts, principles, or provides additional information.
- **providing_further_references:** Whether a reply includes additional resources or references related to the topic.
- **questioning:** Whether a reply asks questions to stimulate thought or constructive discussion.
- **asking_for_elaboration:** Whether a reply requests further details or explanation from the student.
- **praising_and_encouraging:** Whether a reply praises or encourages the student for their efforts or successes.
- **managing_frustration:** Whether a reply addresses the student's negative emotions or frustration.
- **managing_discussions:** Whether a reply organizes the flow of discussion or adjusts the direction of inquiry.
- **giving_answers:** Whether a reply directly provides an answer to the posed question.
- **encouraging_peer_tutoring:** Whether a reply promotes tutoring interactions among peers.
- **guiding_peer_tutoring:** Whether a reply provides feedback on peer tutoring interactions.
- **acknowledging_tutor_issue:** Whether the tutor expresses uncertainty in their reply.
- **other:** Binary indicator for tutoring strategies not classified under the existing labels.

Instructions: Given the student-teacher dialogue below, classify the final teacher move. Return a JSON object with the dialogue ID as the key and the teacher move type(s) as the value. For example:

```
{"1234": ["confirmatory_feedback", "correcting"]}
```

Dialogue:

[BEGIN DIALOGUE]

Student: Can someone help me?

Student: You have to plug in zeros for x and y right

Tutor: Get it into $y=mx+b$ form. ['giving_instruction']

Tutor: SO bring $6x$ to the right side first. ['giving_instruction']

Teacher: Okay Patrice, you want to put that in slope-intercept form ['giving_instruction']

Student: $-5y=30+6x$

Teacher: Now isolate y ['giving_instruction']

Student: -5 on both sides ?

Student: or divide

Teacher: You would divide

[END DIALOGUE]

Figure 8: Full prompt used to classify the final tutor move in an AlgebraNation dialogue.

Assessing Critical Thinking Components in Romanian Secondary School Textbooks: A Data Mining Approach to the ROTEX Corpus

Mădălina Chitez¹, Liviu P. Dinu^{2,3}, Marius Micluța-Câmpeanu⁴

Ana-Maria Bucur⁴, Roxana Rogobete¹

¹West University of Timișoara, Romania

²Faculty of Mathematics and Computer Science, ³HLT Research Center

⁴Interdisciplinary School of Doctoral Studies, University of Bucharest, Romania

{madalina.chitez,roxana.rogobete}@e-uvt.ro, ldinu@fmi.unibuc.ro

marius.micluta-campeanu@unibuc.ro, ana-maria.bucur@drd.unibuc.ro

Abstract

This paper presents a data-driven analysis of Romanian secondary school textbooks through the lens of Bloom’s Taxonomy, focusing on the promotion of critical thinking in instructional design. Using the ROTEX corpus, we extract and annotate almost 2 million words of Romanian Language and Literature textbooks (grades 5-8) with Bloom-aligned labels for verbs associated with pedagogical tasks. Our annotation pipeline combines automatic verb extraction, human filtering based on syntactic form and task relevance, and manual assignment of Bloom labels supported by in-text concordance checks. The resulting dataset enables fine-grained analysis of task complexity both across and within textbooks and grade levels. Our findings reveal a general lack of structured cognitive progression across most textbook series. We also propose a multi-dimensional framework combining cognitive-level and linguistic evaluation to assess instructional design quality. This work contributes annotated resources and reproducible methods for NLP-based educational content analysis in low-resource languages.

1 Introduction

Critical thinking is a key competence in education, shaping students’ ability to analyze, evaluate, and synthesize information. It refers to cognitive and metacognitive processes that enable individuals to question assumptions, construct arguments, and engage in logical reasoning (Ennis, 1985). These processes include argument evaluation (e.g., identifying sound reasoning and fallacies), metacognition (e.g., self-monitoring of thinking), and epistemic skepticism (e.g., questioning the credibility of sources and claims). However, recent studies indicate that these very capacities may be eroding in the age of generative AI, which has been shown to reduce users’ cognitive effort and reliance on reflective thinking (Lee et al., 2025). In the context of

education, the extent to which textbooks promote critical thinking has been a major research concern, particularly regarding curriculum effectiveness (Paul and Elder, 2007) and the role of instructional materials in fostering higher-order thinking skills (Facione, 1990). Studies suggest that educational texts should challenge students intellectually while being cognitively accessible, following developmental frameworks such as the zone of proximal development (ZPD) (Vygotsky and Cole, 1978).

Research on textbooks as facilitators of critical thinking has traditionally relied on manual content analysis and qualitative coding (Halpern, 1998; Kuhn, 2005). However, recent advances in natural language processing (NLP) and computational text analysis have enabled large-scale automated evaluation of educational materials. NLP-based methods allow for the detection and quantification of critical thinking components by analyzing linguistic, structural, and argumentative features in textbooks (Allen et al., 2015; Graesser et al., 2011). These features include argumentative density (e.g., presence of claims, counterclaims, and rebuttals), discourse coherence (e.g., logical connections between ideas), and syntactic complexity (e.g., sentence structures that require higher cognitive processing) (Crossley and McNamara, 2016). The availability of NLP-driven readability and complexity assessment tools (see Section 3) varies across languages, depending on the availability of annotated corpora and computational models designed to process textbook content.

Building on this context, our study is guided by the following research questions:

RQ1: To what extent do Romanian language textbooks at the secondary level include tasks that support higher-order cognitive processes, as defined by Bloom’s Taxonomy?

RQ2: How are these tasks distributed across grades and chapters, and do they reflect a coherent

pedagogical progression?

RQ3: Which textbook series demonstrate the most effective use of instructional tasks for promoting critical thinking, based on verb-level cognitive classification?

These questions aim to bridge linguistic and cognitive evaluation frameworks through a data-driven analysis of instructional content, contributing both methodological tools and empirical insights to the field of educational NLP.

The paper presents the ROTEX corpus analysis and introduces the computational methodology used to extract, classify, and quantify critical thinking elements in school textbooks.¹ We begin by reviewing related work on critical thinking assessment and educational data mining. This is followed by a description of the corpus and the NLP-based methods used to evaluate cognitive complexity. We then present the results of our analysis, focusing on the patterns found in Romanian school textbooks and their implications for curriculum development. The paper concludes with a discussion on the educational relevance of our findings and the potential for integrating AI-driven tools in textbook evaluation and curriculum design.

2 Related Work

Thinking, in its broadest sense, is an active and deliberate process through which individuals make sense of information (Dewey, 2022), ask relevant and purposeful questions (Nosich, 2005), identify what they do not know (ibid.), and revise their beliefs based on new evidence. Dewey (2022) defines reflective thinking as “active, persistent, and careful consideration of any belief or supposed form of knowledge in the light of the grounds that support it and the further conclusions to which it tends” (p. 6). He further characterizes thought as inherently inferential: “the exercise of thought is, in the literal sense of that word, inference; by it one thing carries us over to the idea of, and belief in, another thing. It involves a jump, a leap, a going beyond what is surely known to something else accepted on its warrant” (p. 26). This view frames thinking not as passive reception, but as a generative act of drawing justified conclusions. These foundational processes represent the basis of more specific forms of thinking, such as critical thinking, which adds a layer

of evaluative and reflective judgment. For example, questioning assumptions (Brookfield, 2011), evaluating arguments (Halpern, 2013), and recognizing knowledge gaps (Nosich, 2005) are central to critical engagement. Brookfield (2011) further defines critical thinking as the intentional effort to uncover hidden reasoning structures and challenge taken-for-granted beliefs. Kahneman (2011) complements this view by highlighting the dual-process nature of thinking: fast, intuitive cognition and slow, deliberate reasoning, both of which influence how individuals analyze and respond to information. Together, these perspectives suggest that critical thinking is not separate from general thinking but represents its most reflective, analytical, and self-aware form. To apply these dimensions of thinking in instructional design, educators have adopted structured cognitive frameworks, with Bloom’s Taxonomy being the most widely used.

2.1 Bloom’s Taxonomy and linguistic research

Bloom’s Taxonomy is a foundational framework in pedagogy that categorizes cognitive learning objectives into six hierarchical levels: *remember*, *understand*, *apply*, *analyze*, *evaluate*, and *create* (Bloom et al., 1956; Anderson and Krathwohl, 2001). These levels provide a systematic approach to designing, analyzing, and evaluating educational materials by addressing varying cognitive demands, from basic recall of facts to complex critical thinking and creative tasks. While its original use was in curriculum development, Bloom’s Taxonomy has since been widely applied in educational research and, more recently, in corpus-based studies.

In corpus research, Bloom’s Taxonomy has played a crucial role in evaluating the complexity of educational texts and tasks. For instance, Oravițan et al. (2023) used the ROTEX corpus of Romanian language textbooks and applied Bloom’s Taxonomy to categorize linguistic features in writing tasks. The authors extracted n-grams and verb patterns to align tasks with taxonomy levels. They found that higher-order levels, such as creation (e.g., write, design), were overrepresented in comparison to mid-level skills like analysis (e.g., compare, analyze). Similarly, Graves (2017) employed Bloom’s Taxonomy to examine how writing assignments across university disciplines vary in their cognitive demands, noting the need for balanced progression across the taxonomy levels.

Bloom’s Taxonomy has guided the extraction

¹The code and the annotated data is available here: <https://github.com/mcmarius/ro-textbook-parser>

and classification of cognitive processes from textual datasets. Tools like Coh-Metrix (Graesser et al., 2011) enable the identification of linguistic markers, such as cohesive devices and argumentation patterns, that correspond to taxonomy levels. Cavdar and Doe (2012) linked writing tasks explicitly with Bloom's Taxonomy to assess critical thinking skills in argumentative assignments, showcasing how cognitive skills can be measured quantitatively in textual corpora.

2.2 Textbook design

A growing consensus among researchers emphasizes the pivotal role of textbook design in cultivating students' critical thinking abilities, with Bloom's Taxonomy serving as a widely endorsed framework for structuring cognitive development in learning materials. The revised taxonomy from Anderson and Krathwohl (2001) explicitly advocates for a hierarchical integration of cognitive processes, ranging from remembering to creating, in the design of instructional materials, highlighting the necessity for a scaffolded progression within and across units. Studies analyzing textbooks across various contexts have consistently found a dominance of lower-order thinking skills (LOTS), raising concerns about insufficient cognitive stimulation. For example, Miyazaki (2024) found that 97.3% of tasks in a widely used Japanese Grade 8 textbook fell into the *remember*, *understand*, or *apply* categories, despite national curriculum reforms encouraging *analyze*, *evaluate*, and *create* levels. A similar imbalance was reported by Riazi and Mosalanejad (2010) in Iranian high school and pre-university English textbooks, where lower-order tasks were prevalent, although pre-university materials showed a modest improvement in higher-order inclusion. These findings echo those of Mizbani et al. (2023), who found that in Iran's "Vision 2" textbook, high-order thinking activities were lacking across all four language skills, undermining deeper learning opportunities. These studies reinforce the idea that effective textbooks must not only include all levels of Bloom's Taxonomy but must structure tasks progressively within units and increase the proportion of higher-order thinking tasks by grade level. Such recommendations are further supported by global curriculum standards like those in Japan², which now explicitly aim to foster "the ability to think, make judgments, and ex-

²<https://www.mext.go.jp/en/policy/education/overview/index.htm>

press oneself", outcomes achievable only through textbooks that prioritize higher-order cognition. As Beauchamp and Kennewell (2010) argue, materials that fail to challenge students beyond information recall risk reinforcing surface learning, rather than equipping learners with the reasoning and creativity needed in a complex, unpredictable world.

2.3 Multitasking and critical thinking

While complex, layered tasks are often intended to simulate real-world problem solving, research suggests that combining multiple cognitive demands within a single assignment may, in fact, hinder the development of critical thinking. According to Cognitive Load Theory (Sweller, 1988; Van Merriënboer and Sweller, 2005), the human working memory has limited capacity and overloading it with too many simultaneous instructional demands can lead to superficial engagement rather than deep learning. This is especially problematic when tasks require students to *analyze*, *evaluate*, and *create* under strict, multi-part conditions, prompting a "checklist mindset" rather than genuine intellectual exploration (Torrance, 2007). Paul and Elder (2007) argue that critical thinking flourishes under conditions of conceptual clarity and reflective inquiry, conditions undermined when students are forced to meet narrowly defined sub-goals in one task. Perkins (2008) similarly notes that such instructional designs often lead to "fragile knowledge," where students complete tasks without fully internalizing the concepts involved. While the Revised Bloom's Taxonomy encourages progression toward higher-order thinking (Anderson and Krathwohl, 2001), this does not imply simultaneous execution of all levels in a single prompt. On the contrary, effective critical thinking tasks are often those that isolate and deepen one cognitive demand at a time, especially at the create and evaluate levels, where open-ended exploration is most essential.

3 Method

3.1 Corpus

The analyses are based on the ROLAT subset of the ROTEX corpus (Chitez et al., 2024), the Romanian Corpus of School Textbooks, which comprises Romanian Language and Literature textbooks currently used in secondary schools in Romania. Notably, there is limited continuity within individual publishing house series, as only *ArtKlett* provides a complete set of textbooks for grades 5 through 8

(Table 1).

Textbook	5 th grade	6 th grade	7 th grade	8 th grade
ArtKlett	82,249	80,312	95,968	108,918
Booklet	63,416	93,031	-	-
Corint	58,719	85,836	-	93,214
Litera	55,191	67,895	77,288	-
Intuitext	71,279	78,211	85,143	-
CD	47,618	51,801	71,273	-
Press				
EDP	-	57,096	64,126	-
Paralela	-	88,567	99,002	-
45				
Ars	-	-	62,527	-
Libri				
Aramis	-	-	75,734	66,294
Total	1,880,708			

Table 1: ROTEX sub-corpus size (no. words)

3.2 Annotation

Due to the lack of resources with verbs annotated using Bloom’s Taxonomy labels in Romanian, we have annotated the ROTEX corpus with these labels through a multi-step process. The main computational analysis steps are: (1) task extraction, (2) verb extraction, (3) syntactic filtering, and (4) Bloom-level labeling. The processing pipeline is presented in Appendix A, Figure 4).

First, all tasks were extracted from the ROTEX corpus based on two methods: regular expression heuristics and multimodal prompting with Gemini (Team et al., 2023) (the prompt used is detailed in Appendix A, Table 4). Then, tasks found by both methods were deduplicated. Next, verbs were extracted using the spaCy³ POS tagger for Romanian. To target pedagogical intent, verbs were filtered to retain only those in second person singular or plural, typically indicative of task instructions (e.g., *scrieți*, *gândiți*, *comparați*). This filtering was performed either automatically with spaCy morphological features or manually by human reviewers. For the latter, a group of trained students identified task-related verbs by examining in-text concordances.

Next, the remaining verbs were reviewed for accuracy by expert annotators on the research team. Bloom’s Taxonomy labels were assigned based

on a seed list of verbs from existing literature, directly translating verbs from the Bloom Taxonomy Levels (e.g., “analyze” translated to “analizați”) or from didactic expertise. Verbs without automatic label matches were manually annotated by the same group of human raters. In-text concordances were again used to support disambiguation, particularly for verbs potentially associated with multiple Bloom categories. A final expert verification ensured the accuracy of the annotations, resulting in the finalized list of verbs and their corresponding Bloom labels. Table 2 shows the most frequent verbs, identified as recurring across all instructional prompts and tasks within the ROTEX corpus.

A multi-step human validation process was implemented to ensure the reliability of the Bloom-level annotations. After the initial annotation phase by trained student raters, the assigned labels were reviewed by expert members of the research team, who verified the alignment between each verb’s usage in context and its cognitive category. Special attention was given to polysemous verbs and those that could potentially map to multiple Bloom levels. In these cases, in-text concordances were used to assess task intent and clarify ambiguities. Although no formal inter-annotator agreement metric was calculated, the annotation process was iterative and consensus-based, ensuring consistency in labeling and fidelity to both linguistic form and pedagogical function. It is important to note that assigning cognitive categories to verbs in isolation would be reductive; instead, the full task must be taken into account. Therefore, when applying Bloom’s Taxonomy to real educational materials (such as ROTEX), a contextualized approach has to be used, to ensure that the categorization remains pedagogically meaningful and accurately reflects how the tasks are designed to engage students cognitively.

In total, 434 verbs were annotated: 86 were labeled as “analyze,” 84 as “apply,” 47 as “create,” 112 as “understand,” 26 as “remember,” and 79 as “evaluate”. The annotated list of verbs was then compiled at both the chapter and textbook levels to enable macro- and micro-level distribution analysis. The annotated dataset (Bloom-labeled verbs per task) is publicly available via GitHub⁴, under a CC-BY license.

³<https://spacy.io/>

⁴<https://github.com/mcmarius/ro-textbook-parser>

Bloom Taxonomy Level	Task signal phrases
Remember: Recalling facts, concepts, or basic information.	<i>numește, rememorează, reproduceți, rostește, urmărește</i> (EN: name, recall, reproduce, recite, follow)
Understand: Explaining ideas or concepts.	<i>asociază, caută, centralizează, delimitează, descrie, extrage, indică, identifică, menționează, organizează, precizează, recunoaște, selectează, subliniază</i> (EN: associate, search, cluster, delimitate, describe, extract, indicate, identify, mention, organize, specify, recognize, select, underline)
Apply: Explaining ideas or concepts.	<i>adaugă, adresează, alcătuiește, aplică, arată, combină, completează, construiește, demonstrează, exemplifică, folosește, formează, formulează, îmbină, înlocuiește, încadrează, marchează, rezolvă, transformă, valorifică</i> (EN: add, address, compose, apply, show, combine, complete/fill in, build, demonstrate, exemplify, use, form, formulate, merge, replace, frame, label, solve, transform, utilize)
Analyze: Identifying connections between ideas or breaking down a concept.	<i>analizează, aseamănă, caracterizează, comentează, corectează, corelează, definește, desprinde, separă, stabilește, lucrăți pe echipe</i> (EN: analyze, compare, describe/characterize, comment/interpret, revise, correlate, define, distinguish, differentiate, determine, collaborate)
Evaluate: Forming judgments or justifying a decision or opinion.	<i>alege, argumentează, compară, convinge, dezvoltă, discută, documentează, evaluează, interpretează, justifică, motivează, susține, verifică</i> (EN: choose, argue, compare, convince, develop, discuss, research, evaluate, interpret, justify, motivate, support, check)
Create: Producing something new or original.	<i>compune, concepe, confecționează, continuă, desenează, evocă, gândește, imaginează(-ți), închipuie(-ți), prezintă, realizează, redactează, reformulează, rescrie, scrie, transpune</i> (EN: compose, design, make, complete, draw, evoke, think, imagine, envision, present, create, write, rephrase, rewrite, write, adapt)

Table 2: Bloom-taxonomy verb examples

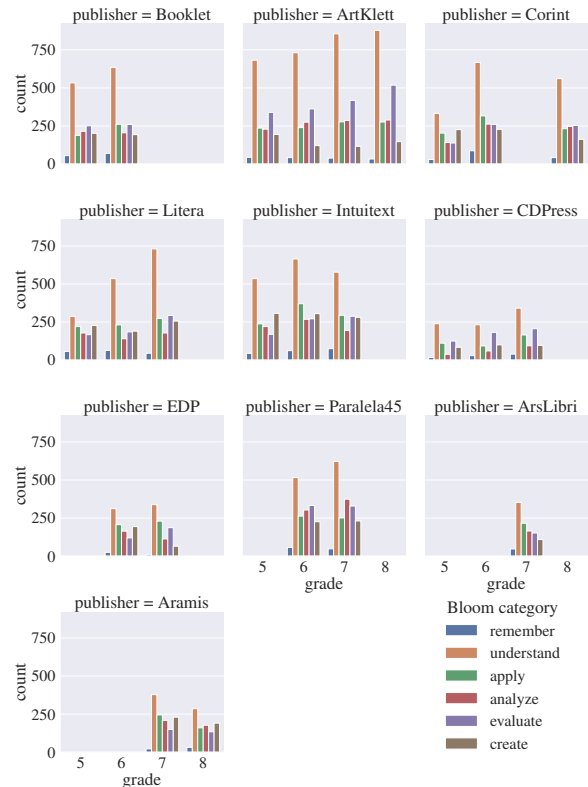


Figure 1: Bloom category counts by publisher and grade

4 Textbooks Analysis

In this section, we present the findings from our analysis of the ROTEX corpus concerning Romanian language textbooks for grades 5 to 8. The analysis indicates a general absence of structured cognitive progression among most publishers. Below, we provide the identified patterns of Task Complexity Distribution (TCD).

4.1 TCD per textbook series

According to Bloom’s Taxonomy, instructional tasks should ideally evolve from foundational cognitive levels, i.e. *remember* and *understand*, toward more complex processes such as *analyze*, *evaluate*, and *create*. However, the distribution of task types across publishers and grades, as shown in Figure 1, suggests minimal scaffolding toward higher-order thinking.

Publishers such as *ArtKlett* exhibit consistently high counts of lower-order tasks (*remember* and *understand*) across all four grades (43.3% or 3304 out of 7626), with surprisingly high rates in higher grades (43.7% or 1805 out of 4132), when the upper-order tasks should prevail. Other textbooks, such as *CD Press* and *EDP*, maintain a near-flat profile across grades, with *understand* tasks mak-

ing up 36% of the content (812 out of 2252), and *create* tasks remaining negligible throughout (12% or 281 out of 2252).⁵

When we look at the growth patterns for higher-order thinking tasks, several trends emerge. *Intuitext* stands out for its clear increase in *analyze* and *evaluate* tasks across grades, and the gradual introduction of *create* tasks by grade 8. *CDPress* and *ArtKlett* also show moderate gains in higher-order categories, especially in the upper grades. *Booklet* and *Paralela45* include a range of task types but lack a clear upward trajectory. In contrast, *Corint*, *Litera*, *ArsLibri*, and *Aramis* remain static, with modest presence of higher-order tasks throughout. *EDP* shows some potential, but lacks full grade coverage.

Based on these findings, a hierarchy of textbooks based on their capacity to promote critical thinking can be established. *Intuitext* ranks highest, showing the most consistent inclusion and progression of *analyze*, *evaluate*, and *create* tasks across grades. *CDPress* follows, with moderate presence of higher-order tasks, especially in upper grades. *ArtKlett* shows some higher-order tasks by grade 8, but lacks consistent instructional sequencing. All other textbooks (*Booklet*, *Paralela45*, *Corint*, *Litera*, *Aramis*, *ArsLibri*) display minimal to no critical thinking tasks, remaining focused on lower-order categories (55% or 10150 out of 18518).

Table 3 shows higher-order tasks included in ROTEX textbooks that exhibit variability and inconsistency regarding Bloom's Taxonomy for several reasons (sequence combining several levels or using multiple verbs and tasks), reflecting the complexity of educational tasks and practices.

4.2 TCD per learning unit within textbooks

By making a fine-grained analysis on the distribution of tasks within each textbook per grade (Appendix A, Figure 5), we can notice similar patterns to the overall task complexity distribution. Among all series, *Intuitext* demonstrates the most coherent and intentional progression of task complexity. Across its chapters, there is a visible build-up from lower-order tasks toward *analyze*, *evaluate*, and even *create*, particularly in the upper grades, reflecting a well-structured approach to competence development. *CDPress* presents a similarly structured pattern, with higher-order tasks becoming more prominent in later chapters, suggesting a

⁵See also Figure 3 for counts shown as percentages (Appendix A).

Task example	Bloom taxonomy level
a) <i>Extrage</i> , din text, enumerația care se asociază peisajului marin.	Understand
b) Cum se raportează instanța lirică la ideea de patrie? <i>Argumentează-ți</i> răspunsul.	Evaluate Create
(EN: a) <i>Extract</i> , from the text, the enumeration associated with the seascape. b) How does the lyric instance relate to the idea of homeland? <i>Give reasons</i> for your answer.)	
Cui <i>consideri</i> că îi aparțin cuvintele așezate între liniile de pauză din versul „– Și Dumnezeu cunoaște cum vorba și-o păzește –”?	Analyze Evaluate
Alege una dintre variantele următoare și <i> motivează-ți</i> opțiunea:	Evaluate
a) personajului, care jură în fața păsărilor domestice pentru a da greutate cuvintelor sale;	
b) naratorului, care intervine astfel spre a avertiza cititorul că personajul minte.	
(EN: Who do you <i>think</i> the words between the pause lines in the line “– And God knows how the word is spoken and keeps it –” belong to? Choose one of the following options and <i>give</i> your reasons:	
a) the character, who swears in front of the domestic birds to give weight to his words;	
b) the narrator, who intervenes to warn the reader that the character is lying.)	
<i>Amintește-ți</i> ultima călătorie pe care ai făcut-o. <i>Formulează</i> enunțuri care să continue următoarele începuturi. . .	Remember Create
(EN: <i>Remember</i> the last trip you took. <i>Make statements</i> that continue the following beginnings. . .)	
<i>Recitește</i> textul Fascinații de George Șovu și <i>notează</i> în caiet o secvență narativă și una descriptivă, <i>precizând</i> ce rol au în cadrul textului.	Remember Understand Evaluate
(EN: <i>Reread</i> the text Fascinations by George Șovu and <i>write down</i> a narrative and a descriptive sequence in your notebook, <i>specifying</i> their role in the text.)	

Table 3: Examples of exercises that exhibit variability and inconsistency regarding Bloom's taxonomy by combining multiple levels in the same task

lightweight but effective progress model. In contrast, *ArtKlett* includes a wide variety of task types, including higher-order ones, in nearly every chapter from the outset. However, the lack of variation or progression across chapters indicates a dense and static design rather than a pedagogically sequenced one. Other textbook series, such as *Booklet*, *Litera*, *Corint*, *Paralela45*, and *Aramis*, show predominantly flat distributions, with chapters consistently focused on lower-order categories like *remember* and *understand*, and little evidence of an intentional cognitive arc. These findings suggest that while some series embed critical thinking tasks, only a few succeed in distributing them progressively and meaningfully throughout the learning units.

4.3 TCD per learning unit within series across grades

A cross-grade analysis of task distribution reveals distinct patterns in how textbook series support cognitive development over time. *Intuitext* is the only series to exhibit a clear upward trajectory in task complexity, with a gradual increase in *analyze*, *evaluate*, and *create* tasks from grades 5 to 7, aligning well with students' developmental stages. *CDPress* also shows moderate progression, with higher-order tasks becoming more prominent in grades 6 and 7, although coverage is limited. In contrast, *ArtKlett* distributes higher-order tasks relatively evenly across all grades, indicating cognitive density without meaningful scaffolding. The remaining series, *Booklet*, *Litera*, *Corint*, *Paralela45*, *Aramis*, and *ArsLibri*, maintain static task profiles, dominated by lower-order categories throughout, with minimal evidence of progression. These results suggest that most textbook series do not implement systematic cognitive progression across grade levels, potentially limiting their effectiveness in supporting long-term competence development.

4.4 Multi-task presence

Since tasks can contain multiple sentences and phrases, it is difficult to assign a single label when multiple verbs correspond to several Bloom levels. This complexity persists despite the annotation efforts detailed in Section 3.2, which aimed to assign a single Bloom level to each verb. In the previous sections, we addressed this ambiguity to assign a single Bloom level per task by dividing each task into individual sentences and then further splitting

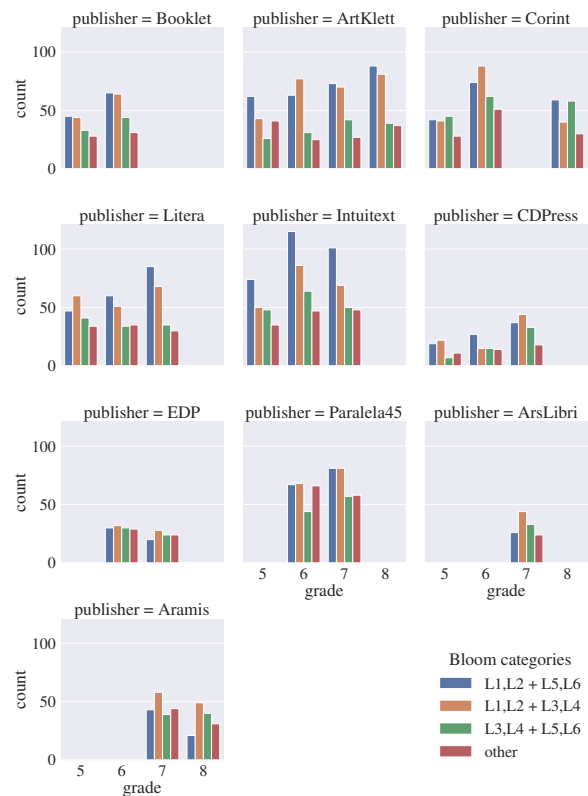


Figure 2: Bloom category counts by publisher and grade for multitask exercises

each sentence based on the main verb identified in the parse tree. Finally, we assigned the label of the first verb in order of appearance.

For this analysis, we keep the labels from all verbs and focus our attention only on these multi-level tasks, which account for 26% of all exercises. After considering tasks at the sentence level, we are left with 15% multi-task exercises. Following the same methodology, we group tasks by textbook series, grades, and learning units (Figure 2). A fine-grained analysis on the distribution of tasks per learning unit within each textbook is presented in Appendix A, Figure 6.

Based on the assumption that a gradual progression would result in multi-task exercises to contain verbs assigned to consecutive Bloom levels, we aggregate these multi-task exercises in four categories: (1) low level + high level, (2) low level + middle level, (3) middle level + high level and (4) any other combination, most likely consisting of all levels. While the number of such tasks is modest (15%), this reveals another concerning trend, once again confirming a general lack of pedagogical principles. Except for *ArsLibri*, all textbooks across all grades mix low levels and high levels in more than half of all multi-task exercises. Fur-

thermore, *ArtKlett*, *Litera* and *Intuitext* exhibit this trend consistently across most grades and chapters.

If we analyze just one example of a task from a textbook designed for the 5th grade (*ArtKlett*), we can justify several challenges: „Select four verbs from the given text and write them on your worksheet. Use them in a composition in which you present a story set in the garden, showing how you make friends with one of the creatures in the text. In your composition, you will fulfill the following requirements: (1) use the four verbs, with the possibility of changing their form; (2) present a story from the garden, showing how you make friends with a creature from the text; (3) give an appropriate title to your composition; (4) follow the structure of a composition: introduction, contents and conclusion; (5) write your essay in at least ten lines; (6) you will write correctly, using correct phrasing, punctuation and layout.”. The assignment asks students to use four verbs in a creative context, which requires higher-order thinking skills such as synthesis (*creating* a story) and *evaluation* (reflecting on their experiences). However, the initial focus on simply selecting and using verbs (*understand*, *apply*) may lead to confusion about the primary cognitive demand. While Bloom’s Taxonomy emphasizes distinct cognitive levels, from remembering to creating, this task blurs those lines by combining various levels without clear differentiation. Moreover, the requirement to “make friends” with a creature introduces an imaginative aspect that can be challenging to evaluate and assess at a cognitive level within Bloom’s Taxonomy. The subjective nature of forming friendships makes it difficult to measure students’ cognitive engagement and understanding effectively. Last, but not least, while structure is essential for effective writing, the task’s emphasis on format (introduction - content - conclusion) may detract from the creative process. A more explicit delineation of the cognitive demands of the instructional tasks, coupled with a more focused pedagogical approach, would facilitate a better achievement of the intended learning outcomes.

5 Discussion and Conclusions

The present analysis provides a systematic view of how Romanian secondary school textbooks promote critical thinking through task design, as operationalized via Bloom’s Taxonomy. This responds directly to the concerns raised in prior research

(Paul and Elder, 2007; Facione, 1990; Halpern, 1998) and discussed in the literature review, where critical thinking was framed as a key educational competence often underrepresented in instructional materials. While previous work by Chitez et al. (2024) focused on the linguistic and structural complexity of these textbooks, highlighting issues such as lexical overload, syntactic density, and redundancy, our current findings complement that perspective by offering a cognitive-level analysis of instructional tasks. Together, both dimensions reveal a misalignment between task complexity and learner accessibility: even when textbooks attempt to include higher-order tasks (*analyze*, *evaluate*, *create*), these are often embedded in overly dense or poorly scaffolded materials, which may negatively impact rather than support competence development (RQ1).

To address RQ2 and RQ3, our results show that only a few textbook series, most notably *Intuitext*, demonstrate a structured progression in cognitive demands across both chapters and grades, reflecting an intentional effort to build students’ reasoning skills over time. In contrast, other series, such as *ArtKlett*, while displaying a large variety of tasks, distribute higher-order activities uniformly, lacking clear instructional sequencing. This confirms earlier concerns (Chitez et al., 2024) that task overload and lack of cognitive pacing may dilute the intended pedagogical impact. Moreover, several series (*Corint*, *Litera*, *Paralela45*, *Aramis*) display static cognitive profiles dominated by lower-order tasks, further reinforcing a pattern of surface-level engagement already observed in their linguistic structures.

These findings reinforce arguments from the literature that effective textbook design must not only include a range of cognitive operations but must also organize them in a way that reflects developmental progression (Anderson and Krathwohl, 2001; Graesser et al., 2011). Critical thinking cannot be effectively sustained by simply including complex verbs or isolated higher-order tasks. These should be embedded within a deliberate instructional sequence and supported by clear, accessible language. For textbook designers, this implies a double perspective: to align task design with students’ cognitive growth and to calibrate language complexity to ensure engagement and understanding.

The combined model of cognitive and linguistic evaluation offers a replicable framework for

assessing instructional quality in educational materials. Future research could extend this methodology by integrating argumentation mining, dialogic task analysis, or learner performance data, with the goal of aligning curriculum design more closely with evidence-based models of competence development.

Finally, the annotated set of Bloom Taxonomy labels for verbs in instructional materials developed in this study provides a valuable resource for NLP research in education. By linking specific verb forms to cognitive processes such as *remember*, *analyze*, or *create*, this dataset enables more granular and automated assessments of instructional intent. Unlike traditional readability metrics, which focus on surface-level linguistic features, Bloom-aligned annotations allow for the identification of pedagogical depth and cognitive demand. This enables a range of novel applications, including automatic task classification, educational question generation, and curriculum alignment modeling, particularly in low-resource educational settings like Romanian. Moreover, integrating Bloom-labeled data into NLP models can enhance the interpretability of text complexity predictions and support the development of AI-driven tools for textbook evaluation, instructional design, and adaptive learning systems.

Limitations

While this study effectively identifies and categorizes the most frequent verbs associated with Bloom's Taxonomy within the ROTEX corpus, it has limitations that must be acknowledged. From a pedagogical standpoint, focusing solely on verbs does not provide a complete picture of the instructional tasks and learning objectives. The meaning and effectiveness of verbs can vary significantly depending on the context in which they are used. A verb might imply different cognitive processes based on the surrounding tasks, objectives, and instructional strategies. Therefore, the rating and categorization process was highly contextual in the case of ROTEX.

In this context, one limitation of this study is that automated keyword matching alone cannot capture the full nuance of instructional language, often resulting in oversimplified cognitive labels. To address this, we applied a contextual annotation strategy that considers how verbs function within the broader pedagogical framing of each

task. This method improves the accuracy of classification by accounting for cases where the intended cognitive process is not clearly expressed through verbs alone. This variation underscores the need for an interpretive approach, and our annotation method responds by capturing how Bloom's Taxonomy operates in real instructional contexts, where taxonomic intent is often implicit.

A separate limitation is the absence of formal inter-annotator agreement scores in the Bloom Taxonomy label annotation process. Although we did not compute quantitative measures of annotation reliability, we addressed annotation reliability by implementing a multi-phase validation pipeline involving trained student raters and subsequent expert review. This layered approach, particularly the involvement of domain experts in the final verification, helped ensure that annotations aligned closely with pedagogical intent and contextual usage. Similarly, while the assignment of a single Bloom Taxonomy level per task, based on the first verb, might appear reductive given the complexity of some prompts, this simplification was necessary for large-scale processing and related automatic disambiguation procedures. The automation is further supported by an additional sentence-level analysis of multi-task exercises. Such choices balance methodological rigor with the practical demands of corpus-level annotation.

Furthermore, although the ROTEX corpus focuses exclusively on Romanian Language and Literature textbooks and includes uneven grade-level representation for some publishers, it still captures the full range of instructional materials currently in use. Finally, while our study does not include learner performance data, it offers a strong foundation for future work linking task design with educational outcomes. By combining linguistic analysis with pedagogical classification, the article effectively contributes a resource and methodology that can inform both textbook development and automated curriculum evaluation.

Acknowledgments

This research is supported by the project "Romanian Hub for Artificial Intelligence - HRIA", Smart Growth, Digitization and Financial Instruments Program, 2021-2027, MySMIS no. 334906 and CNCS/CCCDI UEFISCDI, SiRoLa project, PN-IV-P1-PCE-2023-1701, within PNCDI IV, Romania.

References

- Laura K Allen, Erica L Snow, and Danielle S McNamara. 2015. [Are you reading my mind? Modeling students' reading comprehension skills with natural language processing techniques](#). In *Proceedings of the fifth international conference on learning analytics and knowledge*, LAK '15, pages 246–254, New York, NY, USA. Association for Computing Machinery.
- Lorin W Anderson and David R Krathwohl. 2001. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition*. Addison Wesley Longman, Inc.
- Gary Beauchamp and Steve Kennewell. 2010. [Interactivity in the classroom and its impact on learning](#). *Computers & Education*, 54(3):759–766. Learning in Digital Worlds: Selected Contributions from the CAL 09 Conference.
- Benjamin S Bloom, Max D Engelhart, Edward J Furst, Walker H Hill, David R Krathwohl, and 1 others. 1956. *Taxonomy of educational objectives: The classification of educational goals. Handbook 1: Cognitive domain*. Longman New York.
- Stephen D Brookfield. 2011. *Teaching for critical thinking: Tools and techniques to help students question their assumptions*. John Wiley & Sons.
- Gamze Çavdar and Sue Doe. 2012. [Learning through writing: Teaching critical thinking skills in writing assignments](#). *PS: Political Science & Politics*, 45(2):298–306.
- Madalina Chitez and 1 others. 2024. [Linguistic overload in secondary school textbooks: A corpus-informed case study of Romanian 6th grade textbooks](#). In *Conference Proceedings. Innovation in Language Learning 2024*.
- Scott A Crossley and Danielle S McNamara. 2016. [Text-based recall and extra-textual generations resulting from simplified and authentic texts](#). *Reading in a Foreign Language*, 28:1–19.
- John Dewey. 2022. *How we think*. DigiCat.
- Robert H Ennis. 1985. [A logical basis for measuring critical thinking skills](#). *Educational Leadership*, 43(2):44–48.
- Peter Facione. 1990. [Critical thinking: A statement of expert consensus for purposes of educational assessment and instruction \(The Delphi Report\)](#). *Research Findings and Recommendations*, 315.
- Arthur C Graesser, Danielle S McNamara, and Jonna M Kulikowich. 2011. [Coh-Matrix: Providing multi-level analyses of text characteristics](#). *Educational Researcher*, 40(5):223–234.
- Roger Graves. 2017. *Writing assignments across university disciplines*. Trafford Publishing.
- Diane F Halpern. 1998. [Teaching critical thinking for transfer across domains: Disposition, skills, structure training, and metacognitive monitoring](#). *American Psychologist*, 53(4):449.
- Diane F Halpern. 2013. *Thought and knowledge: An introduction to critical thinking*. Psychology Press.
- Daniel Kahneman. 2011. *Thinking, fast and slow*. Macmillan.
- Deanna Kuhn. 2005. *Education for thinking*. Harvard University Press.
- Hao-Ping Lee, Advait Sarkar, Lev Tankelevitch, Ian Drosos, Sean Rintel, Richard Banks, and Nicholas Wilson. 2025. [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*, pages 1–22.
- Ryo Miyazaki. 2024. [Using revised Bloom's taxonomy to evaluate higher order thinking skills \(HOTS\) in tasks from an 8th grade English language textbook in Japan](#). Master's thesis, Southeast Missouri State University.
- Maryam Mizbani, Hadi Salehi, Omid Tabatabaei, and Mohammadreza Talebinejad. 2023. [Textbook evaluation based on Bloom's revised taxonomy: Iranian senior high school textbook in focus](#). *Language and Translation*, 13(1):85–99.
- Gerald M. Nosich. 2005. *Learning to think things through: A guide to critical thinking across the curriculum (2nd ed.)*. Upper Saddle River, NJ: Pearson/Prentice Hall.
- Alexandru Oravițan, Mădălina Chitez, and Roxana Rogobete. 2023. [A linguistically-informed assessment model for multidimensional competence building in Romanian school writing](#). *Educatia 21*, pages 85–91.
- Richard Paul and Linda Elder. 2007. [Critical thinking: The art of Socratic questioning](#). *Journal of Developmental Education*, 31(1):36.
- David Perkins. 2008. *Smart schools: From training memories to educating minds*. Simon and Schuster.
- A Mehdi Riazi and Narjes Mosalanejad. 2010. [Evaluation of learning objectives in Iranian high-school and pre-university English textbooks using Bloom's taxonomy](#). *Tesl-Ej*, 13(4).
- John Sweller. 1988. [Cognitive load during problem solving: Effects on learning](#). *Cognitive Science*, 12(2):257–285.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. [Gemini: a family of highly capable multimodal models](#). *arXiv preprint arXiv:2312.11805*.

Harry Torrance. 2007. *Assessment as learning? how the use of explicit learning objectives, assessment criteria and feedback in post-secondary education and training can come to dominate learning.* *Assessment in Education: Principles, Policy & Practice*, 14(3):281–294.

Jeroen JG Van Merriënboer and John Sweller. 2005. *Cognitive load theory and complex learning: Recent developments and future directions.* *Educational Psychology Review*, 17:147–177.

Lev Semenovich Vygotsky and Michael Cole. 1978. *Mind in society: Development of higher psychological processes.* Harvard University Press.

A Additional details

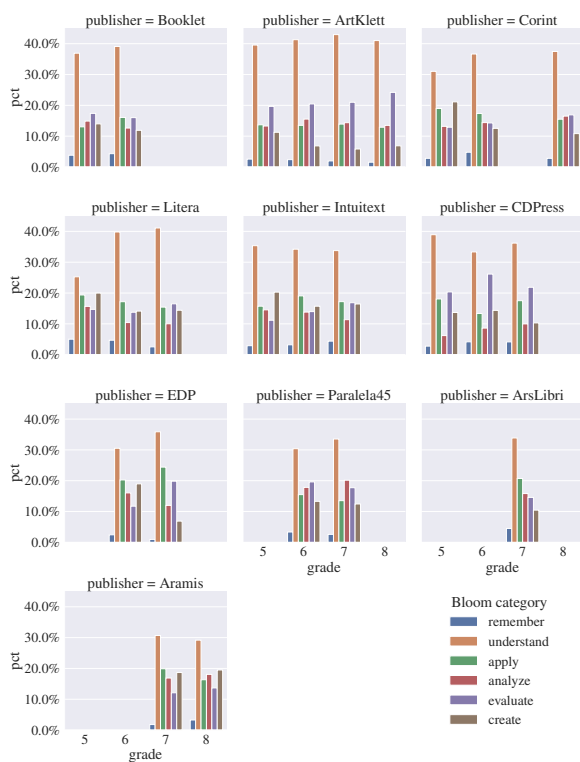


Figure 3: Bloom category percentage counts by publisher and grade

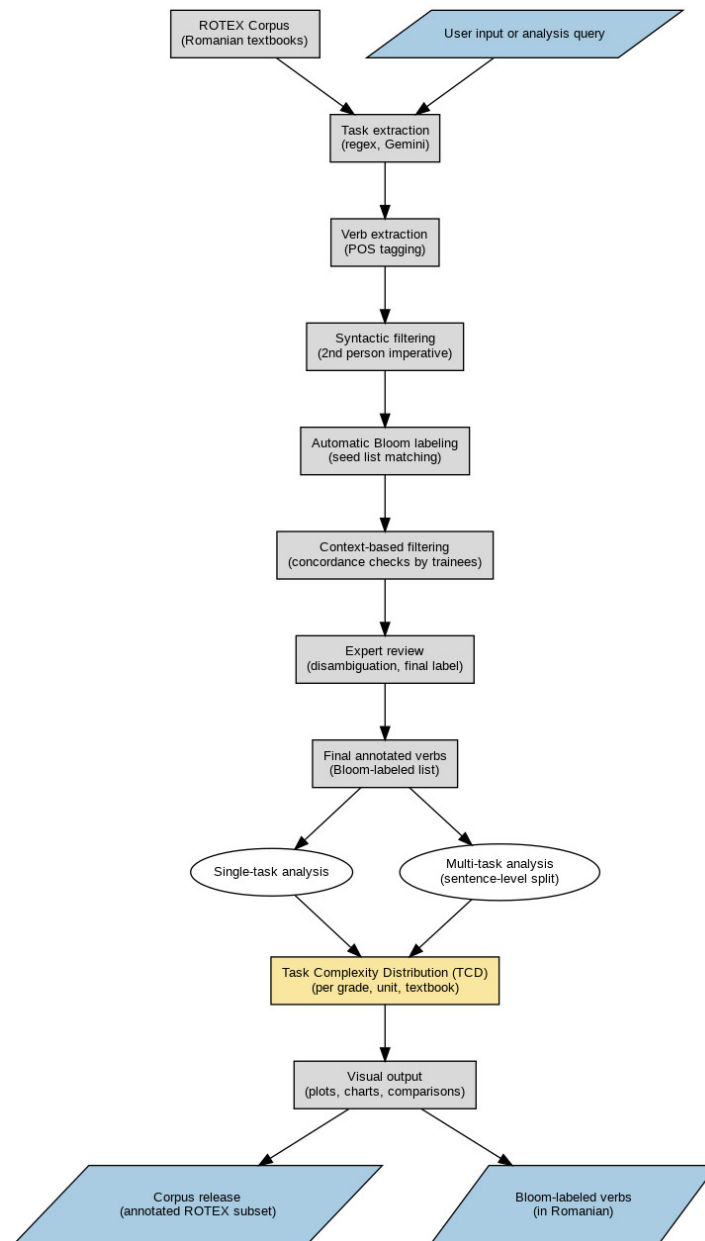


Figure 4: Pipeline for annotating Bloom's Taxonomy labels

Gemini Prompt

You are given a page from a Romanian language and literature textbook written in Romanian.

Extract all exercises from the page. Usually, exercises are numbered. Keep the number of exercises in the structured output.

Do not shorten or summarize the text of the exercises. Use the full text that is presented in the document. Do not remove any sentence from the exercise text. Keep the order of the exercises as they appear on the page. Unite the syllabified words in the exercises.

Make a JSON file of the output. The output JSON should include all the exercises from the file with their full text.

Include the page number and the name of the section that each exercise belongs to in the JSON file. The section name is typically found just before the exercises. If the section name is not provided on the page, leave that field empty. Use an integer to indicate the order of the sections in the document; the first section should be labeled as 1, the second as 2, and so on. There can be multiple sections with the same name in the document.

If the document does not contain any exercise, leave the JSON file empty.

Table 4: Prompt used for extracting tasks from textbooks

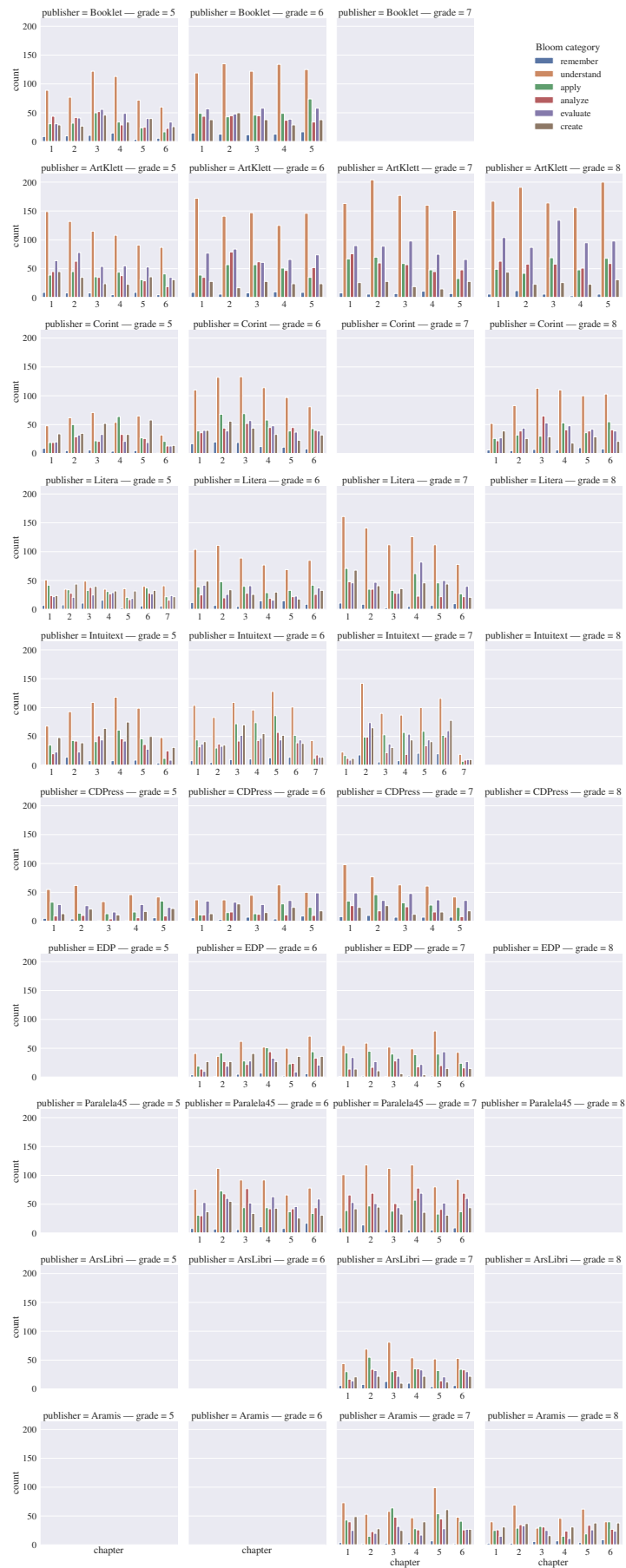


Figure 5: Bloom category counts by publisher, grade, and chapter

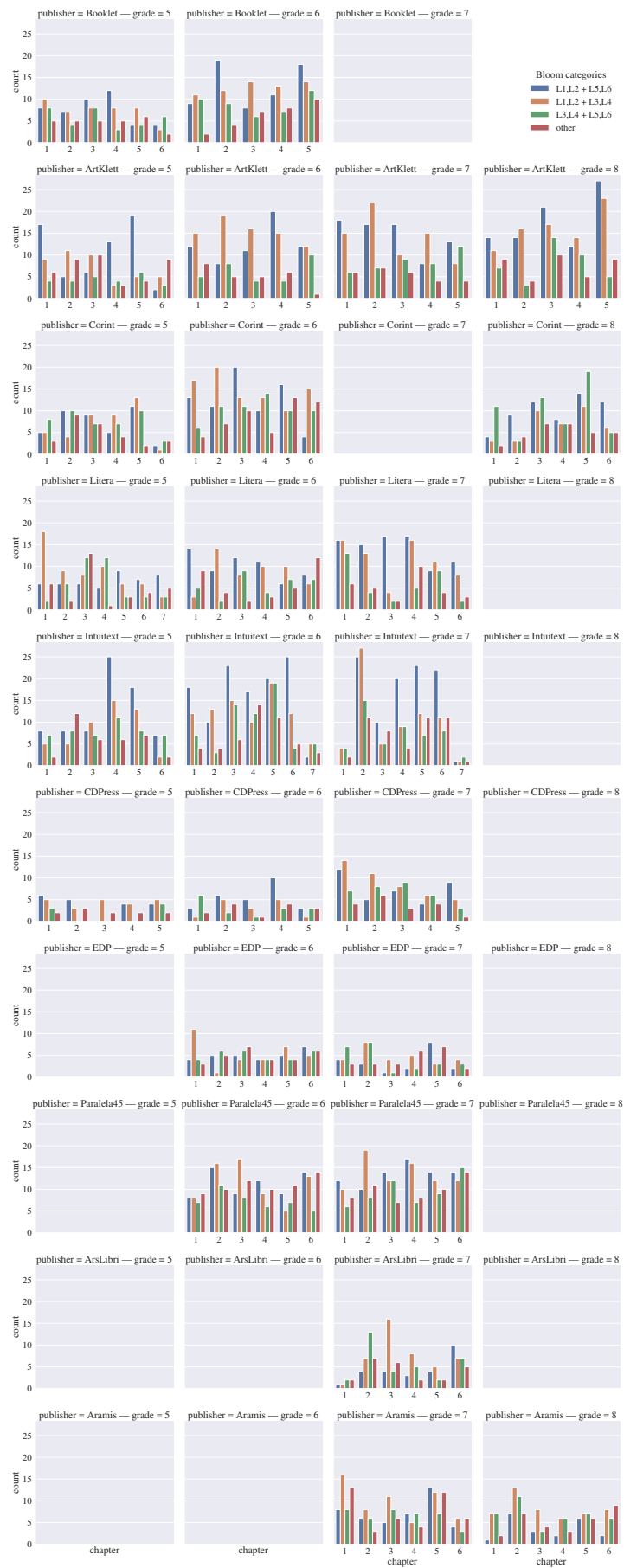


Figure 6: Bloom category counts for multitask exercises by publisher, grade, and chapter

Improving AI assistants embedded in short e-learning courses with limited textual content

Jacek Marciniak, Marek Kubis, Michał Gulczyński
Adam Szpilkowski, Adam Wieczarek, Marcin Szczepański

Faculty of Mathematics and Computer Science
Adam Mickiewicz University, Poznań
ul. Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland
{jacekmar,mkubis,marcin.szczepanski}@amu.edu.pl
{micgull,adaszp,adawiel}@st.amu.edu.pl

Abstract

This paper presents a strategy for improving AI assistants embedded in short e-learning courses. The proposed method is implemented within a Retrieval-Augmented Generation (RAG) architecture and evaluated using several retrieval variants. The results show that query quality improves when the knowledge base is enriched with definitions of key concepts discussed in the course. Our main contribution is a lightweight enhancement approach that increases response quality without overloading the course with additional instructional content.

1 Introduction

AI assistants based on large language models (LLMs) are increasingly used to support learning and access to educational content. Most research in this area assumes access to large-scale textual resources, such as entire textbooks or extensive document collections. To reduce hallucinations and improve grounding, many approaches rely on techniques such as Retrieval-Augmented Generation (RAG; Lewis et al., 2020), where relevant documents are retrieved from a knowledge base and passed to the model at inference time. However, the effectiveness of such methods typically depends on the availability of rich textual input—an assumption that often does not hold in real-world educational contexts.

In practice, modern educational programs often rely on short e-learning modules designed to teach narrowly defined learning objectives within a limited timeframe. These modules—especially in higher education—are intentionally concise to preserve instructional clarity and reduce cognitive load. When AI assistants are embedded in such courses, they are expected to provide accurate, context-aware support without relying on large external corpora or hallucinating irrelevant content.

Despite the growing popularity of LLM-based assistants, there is a lack of research on how to

design such systems when instructional content is minimal. Existing work typically targets high-resource settings, and it remains unclear whether techniques developed for large-scale retrieval transfer effectively to low-resource educational contexts. Moreover, instructors often have limited time and must make strategic decisions about which concepts or materials are worth covering. Expanding materials solely to meet model requirements is pedagogically undesirable.

This paper investigates how to improve the effectiveness of AI assistants embedded in short e-learning courses with limited textual content. Rather than expanding the course, we propose a lightweight enhancement strategy: injecting definitions of key course concepts into the assistant’s knowledge base. We evaluate this approach using a real-world e-learning course on machine learning fundamentals (approx. 30 learning objects) and a benchmark of 94 questions collected from students who completed the course.

Our main contributions are as follows:

1. We identify and address the challenge of building AI assistants for short e-learning courses with limited instructional content.
2. We show that augmenting the knowledge database with definitions of key course concepts improves response quality, even without modifying the course itself.
3. We demonstrate that retrieval method variants have relatively little impact compared to content enrichment, providing a practical and scalable solution for educators with limited time and resources.

2 Related work

Recent research on AI-powered educational assistants has largely relied on large-scale datasets. For example, Wang et al. (2024) introduced

Book2Dial, which generates synthetic teacher-student dialogues from 35 textbooks to fine-tune chatbots—though issues like hallucinations and repetitive content remain. Similarly, [Fernandez et al. \(2024\)](#) proposed SyllabusQA, a 5k QA-pair dataset from 63 course syllabi, aimed at handling logistical queries. Despite high similarity scores, factual accuracy remained a challenge.

[Huang et al. \(2025\)](#) presented RAM2C, a RAG-based system generating pedagogically grounded dialogues in liberal arts education. The method depends on rich, curated knowledge bases, which limits applicability to low-resource contexts. [Garcia \(2025\)](#) combined RAG and LLMs to help instructors analyze student reflections and identify course-wide learning challenges through topic modeling. While RAG offered valuable insights, it did not consistently outperform standalone LLMs.

To the best of our knowledge, no prior work has addressed how to design AI assistants for courses with limited textual content, where expanding the material is not feasible due to instructional constraints.

3 AI-assisted course

The study was conducted using the e-learning course *Introduction to Machine Learning*, designed to provide foundational knowledge and develop practical skills in constructing and analyzing simple machine learning models. ([Szczepański et al., 2025](#)). The course emphasizes applied learning through examples and hands-on exercises using Google Teachable Machine. Figure 1 shows an excerpt from the course materials.

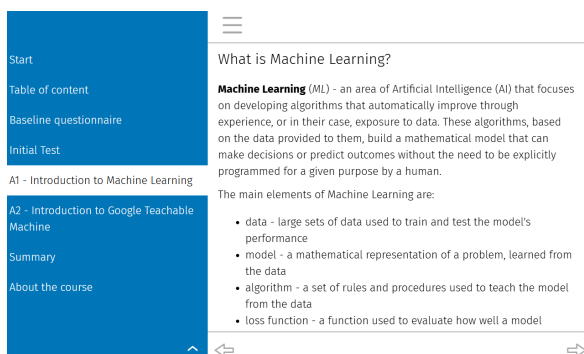


Figure 1: Fragment of the learning materials.

The course is organized into four modules: (1) introduction, (2) data preparation, (3) model training, and (4) evaluation metrics. It is used in AI-related computer science classes as a preparatory resource

to align students' baseline knowledge in machine learning. Estimated completion time is 3–4 hours.

The content is intentionally concise, focusing on the fundamentals of classification. While broader AI and machine learning topics are briefly mentioned, they are not developed in detail. As an introductory resource, it is used by students with varying prior knowledge, leading to diverse questions during the learning process. To support this, an AI assistant was introduced to help in four areas: (1) clarifying course content, (2) deepening understanding of key concepts, (3) addressing related but uncovered topics, and (4) summarizing material.

Expanding the core content was considered pedagogically inappropriate. The course was purposefully limited to foundational topics, with advanced material reserved for later stages in the curriculum. Nonetheless, students may still raise more complex questions. The course is a validated educational resource, positively received by students in earlier editions, and modifying it solely to enhance AI assistant performance was not an option.

Designed for self-paced learning outside of class—where instructor support may be unavailable—the course positions the AI assistant as a key element of the learning experience, offering targeted guidance as students navigate the material independently.

4 Method

The AI assistant embedded in the e-learning course follows RAG architecture, which combines the strengths of large language models with the ability to incorporate domain-specific knowledge—in our case, the textual content of the course.

To develop the system, we compared several RAG variants. After a series of preliminary tests, we decided to adopt AdvanceRAG approach with a query routing mechanism as it yielded the best results. This model involves classifying the user query into a predefined type—such as requesting a citation, paraphrase, summary, or elaboration—and dynamically selecting a tailored prompt accordingly. Based on the chosen prompt, the system retrieves relevant data from a knowledge database, which is then passed to the language model along with the prompt to generate the final response. We chose *LLaMA 3.1 8B* as our foundation model due to its balance between output quality and hardware requirements.

To evaluate how different retrieval strategies im-

pact response quality, we tested four configurations of increasing complexity and contextual coverage:

Baseline The initial setup used only the course text as the knowledge base, paired with a dense retriever. To improve retrieval precision, the material was preprocessed to remove auxiliary or transitional content (e.g., phrases such as “*Let’s move on to the next section*”) that could degrade semantic relevance. This minimal configuration reflects a real-world scenario where the assistant operates solely on content provided by course authors.

Reranking This variant introduced a reranking stage using the ms-marco-MiniLM-L-6-v2 cross encoder to improve semantic relevance.

Extended In this configuration we added curated Wikipedia articles containing definitions of key concepts presented in the course to the document set, to increase the breadth of available information.

Combined The final configuration employs both reranking and the extension of the document set.

5 Experiments

5.1 Data

To evaluate the performance of the proposed system, we collected a set of real user questions related to a short e-learning course on the basics of machine learning. The questions were formulated by students who had previously completed the course. To help participants simulate realistic interactions with an AI assistant, they were instructed to first ask their question and then obtain an answer from ChatGPT, followed by an evaluation of whether the response was satisfactory.

A total of 94 questions were collected from 14 students. Each student submitted between 2 and 10 questions, covering all four modules of the course (28 questions from module 1, 19 from module 2, 29 from module 3, and 18 from module 4).

The resulting dataset consists of natural, goal-oriented queries and can be categorized into four main types: (1) clarifying course content (45 questions); (2) deepening understanding of key concepts (33 questions); (3) addressing related but uncovered topics (10 questions); (4) summarizing material (6 questions).

This dataset forms the basis for evaluating retrieval configurations under realistic student-like usage scenarios.

5.2 Retrieval evaluation

For the purpose of evaluating retrieval performance we measured Reciprocal Rank@K (**RR@K**), Normalized Discounted Cumulative Gain@K (**nDCG@K**), Average Precision@K (**AP@K**), Recall@K (**R@K**), Precision@K (**P@K**) and determined F1 scores. The results are given in Table 1. The *Baseline* solution while competitive in precision for top-ranked results ($RR@3 = 0.2579$), lacked contextual depth, limiting the assistant’s ability to handle more complex or exploratory queries. In case of *Reranking* model the performance improved for $K=1$, however all metrics for $K=3, 5$ decreased. This suggests that reranking narrows the focus at the cost of contextual diversity—an undesirable trade-off in educational settings, where broader context is often beneficial for comprehension. The *Extended* configuration significantly improved context diversity and precision, especially for $K=5$, where precision rose from 0.1167 to 0.1663. However, $nDCG@5$ declined, likely due to the added noise from general-purpose content. The *Combined* approach yielded the best performance for $K=1$ but consistently underperformed for higher values of K , indicating a trade-off between precision at the top and overall contextual coverage. Among all tested configurations, the *Extended* configuration proved most effective. It provided the best balance between precision and recall at $K=3$ and $K=5$ (e.g., $F1@3 = 0.2306$; $F1@5 = 0.2179$), making it well-suited for educational assistants that must deliver context-rich responses aligned with instructional goals.

5.3 End-to-end assessment

To measure end-to-end performance of the AI assistant we asked a group of three experts to assess the quality of the responses yielded by the system. Each expert was provided with reference answers, responses predicted by the system and the contextual information retrieved from the knowledge base for the given question. The experts were requested to verify, if the answer returned by the system is adequate given the provided reference answer and the context being retrieved, with three options available *Yes*, *No* and *Don’t know*. Three configurations were evaluated *Baseline*, *Extended* and *Combined* to measure the impact of enhancing the AI assistant with the definitions of key course concepts on re-

Table 1: Retrieval results

Model	K	RR@K	nDCG@K	AP@K	R@K	P@K	F1 Score
Baseline	1	0.1904	0.1904	0.1785	0.1785	0.1904	0.1843
	3	0.2579	0.3104	0.2569	0.4642	0.1626	0.2410
	5	0.2000	0.2878	0.2012	0.5595	0.1166	0.1931
Reranking	1	0.2142	0.2142	0.2023	0.2023	0.2142	0.2082
	3	0.1607	0.1932	0.1488	0.3134	0.1111	0.1641
	5	0.1644	0.2165	0.1481	0.4146	0.0904	0.1485
Extended	1	0.2530	0.2530	0.0957	0.0957	0.2530	0.1390
	3	0.3092	0.2323	0.1478	0.2574	0.2088	0.2306
	5	0.2263	0.2095	0.1307	0.3160	0.1662	0.2179
Combined	1	0.2650	0.2650	0.1128	0.1128	0.2650	0.1583
	3	0.2168	0.1530	0.0913	0.1720	0.1405	0.1547
	5	0.1903	0.1452	0.0788	0.2192	0.1108	0.1473

Table 2: End-to-end expert evaluation

System type	% Yes	% No	% Don't know	Fleiss' κ
Baseline	47	41	12	0.14
Extended	53	38	9	0.26
Combined	55	31	14	0.24

sponse quality.¹ The evaluation results along with inter-annotator agreement measured with Fleiss' κ are reported in Table 2. The results show that extending the knowledge base with definitions of key concepts lead to the rise of the percentage of adequate answers from 47% for *Baseline* system to 53% in case of *Extended* configuration. The introduction of reranking resulted in additional 2% performance rise as shown by *Combined* configuration, however it has to be noted that it also lead to 5% increase of responses that were considered by the experts to be ambiguous.

5.4 Challenges in human annotation of end-to-end AI assistant responses

The annotation process highlighted how challenging human evaluation can be in the development of AI assistants. System outputs were assessed by annotators (two instructors and one student) familiar with the course content and its pedagogical intent, and the resulting labels reflect the complexity of the task — yet they show that the adopted labeling scheme provided a solid basis for meaningfully capturing the quality of the assistant's responses.

¹Due to the limited availability of human annotators, the *Reranking* configuration was not included in end-to-end evaluation. Nevertheless, the impact of reranking can be observed by contrasting the *Extended* configuration, which does not employ reranking, with the *Combined* configuration, which incorporates both reranking and the extended document set.

For example, the following response to Question 14 (“*What is the reward and punishment system in machine learning?*”) generated by the *Combined* configuration was labeled *No*: *The answer to the student's question: The reward and punishment system in machine learning is a model that learns by interacting with the environment, through a system of rewards and punishments.* The sentence is incoherent, repetitive, and lacks both clarity and educational value; it relies on a tautological structure, effectively rephrasing the question instead of providing a meaningful or instructive definition.

Another example is the response to Question 38 (“*Does class imbalance matter?*”), generated by the *Combined* configuration, which was labeled *Don't know*: *Yes, class imbalance does matter. In the case of a very rare disease and an extremely uneven class distribution — between sick and healthy individuals — both in the training set and in the test set randomly drawn from the population, there is only one sick person for every 99 healthy ones. In such a scenario, the model ignored the minority class and automatically assigned all examples to the dominant one, achieving a very high accuracy of 99%, yet failing to fulfill its primary purpose — correctly diagnosing the sick individual. Consequently, it was to be expected that the model, having ‘seen’ only a single instance of that digit during training, would be unable to learn to recog-*

nize it, and would instead simply ignore it. While the response is generally correct, the explanation is lengthy and somewhat dense. It relates to course content but does not explicitly reference it or direct the learner to definitions provided in the material. As a result, it may hinder comprehension rather than support it, especially for learners unfamiliar with the concept.

A third case illustrates a different kind of difficulty — when the relevant information is simply missing. In response to Question 65 (“*What is an edge case in machine learning?*”), the *Combined* configuration returned: *I cannot answer this question because the context does not contain any passages that could be quoted as a valid answer.* This response was annotated as *No*, although it is, in principle, correct — the concept of edge cases was not covered in the course content nor included in the extended knowledge base. The annotator’s negative judgment likely resulted from the system’s failure to provide a pedagogically helpful justification, such as: *This topic falls outside the scope of the course and is therefore not addressed here.* This example demonstrates that analyzing student queries alongside system responses can help identify gaps in coverage and motivate iterative updates to the knowledge base.

The full answers returned by the system across all three configurations discussed above are included in Appendix C.

6 Discussion

The results of our experiments show that even modest extensions to the assistant’s knowledge base—specifically, the inclusion of carefully selected definitions of course-relevant concepts—can lead to measurable improvements in response quality. However, while the improvements were consistent, they remained moderate in scope. Expert assessments showed only fair agreement (Fleiss’ $\kappa = 0.26$), highlighting the inherent challenges of evaluating AI-generated responses in educational contexts, where interpretation often depends on the perceived intent behind a student’s question.

Instructors providing feedback to students must often determine whether a question stems from confusion, a need for clarification, or simple curiosity. The experts participating in our evaluation may have applied similarly critical reflection when judging the assistant’s answers. During annotation, they likely evaluated the responses based on cri-

teria such as factual correctness, relevance to the question, and linguistic clarity, as well as pedagogical usefulness, alignment with course terminology, and the ability to communicate uncertainty when appropriate.

Additionally, some limitations in response quality likely stem from the assistant’s lack of access to richer content. This may have particularly affected questions aimed at deepening understanding (e.g., through examples beyond those given in the course) or exploring topics that, while present in the instructional material, were not discussed in sufficient detail due to being outside the intended scope of instruction. In these cases, although the retrieved context included terms relevant to the student’s question, the absence of detailed explanation or clear definitions reduced the educational usefulness of the assistant’s response. Such cases highlight the need for a more nuanced expansion of the knowledge base, especially when dealing with boundary concepts that are implicitly acknowledged in course materials but not explicitly explained.

7 Future work

This study did not examine the impact of enriching the assistant’s context with broader resources, such as domain-specific books or curated examples from outside the course scope. Future work should also explore how different segmentation strategies for content added to the knowledge base influence AI assistant performance. Another important direction for future work is expanding the knowledge base with content addressing topics raised by students that are currently missing from both the course and the extended resources. It is also planned to collect feedback on the usefulness of AI assistants during learning, with particular attention to their perceived limitations.

8 Conclusions

Our study shows that AI assistants embedded in short e-learning courses can be improved without expanding the core instructional content. Instead of increasing course length or adding in-line material—which could compromise clarity and coherence—instructors can enhance assistant performance by supplying concise, reference-style content directly to the RAG knowledge base.

9 Limitations

The use of a general-purpose dense retriever not tailored to educational content represents a limitation of this study. Future research should investigate task-adapted or hybrid retrieval methods more closely aligned with instructional needs.

The course materials utilized in the experiments cover only one specific STEM subject. To what extent the presented results can be generalized to social sciences and humanities coursework requires further investigation.

References

- Nigel Fernandez, Alexander Scarlatos, and Andrew Lan. 2024. [SyllabusQA: A course logistics question answering dataset](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10344–10369, Bangkok, Thailand. Association for Computational Linguistics.
- Frank Ley Garcia. 2025. [Llm+rag driven topic modeling](#). In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2, SIGCSETS 2025*, page 1754, New York, NY, USA. Association for Computing Machinery.
- Haoyu Huang, Tong Niu, Rui Yang, and Luping Shi. 2025. [RAM2C: A liberal arts educational chatbot based on retrieval-augmented multi-role multi-expert collaboration](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 448–458, Abu Dhabi, UAE. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, Red Hook, NY, USA. Curran Associates Inc.
- Marcin Szczepański, Grzegorz Gapiński, and Jacek Marciniak. 2025. [Ai tutor: Adaptive e-learning system using expert fuzzy controllers](#). In *Proceedings of the 17th International Conference on Computer Supported Education - Volume 2: CSEDU*, pages 96–107. INSTICC, SciTePress.
- Junling Wang, Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024. [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.

A Sample Student Questions Used for Evaluation

1) Clarifying course content:

- Explain the difference between a training set and a test set in Machine Learning.
- Explain in one sentence what overfitting and underfitting mean in machine learning.

2) Deepening understanding of key concepts:

- Why is precision worse than accuracy?
- When is the F-score a good evaluation metric? In what situations should it be used?

3) Addressing related but uncovered topics:

- What are large language models?
- Why can't you switch between browser tabs during training in Google Teachable Machine?

4) Summarizing material:

- Shorten it by half: ...
- Write it out for me shorter, so that each definition takes up one sentence. ...

B RAG prompts

Common system context (used in all prompts):

You are an educational assistant in an e-learning course on deep learning and artificial intelligence. Your task is to answer student questions only when they are relevant to the course topics. You are capable of analyzing text, answering questions, explaining concepts, summarizing, and paraphrasing content.

Citation prompt: Answer the student's question using only quotations from the context below.

Context: {context}

Chat history: {chat_history}

Student question: {question}

Respond by quoting only fragments of text found in the context. Citations should include full sentences or coherent passages. You may use multiple quotes if needed to fully answer the question. Do not add any words, interpretations, or paraphrasing of your own. If the context does not contain suitable quotes, state this clearly.

C End-to-end expert evaluation

Summarization prompt: Answer the student's question by creating a concise but informative summary based solely on the context.

Context: {context}
Chat history: {chat_history}
Student question: {question}

Craft a summary that synthesizes key information from the context using your own words. The summary should be complete, accurate, and based only on the context. If the context lacks sufficient data, state this clearly.

Paraphrasing prompt: Answer the student's question by explaining the concept using different words, based on the context.

Context: {context}
Chat history: {chat_history}
Student question: {question}

Provide a paraphrased or simplified explanation using only information from the context. Do not quote directly, add new information, or change the original meaning. If the context lacks sufficient information, state this clearly.

General fallback prompt: Answer the student's question using only the context and chat history provided.

Context: {context}
Chat history: {chat_history}
Student question: {question}

Respond clearly and precisely based solely on the available information. Do not use external knowledge. If the context lacks sufficient information, state this clearly.

Routing prompt (for classification): You are an intelligent e-learning assistant that analyzes user questions and classifies them into one of four categories: citation, summarization, paraphrasing, or other.

Input: {question}

Return a valid JSON object with two keys: "option" and "explanation".

Example output:

```
{  
  "option": "citation",  
  "explanation": "The user is asking for a  
  precise definition."  
}
```

Return only valid JSON, with no extra text.

Table C.1: Example 1 of end-to-end expert evaluation

Question 14: What is the reward and punishment system in machine learning?		
System type	System answer	Evaluation
Baseline	The answer to the student’s question: The reward and punishment system in machine learning is a model that learns by interacting with the environment, through a system of rewards and punishments.	No
Extended	Here’s the answer to the student’s question: Reinforcement learning is a subfield of machine learning focused on how software agents should take actions in an environment to maximize a notion of cumulative reward. Reinforcement learning is a way of machine learning where the agent receives feedback in the form of rewards or punishments. In machine learning, the reward and punishment system is part of the learning process, where the agent is rewarded for actions that lead to desirable outcomes and punished for those that lead to undesirable ones.	Yes
Combined	The answer to the student’s question: The reward and punishment system in machine learning is a model that learns by interacting with the environment, through a system of rewards and punishments.	No

Table C.2: Example 2 of end-to-end expert evaluation

Question 38: Does class imbalance matter?		
System type	System answer	Evaluation
Baseline	Yes, class imbalance does matter. In the case of a very rare disease and an extremely uneven class distribution — sick versus healthy individuals — both in the training set and in the test set randomly sampled from the population, there may be only one sick person for every 99 healthy ones. In such a scenario, the model simply ignored the minority class and automatically assigned all examples to the dominant one, achieving a seemingly high accuracy of 99%, while entirely failing at its main task: correctly diagnosing the sick individual. It was therefore to be expected that a model exposed to only a single instance of that digit during training would be unable to learn to recognize it — and would instead simply ignore it. This example clearly shows how important it is to ensure that all classes in the training set are represented in roughly equal numbers.	Don't know
Extended	Yes, class imbalance does matter. For instance, in the case of a very rare disease and an extremely uneven distribution between the sick and the healthy, the model may ignore the minority class and automatically assign all examples to the dominant one — achieving very high accuracy, yet failing to accomplish its primary objective. In such a case, the model will not be able to correctly diagnose the sick individual.	Yes
Combined	Yes, class imbalance does matter. In the case of a very rare disease and an extremely uneven class distribution — between sick and healthy individuals — both in the training set and in the test set randomly drawn from the population, there is only one sick person for every 99 healthy ones. In such a scenario, the model ignored the minority class and automatically assigned all examples to the dominant one, achieving a very high accuracy of 99%, yet failing to fulfill its primary purpose — correctly diagnosing the sick individual. Consequently, it was to be expected that the model, having "seen" only a single instance of that digit during training, would be unable to learn to recognize it, and would instead simply ignore it.	Don't know

Table C.3: Example 3 of end-to-end expert evaluation

Question 40: What is an edge case in machine learning?		
System type	System answer	Evaluation
Baseline	There is no answer to this question in the given context.	No
Extended	There is no answer to this question in the context.	No
Combined	I cannot answer this question because the context does not contain any passages that could be quoted as a valid answer.	No

Table C.4: Original Polish version of Example 1

Question 14 in Polish: Co to jest system kar i nagród w uczeniu maszynowym?	
System type	System answer in Polish
Baseline	Odpowiedź na pytanie kursanta: System kar i nagród w uczeniu maszynowym to model, który uczy się wchodząc w interakcje ze środowiskiem, poprzez system kar i nagród.
Extended	Oto odpowiedź na pytanie kursanta: Reinforcement learning jest obszarem uczenia maszynowego, który koncentruje się na tym, jak agenty software powinni podejmować działania w środowisku, aby maksymalizować pewną koncepcję nagród kumulatywnych. Reinforcement learning to sposób uczenia maszynowego, w którym agent otrzymuje feedback w postaci nagród lub kar za podejmowane działania. W uczeniu maszynowym system kar i nagród jest częścią procesu uczenia się, w którym agent otrzymuje nagrody za podejmowane działania, które prowadzą do pożądaných wyników, a kary za działania, które prowadzą do niepożądanych wyników.
Combined	Odpowiedź na pytanie kursanta: System kar i nagród w uczeniu maszynowym to model, który uczy się wchodząc w interakcje ze środowiskiem, poprzez system kar i nagród.

Table C.5: Original Polish version of Example 2

Question 38 in Polish: Czy ma znaczenie liczebność klas	
System type	System answer in Polish
Baseline	Tak, znaczenie ma liczebność klas. W przypadku bardzo rzadkiej choroby i skrajnie nierównej liczebności klas: chorych i zdrowych - zarówno w zbiorze uczącym, jak i w zbiorze testowym, losowo wybranym z populacji, mamy tylko jednego chorego na 99 zdrowych. Model w takim przypadku zignorował klasę mniej liczną i automatycznie przypisał wszystkie przykłady do klasy dominującej, osiągając przy tym bardzo wysoką dokładność (accuracy) na poziomie 99%, nie spełniając swojego głównego zadania, czyli prawidłowego zdiagnozowania osoby chorej. W związku z tym, należało spodziewać się, że model, "widząc" w procesie treningu tylko pojedyncze wystąpienie tej cyfry, nie będzie w stanie nauczyć się go rozpoznawać, a zamiast tego po prostu będzie je ignorował. Na tym przykładzie widać, jak ważne jest to, by wszystkie klasy w zbiorze uczącym miały podobną liczbę reprezentantów.
Extended	Tak, znaczenie ma liczebność klas. Na przykład, w przypadku bardzo rzadkiej choroby i skrajnie nierównej liczebności klas: chorych i zdrowych, model może zignorować klasę mniej liczną i automatycznie przypisać wszystkie przykłady do klasy dominującej, osiągając przy tym bardzo wysoką dokładność, ale nie spełniając swojego głównego zadania. W takim przypadku model nie będzie w stanie prawidłowo zdiagnozować osoby chorej.
Combined	Tak, znaczenie ma liczebność klas. W przypadku bardzo rzadkiej choroby i skrajnie nierównej liczebności klas: chorych i zdrowych - zarówno w zbiorze uczącym, jak i w zbiorze testowym, losowo wybranym z populacji, mamy tylko jednego chorego na 99 zdrowych. Model w takim przypadku zignorował klasę mniej liczną i automatycznie przypisał wszystkie przykłady do klasy dominującej, osiągając przy tym bardzo wysoką dokładność (accuracy) na poziomie 99%, nie spełniając swojego głównego zadania, czyli prawidłowego zdiagnozowania osoby chorej. W związku z tym, należało spodziewać się, że model, "widząc" w procesie treningu tylko pojedyncze wystąpienie tej cyfry, nie będzie w stanie nauczyć się go rozpoznawać, a zamiast tego po prostu będzie je ignorował.

Table C.6: Original Polish version of Example 3

Question 40 in Polish: Czym jest przypadek graniczny w uczeniu maszynowym?	
System type	System answer in Polish
Baseline	Nie ma odpowiedzi na to pytanie w podanym kontekście.
Extended	Nie ma odpowiedzi na to pytanie w kontekście.
Combined	Nie mogę udzielić odpowiedzi na to pytanie, ponieważ w kontekście nie ma fragmentów, które mogłyby być zacytowane jako odpowiedź na to pytanie.

Beyond Linear Digital Reading: An LLM-Powered Concept Mapping Approach for Reducing Cognitive Load

Junzhi Han

Emory University
Atlanta, GA 30322
molly.han@emory.edu

Jinho D. Choi

Emory University
Atlanta, GA 30322
jinho.choi@emory.edu

Abstract

This paper presents an LLM-powered approach for generating concept maps to enhance digital reading comprehension in higher education. While particularly focused on supporting neurodivergent students with their distinct information processing patterns, this approach benefits all learners facing the cognitive challenges of digital text. We use GPT-4o-mini to extract concepts and relationships from educational texts across ten diverse disciplines using open-domain prompts without predefined categories or relation types, enabling discipline-agnostic extraction. Section-level processing achieved higher precision (83.62%) or concept extraction while paragraph-level processing demonstrated superior recall (74.51%) in identifying educationally relevant concepts. We implemented an interactive web-based visualization tool <https://simplified-cognitext.streamlit.app> that transforms extracted concepts into navigable concept maps. User evaluation (n=14) showed that participants experienced a 31.5% reduction in perceived cognitive load when using concept maps, despite spending more time with the visualization (22.6% increase). They also completed comprehension assessments more efficiently (14.1% faster) with comparable accuracy. This work demonstrates that LLM-based concept mapping can significantly reduce cognitive demands while supporting non-linear exploration.

1 Introduction

Complex academic texts in higher education present cognitive challenges for students who must process and retain extensive information across diverse disciplines. These challenges are particularly pronounced for neurodivergent students, including those with ADHD (Attention Deficit Hyperactivity Disorder), who process information differently and often struggle to identify and retrieve central ideas from traditional linear texts despite recognizing

their importance (Ben-Yehudah and Brann, 2019; Yeari et al., 2018).

These challenges are further amplified in digital reading environments. Rather than attempting to improve traditional digital reading directly, we propose concept maps as an alternative digital interface that transforms linear text into interactive visual knowledge structures, bypassing linear reading challenges entirely while maintaining comprehensive coverage of educational content. These visual representations externalize knowledge structures, potentially reducing cognitive load while supporting the visual-spatial processing strengths often seen in students with attention-related learning differences (Sperotto, 2016; Sweller, 1988). Rather than attempting to improve traditional reading directly, concept maps provide an alternative knowledge access method that bypasses linear reading challenges entirely while maintaining comprehensive coverage of educational content.

Despite advances in automated concept mapping, significant gaps remain in current tools. Existing automated approaches typically rely on rule-based systems or predefined ontologies that lack flexibility across different domains and disciplines, often struggling with domain-specific terminology and conceptual relationships. Furthermore, existing approaches frequently extract concepts without adequately capturing the nuanced relationships between them, resulting in concept maps that lack the semantic depth necessary for comprehensive understanding.¹

This paper investigates how large language models (LLMs) can generate comprehensive concept maps from educational texts across diverse academic disciplines. Rather than enhancing traditional digital reading, we propose concept maps as an alternative educational interface that trans-

¹All our resources including source codes and concept map data are publicly accessible through our open source project: <https://github.com/emorynlp/cognitext>

forms linear text into interactive visual knowledge structures, enabling students to access the same information through non-linear exploration. We examine three research questions:

1. How effectively can LLMs identify key concepts across diverse academic disciplines without domain-specific training or ontologies?
2. What differences exist in the extraction and representation of knowledge relationships across different academic disciplines?
3. To what extent do automatically generated concept maps reduce cognitive load and improve reading comprehension compared to traditional linear reading?

Our approach implements concept extraction to identify key terms and ideas, relation identification to determine semantic connections between concepts, and concept map generation to organize these elements into visual knowledge structures. We evaluate system performance across ten academic disciplines and assess the impact of resulting concept maps on reading comprehension and cognitive load.

This work makes several key contributions: (1) we establish a methodological framework for LLM-based concept extraction across different types of academic content; (2) we provide empirical evidence for domain-specific knowledge structures that inform adaptive concept mapping; (3) we demonstrate the practical application of language models for educational concept map generation; and (4) we present evidence that concept map visualization as a reading alternative significantly reduces cognitive load while maintaining or improving comprehension outcomes.

While each system component (preprocessing granularity, LLM-based extraction, and user evaluation) merits individual investigation, this work demonstrates their integration for practical educational applications. We focus on the educational impact as our primary contribution, with detailed component analysis providing supporting evidence for system design decisions.

2 Related Work

2.1 Cognitive Load in Educational Contexts

Cognitive Load Theory, developed by (Sweller, 1988), distinguishes between three types of mental

processing: intrinsic load (inherent task complexity), extraneous load (poor instructional design), and germane load (meaningful learning processes). Educational interventions that reduce extraneous load while maintaining or enhancing germane load can greatly improve learning outcomes, particularly for students with diverse cognitive processing patterns.

Complex academic texts impose substantial cognitive demands through dense information presentation, abstract concept relationships, and linear narrative structures that may not align with individual learning preferences. For neurodivergent students, these challenges are particularly pronounced. Le Cunff et al. (2024) demonstrated that neurodivergent students, particularly those with ADHD, reported significantly higher extraneous cognitive load compared to neurotypical peers, while showing no differences in intrinsic or germane cognitive load. This pattern suggests that the presentation format of educational materials, rather than their inherent complexity, creates disproportionate challenges.

While digital reading environments present additional complications—with neurobiological research by Zivan et al. (2023) revealing higher cognitive load patterns in screen-based reading—the fundamental challenge extends beyond medium to the linear, text-heavy presentation of complex information. Bahari et al. (2023) identified several approaches that successfully manage cognitive load in educational environments, including visualization-based approaches and argument mapping, with most strategies aimed at reducing extraneous cognitive load while fostering germane load through generative learning practices.

Concept maps specifically address cognitive load by transforming extraneous processing demands into germane learning opportunities. By externalizing knowledge structures and providing visual-spatial representations, concept maps can reduce the working memory burden of maintaining conceptual relationships while reading, allowing students to focus cognitive resources on understanding and connecting ideas rather than tracking linear narrative flow.

2.2 Automated Concept and Relation Extraction for Education

Automated concept extraction and concept map generation has evolved from rule-based approaches to sophisticated machine learning methods. Early

computational approaches established foundations for extracting concept maps from educational texts, with [Aguiar et al. \(2018\)](#) providing comprehensive approaches for concept maps mining from text.

Recent work has applied large language models to concept map generation. [Perin et al. \(2023\)](#) demonstrated automated concept map generation using fine-tuned large language models, while other work has shown that LLMs can identify conceptually complex regions of text ([Garbacea et al., 2021](#)) and extract concepts from academic materials while preserving semantic relationships ([Zhang et al., 2023](#)).

Educational relation extraction differs from general-purpose approaches in its emphasis on pedagogically meaningful connections that support learning progression ([Dessi et al., 2020](#)). Recent advances in prompt-based approaches have shown particular promise for educational contexts. [Chen et al. \(2022\)](#) introduced KnowPrompt, incorporating knowledge from relation labels into prompt construction. Advancing this, [Chen et al. \(2024\)](#) developed a Generative context-Aware Prompt-tuning method (GAP) that eliminates the need for domain experts to design prompts. Large language models can extract relational knowledge when prompted appropriately ([Jiang et al., 2020](#)), with models like GPT-3.5 and GPT-4 demonstrating competitive performance in processing domain-specific educational content with minimal training requirements ([Hu et al., 2024](#)).

The key challenge lies in distinguishing concepts and relations of varying pedagogical importance while capturing both local conceptual connections and broader structural relationships that reflect disciplinary knowledge organization. Our work builds on these foundations by implementing hierarchical concept classification and multi-level relation identification for cross-disciplinary educational applications.

2.3 Visualization Techniques for Knowledge Representation

Concept maps, originally developed by Novak ([Novak, 1998](#)) and refined by Novak and Cañas ([Novak and Cañas, 2008](#)), organize knowledge through explicit propositions with labeled relationships (e.g., "Photosynthesis → produces → Oxygen"). Unlike mind maps ([Buzan, 1993](#)), which support ideational writing through brainstorming via radial structures, concept maps employ hierarchical representations optimized for reading comprehension

and systematic knowledge acquisition.

Meta-analyses demonstrate educational effectiveness: [Anastasiou et al. \(2024\)](#) found a moderate positive effect of concept maps on science achievement ($g = 0.776$) across 55 studies, while [Schroeder et al. \(2018\)](#) reported similar benefits of learning with concept maps on educational outcomes ($g = 0.58$) across 142 studies. Recent neurological research by [Shealy et al. \(2022\)](#) demonstrated that concept mapping alters cognitive activation patterns, increasing activity in brain regions associated with divergent thinking. [Yang et al. \(2025\)](#) addressed cognitive load concerns by introducing progressive concept maps that integrate information incrementally, improving learning outcomes compared to conventional approaches.

Our approach builds on Novakian concept mapping theory specifically for reading support rather than writing facilitation. Alternative text representation frameworks include Schema Theory ([Bartlett, 1932](#); [Rumelhart, 1977](#)), van Dijk and Kintsch's macrostructures ([van Dijk and Kintsch, 1983](#)), and Rhetorical Structure Theory ([Mann and Thompson, 1988](#)). Our work combines LLM contextual understanding with comprehensive knowledge representation across diverse disciplines, implementing progressive disclosure to manage cognitive load in educational concept maps.

3 Methodology

We developed a systematic approach for extracting concepts and relationships from educational texts and transforming them into interactive concept maps. Figure 1 illustrates the key components of our methodology.

3.1 Dataset Selection and Preparation

We selected ten Wikipedia articles representing diverse academic disciplines: biology, mathematics/statistics, computer science, linguistics, art, history, philosophy, political science, health/medicine, and one general non-academic field. Articles were chosen based on specific criteria to represent content that undergraduate students would likely encounter during their academic studies but would not be familiar with from prior education.

Each selected article maintained sufficient conceptual depth, a neutral academic tone, and introduced new concepts rather than common basics. The articles ranged from 1,383 to 11,337 words (mean: 4,839). Preprocessing steps included

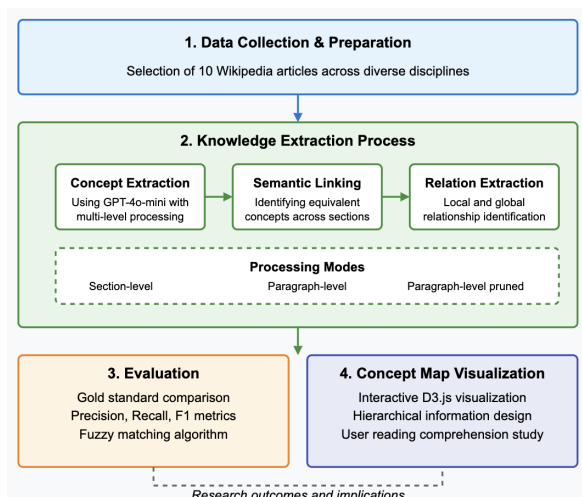


Figure 1: Overview of the methodology for concept & relation extraction, and concept map visualization.

HTML removal, section identification, reference removal, and tokenization using spaCy.

3.2 Text Processing Modes

We implemented three distinct text processing modes to evaluate how different granularities of input text affect extraction quality:

1. **Section-level processing** utilizes complete sections from articles as input units, enabling the system to process larger chunks of coherent text and potentially capture broader thematic relationships. Each complete section was provided as single input to GPT-4o-mini for concept and relation extraction.

2. **Paragraph-level processing** operates on individual paragraphs, allowing the system to focus on local concepts and relationships within more concentrated contexts. Individual paragraphs were processed separately as independent input units, with results from all paragraphs subsequently aggregated to form the complete concept map for each article.

3. **Paragraph-pruned processing** applies post-processing filters to the results obtained from paragraph-level processing to address noise and irrelevant concepts. After completing the standard paragraph-level extraction described above, we applied two filtering mechanisms: first, elimination of concepts appearing exclusively in single paragraphs to reduce noise from isolated mentions; second, semantic filtering using the allMiniLM-L6-v2 transformer model by Reimers and Gurevych (2019) to calculate similarity scores between concepts and their section contents, preserving only

concepts with similarity scores above 0.6.

3.3 Extraction Framework

3.3.1 Concept Extraction

The concept extraction process utilizes GPT-4o-mini with precise prompting strategies to identify educationally relevant concepts within academic texts. A concept is defined as "a significant term or phrase that represents a fundamental idea, entity, or phenomenon within a discipline."

The extraction methodology implements a hierarchical schema that categorizes concepts into three distinct layers based on educational significance:

1. **Priority Layer (Core Concepts):** Fundamental principles, key terminology, major themes, and critical processes (15-20%)
2. **Secondary Layer (Supporting Concepts):** Sub-processes, related theories, and component parts (40-50%)
3. **Tertiary Layer (Contextual Elements):** Author contributions, specific examples, and historical developments (30-40%)

We developed a specialized prompt structure that explicitly targets concepts answering fundamental knowledge questions ("what," "how," "why," and "when"), ensuring comprehensive coverage across knowledge dimensions.

3.3.2 Relation Extraction

The relation extraction framework defines semantic connections between previously extracted concepts as structured triplets consisting of a source concept, a target concept, and a descriptive relation type. The extraction employs a multi-tiered approach:

1. **Local Relations:** Connections between concepts within the same textual segment (section or paragraph)
2. **Global Relations:** Higher-order connections between concepts across different sections

This dual-layer approach addresses the challenge of aligning relation extraction with learning objectives by prioritizing pedagogically meaningful connections. Local relations establish foundational concept understanding within focused contexts, while global relations reveal broader conceptual frameworks essential for comprehensive domain knowledge. Combined with our hierarchical schema (Priority, Secondary, Tertiary layers) from concept extraction (Section 3.3.1), this ensures that extracted

relations support progressive learning objectives, with core concepts and their relationships receiving priority in visualization to align with pedagogical goals of foundational understanding before detailed exploration. Each identified relationship includes supporting evidence from the source text that validates its authenticity and enhances educational value, ensuring extracted relations support structured learning goals rather than arbitrary semantic associations.

3.4 Evaluation Framework

To establish a gold standard dataset, two undergraduate annotators independently performed complete manual concept and relation extraction from each article. The first annotator had greater familiarity with STEM disciplines, while the second had stronger background in liberal arts and humanities. Each annotator identified all educationally relevant concepts and their relationships within each text, creating comprehensive manual annotations that served as ground truth for evaluating our automated extraction performance.

Concepts were rated on a 4-point scale (0-3) assessing educational significance from irrelevant (0) to core concepts (3). Relations were evaluated on a 3-point scale (0-2) based on pedagogical utility. Inter-annotator agreement was assessed using Cohen's Kappa coefficient, with values of $\kappa = 0.76$ for concepts and $\kappa = 0.71$ for relations, indicating substantial agreement between annotators despite their different disciplinary backgrounds.

The evaluation employed precision, recall, and F1 scores, comparing our automated extractions against the manually created gold standard annotations. Fuzzy matching was applied to accommodate linguistic variability between automated and manual annotations. The matching algorithm applied a hierarchical process beginning with exact string comparisons and progressively applying more flexible techniques based on edit distance and word-level similarity to identify semantically equivalent concepts and relations across the two annotation sets.

We acknowledge that using undergraduate annotators rather than domain experts across all ten disciplines represents a limitation in our evaluation methodology, as disciplinary expertise could affect the identification of field-specific concepts and relationships.

3.5 Concept Map Visualization

We implemented an interactive concept map visualization system called Cognitext using D3.js force-directed layouts. The interface, shown in Figure 2, incorporates several features designed to support effective navigation:

1. **Hierarchical information architecture:** Priority concepts are displayed with the darkest node coloring to indicate their fundamental importance, while secondary and tertiary concepts remain initially hidden to prevent cognitive overload
2. **Self-directed exploration:** When priority concepts have connections to hidden lower-level concepts, they display a pulsing orange indicator circle that signals available deeper exploration and encourages user interaction
3. **Visual focus management:** Selecting concepts brings related nodes to the foreground while fading others
4. **Relationship transparency:** Hovering over connections reveals relationship types and supporting textual evidence
5. **Intelligent content enhancement:** An integrated concept chatbot provides contextual explanations and answers questions based on the extracted knowledge

Users interact with the system through multiple methods: exploring from central nodes outward, clicking for concept explanations, and rearranging nodes to customize their view. The implementation uses Python for backend processing and Streamlit Cloud for web deployment: <https://simplified-cognitext.streamlit.app>.

3.6 User Evaluation

To assess the efficacy of concept maps for reading comprehension, we conducted a controlled study with 14 undergraduate participants (8 female, 6 male; ages 19-24; 4 self-identified as ADHD). While the sample size is modest, it provides initial insights into concept map effectiveness.

Concept Map Generation. For the user study, concept maps were generated using section-level processing results, which achieved the highest average F1 score in our evaluation. To ensure valid

4 Results

4.1 Extraction Performance

Table 1 presents the performance metrics for concept extraction across various academic disciplines using fuzzy matching. We compared three distinct text processing approaches.

Section-level processing demonstrated superior precision across all disciplines, achieving an average precision of 83.62%, with biology (89.86%) and general domain (87.35%) articles showing highest performance. However, this approach showed comparatively lower recall (62.18% average), indicating that while it extracted highly relevant concepts, it missed some concepts present in the gold standard dataset.

Paragraph-level processing yielded considerably higher recall metrics (74.51% average), with political science (81.63%) and history (81.35%) articles showing highest recall. This improvement came at the cost of precision, which dropped to 57.49% average. The paragraph-level pruned approach achieved intermediate performance in both precision (66.87%) and recall (70.92%).

When comparing F1 scores, section-level processing performed best overall (71.20% average), followed by paragraph-level pruned (68.82%) and standard paragraph-level (64.89%) approaches.

Relation extraction performance followed similar patterns across processing approaches and disciplines (detailed results in Appendix A Table 3). Section-level processing achieved highest overall precision (78.61%), with biology and general articles showing particularly strong performance (82.09% and 83.51% respectively) and moderate recall (59.76%). Paragraph-level processing exhibited substantially lower precision (51.95%) but higher recall (69.08%), while the pruned approach demonstrated intermediate performance with 62.01% precision and 67.29% recall. Overall, section-level processing performed best with an average F1 score of 67.71%, followed by paragraph-level pruned (64.52%) and standard paragraph-level (59.28%) approaches.

4.2 Concept/Relation Distribution Patterns

Figure 3 presents a heatmap visualization of the normalized concept distribution across academic disciplines. When normalized for text length, the philosophy article showed the highest concept density (31.09 concepts per 1,000 words), followed by health/medicine (30.93) and political science

(27.66). In contrast, the history (9.61) and art (10.86) articles exhibited the lowest density.

Normalization revealed discipline-specific patterns in concept distribution. The computer science article emphasized problems & solutions and mathematical foundations (both 3.14 concepts per 1,000 words). The health/medicine article showed a pronounced focus on medical & safety concepts (7.73) and processes & mechanisms (6.87). The philosophy article demonstrated a great emphasis on core concepts (6.51) and socio-cultural contexts (4.34).

For relation distribution, Table 5 in Appendix A shows that structural relations emerged as the most frequent relation type across all disciplines, with the health/medicine (12.89), philosophy (11.57), and political science (10.19) articles showing the highest normalized frequencies. The overall relation density varied substantially across disciplines, with health/medicine exhibiting the highest density (54.22 relations per 1,000 words), followed by philosophy (45.55) and political science (41.48).

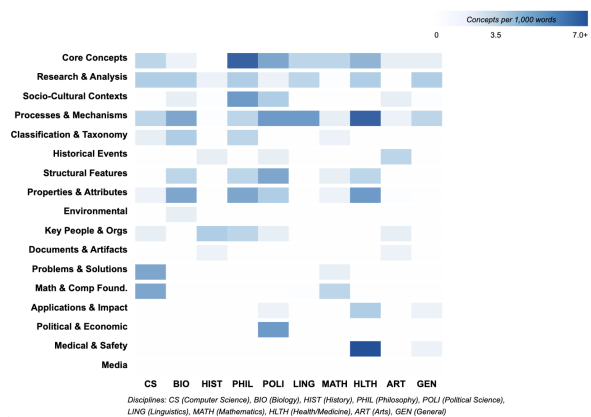


Figure 3: Concept distribution heatmap across academic disciplines, normalized per 1,000 words. Color intensity represents frequency, darker blue = higher density.

4.3 User Study Results

To evaluate concept map effectiveness on reading comprehension, we compared traditional linear reading with concept map-assisted reading (Table 2). Participants spent more time with the concept mapping tool (32.5 vs. 26.5 minutes, 22.6% increase) but completed comprehension assessments more quickly (18.3 vs. 21.3 minutes, 14.1% decrease).

Most significantly, participants reported substantially reduced perceived mental effort when using concept maps (5.0 vs. 7.3 on NASA TLX scale), representing a 31.5% decrease in cognitive

load that was statistically significant ($p = 0.00092$). Comprehension accuracy showed slight improvement (98% vs. 97%), though not statistically significant.

We acknowledge that our assessment instrument focused on propositional knowledge, which concept maps represent effectively, and this should be considered when interpreting the comparable accuracy results.

User feedback was generally positive (mean rating 4.21/5), with participants highlighting the tool's ability to "visually analyze basic concepts" and "show hierarchical relationships between different concepts."

Metric	Without Tool	With Tool
Reading Time	26.5 min	32.5 min
Assessment Time	21.3 min	18.3 min
Mental Effort	7.3	5.0
Correctness	97%	98%

Table 2: Comparison of reading performance with and without concept mapping tool. Reading time refers to text reading (linear condition) vs. concept map exploration (concept map condition)

5 Analysis and Discussion

5.1 Extraction Performance and Disciplinary Patterns

The consistent precision-recall trade-off across processing approaches demonstrates a fundamental tension in concept extraction methodology. Section-level processing achieved higher precision through comprehensive contextual understanding, while paragraph-level processing captured more concepts through granular analysis. The substantial precision advantage for relations (78.61% vs. 51.95%) suggests that relational semantics often depend on broader contextual understanding than can be captured within paragraph boundaries, aligning with discourse coherence theory that semantic relationships emerge from macro-level textual structures.

The paragraph-level pruned approach effectively mitigated many limitations of standard paragraph-level processing while preserving much of its recall advantage. This suggests that incorporating multi-stage validation processes into extraction pipelines can substantially improve performance without requiring contextual windows as large as section-level processing.

Cross-disciplinary analysis revealed systematic knowledge organization patterns reflecting epistemological differences between fields. Scientific text (biology) showed superior extraction metrics across all approaches, suggesting more explicit externalization of conceptual relationships through standardized linguistic patterns. Historical text performed better with paragraph-level processing, indicating conceptual relationships are established within localized narrative units rather than extended theoretical frameworks. The linguistics article's extraction challenges highlight complexities in meta-disciplinary discourse where language is both medium and subject of analysis.

The distinctive patterns in concept and relation distribution reflect disciplinary epistemologies. Philosophy's high density of causal relations but absence of functional relations suggests emphasis on conceptual reasoning, while health/medicine's high functional relation density points to procedural knowledge focus. These findings suggest that universal knowledge representation approaches may not be optimal, and concept maps might benefit from tailoring to specific relation structures observed in different academic content types.

5.2 Concept Map Effectiveness for Cognitive Load Reduction

The comparison of concept mapping visualization to traditional linear reading reveals a complex relationship between time investment, cognitive load, and comprehension outcomes. The observed increase in reading time (22.6%) paired with a decrease in assessment time (14.1%) when using the concept mapping tool suggests a shift in cognitive resource allocation. While users invested more time in initial exploration, they subsequently completed assessment tasks more efficiently.

The substantial reduction in perceived mental effort (31.5%) despite longer engagement time represents one of our most significant findings. This relationship suggests the visualization transformed extraneous cognitive load into germane cognitive load, enabling more productive mental processing rather than simply reducing overall demands. This transformation is particularly valuable for educational applications where sustained engagement with complex material is desirable.

While the marginal improvement in comprehension accuracy (1%) appears modest, this should be interpreted within the context of the already high baseline performance (97%), suggesting a po-

tential ceiling effect. The combination of comparable comprehension outcomes with significantly reduced cognitive effort indicates an improved efficiency ratio—participants achieved similar results with less mental strain.

The generally positive user feedback (4.21/5) confirms participants recognized value in the visualization approach. The implementation of hierarchical information architecture with progressive disclosure aligns with cognitive load theories by preventing information overload while maintaining access to comprehensive content, enabling incremental mental model construction while preserving underlying concept connections.

5.3 Implications for Educational Technology

These findings suggest that domain-agnostic extraction systems face inherent limitations, and maximizing extraction performance might benefit from adaptive approaches tailored to different discourse types. The cognitive load reduction without compromising comprehension indicates concept mapping tools could be particularly valuable for students experiencing cognitive fatigue during traditional reading, including those with attention-related difficulties.

Moreover, the implementation of hierarchical information architecture with progressive disclosure aligns with cognitive load theories by preventing information overload while maintaining access to comprehensive content. The observed self-directed exploration through concept maps aligns with constructivist learning principles, suggesting these tools can support diverse learning approaches and accommodate individual differences in background knowledge and processing styles while transforming extraneous cognitive load into germane cognitive load.

6 Conclusion and Future Work

This work demonstrates that LLM-based concept mapping can significantly reduce cognitive demands while supporting non-linear exploration of educational content. Our findings suggest concept mapping tools particularly benefit students experiencing cognitive fatigue, transforming linear text into interactive visualizations that support self-directed exploration.

Future work should prioritize three key directions based on our empirical findings:

Adaptive Extraction Methodologies. Our anal-

ysis revealed systematic variation across disciplines, with philosophy articles showing highest concept density (31.09 per 1,000 words) while history articles exhibited lowest (9.61). Future research should develop adaptive methodologies that automatically adjust to disciplinary discourse patterns by: (1) implementing discourse pattern recognition to select optimal processing granularity, (2) developing domain-specific relation taxonomies based on our finding that structural relations dominate in health/medicine (12.89 per 1,000 words) while causal relations are prominent in philosophy (10.85 per 1,000 words), (3) creating hierarchical processing pipelines that combine section-level precision (83.62%) with paragraph-level recall (74.51%), and (4) developing automated quality assurance methodologies that can refine extracted concepts and relations without manual intervention.

Longitudinal Impact Studies. While our study (n=14) demonstrated 31.5% cognitive load reduction, critical questions remain about long-term educational impact. Future studies should examine knowledge retention and transfer beyond immediate comprehension, conduct targeted research with larger samples of neurodivergent students (our study included only four self-identified participants), and investigate whether concept maps sustainably reduce cognitive load across time and contexts.

LMS Integration. Current implementation barriers limit practical deployment. Priority efforts should focus on developing plugins for major learning management systems (Canvas, Blackboard, Moodle), implementing collaborative features for shared editing and instructor annotations, and addressing processing latency through scalable architectures for institutional deployment.

These directions address our core finding that concept mapping transforms extraneous cognitive load into germane cognitive load while maintaining comprehension outcomes.

Limitations

Despite the promising results of this research, several limitations should be acknowledged when interpreting our findings.

Corpus limitations. Our analysis was restricted to examining only one article per academic discipline, which limits generalizability. The findings should be interpreted as article-specific observa-

tions that suggest potential disciplinary patterns rather than definitive characterizations of entire domains. Future work should expand the corpus to include multiple texts from each discipline to establish more generalizable patterns.

Ecological validity limitations. Our use of Wikipedia articles, while providing standardized, neutral content across disciplines, may not fully represent typical educational materials such as textbooks, journal articles, or course-specific readings. Wikipedia's encyclopedic structure and hyperlinked format may facilitate concept extraction differently than traditional academic texts. Future work should evaluate performance on authentic course materials to establish broader applicability.

Evaluation methodology limitations. Our evaluation relied on undergraduate annotators rather than domain experts, and comprehension questions were developed by the lead researcher rather than assessment professionals. While inter-annotator agreement was substantial and questions assessed multiple understanding levels, expert involvement would strengthen future evaluations.

Generalizability across learning differences. While we hypothesized particular benefits for students with attention-related learning differences, our sample included only a small number of self-identified neurodivergent participants (n=4). More targeted research with larger samples of neurodivergent students would be necessary to substantiate claims about differential benefits across students.

Assessment methodology limitations. The node-and-edge structure of concept maps excels at representing straightforward propositions (e.g., "Photosynthesis produces Oxygen") but struggles with conditional relationships (e.g., "A causes B only under specific conditions"), complex temporal sequences, or counterfactual reasoning. Consequently, our question development may have inadvertently excluded assessment items requiring these more complex cognitive operations, potentially favoring the concept map condition. While this limitation does not invalidate our findings within the scope of propositional knowledge comprehension, it restricts generalizability to the full spectrum of reading comprehension skills typically assessed in educational contexts.

Technical constraints. The use of GPT-4o-mini, while cost-efficient, introduced model-specific constraints including knowledge cutoff limitations, reduced parameter capacity compared to larger models, and context window restrictions that particu-

larly affected global relation extraction. These limitations may have impacted extraction performance, especially for longer documents. The system also demonstrated significant processing latency with longer articles, which could limit practical deployment in time-sensitive educational contexts.

Implementation challenges. The current implementation faces practical deployment barriers including limited integration with existing learning management systems, basic visualization capabilities without advanced features like collaborative editing, and minimal customization options for educators. These constraints, while providing clear directions for future development, limit immediate broad adoption in educational settings.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback that significantly improved this work. We are grateful to our user study participants for their time and valuable insights. Special thanks to the Computer Science Department of Emory University and the Emory NLP Lab for providing computational resources and technical support. We are grateful to everyone who contributed to the creation of Cognitext and supported our efforts to improve educational comprehension for all learners.

References

- Camila Zacche Aguiar, Davidson Cury, and Amal Zouaq. 2018. [Towards technological approaches for concept maps mining from text](#). *CLEI Electronic Journal*, 21(1):7.
- Dimitris Anastasiou, Clare Nangsin Wirngo, and Pantelis Bagos. 2024. [The effectiveness of concept maps on students' achievement in science: A meta-analysis](#). *Educational Psychology Review*, 36(39):1–18.
- Akbar Bahari, Sumei Wu, and Paul Ayres. 2023. [Improving computer-assisted language learning through the lens of cognitive load](#). *Educational Psychology Review*, 35:53.
- Frederic C. Bartlett. 1932. *Remembering: A Study in Experimental and Social Psychology*. Cambridge University Press.
- Gal Ben-Yehudah and Adi Brann. 2019. [Pay attention to digital text: The impact of the media on text comprehension and self-monitoring in higher-education students with ADHD](#). *Research in Developmental Disabilities*, 89:120–129.

- Tony Buzan. 1993. *The Mind Map Book: Unlock Your Creativity, Boost Your Memory, Change Your Life*. BBC Books, London.
- Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. [Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction](#). In *Proceedings of the ACM Web Conference 2022 (WWW '22)*, pages 1–11, New York, NY, USA. ACM.
- Zhenbin Chen, Zhixin Li, Yufei Zeng, Canlong Zhang, and Huifang Ma. 2024. [GAP: A novel generative context-aware prompt-tuning method for relation extraction](#). *Expert Systems with Applications*, 248:123478.
- Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. 2020. [Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain](#). *CoRR*, abs/2011.01103.
- Cristina Garbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. [Explainable prediction of text complexity: The missing preliminaries for text simplification](#). 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 1086–1097. Association for Computational Linguistics.
- Sandra G Hart and Lowell E. Staveland. 1988. [Development of NASA-TLX \(Task Load Index\): Results of empirical and theoretical research](#). In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, pages 139–183. North-Holland.
- Yan Hu, Qingyu Chen, Jingcheng Du, Xueqing Peng, Vipina Kuttichi Keloth, Xu Zuo, Yujia Zhou, Zehan Li, Xiaoqian Jiang, Zhiyong Lu, Kirk Roberts, and Hua Xu. 2024. [Improving large language models for clinical named entity recognition via prompt engineering](#). *Journal of the American Medical Informatics Association*, 31(9):1812–1820.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. [How can we know what language models know?](#) *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Anne-Laure Le Cunff, Vincent Giampietro, and Eleanor Dommett. 2024. [Neurodiversity positively predicts perceived extraneous load in online learning: A quantitative research study](#). *Education Sciences*, 14(5).
- William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.
- Joseph D. Novak. 1998. *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*. Lawrence Erlbaum Associates.
- Joseph D. Novak and Alberto J. Cañas. 2008. The theory underlying concept maps and how to construct and use them. Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition.
- Wagner Perin, Davidsom Cury, Camila Aguiar, and Crediné Menezes. 2023. [From text to maps: Automated concept map generation using fine-tuned large language model](#). In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 1317–1328, Porto Alegre, RS, Brasil. SBC.
- 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.
- David E. Rumelhart. 1977. Schemata: The building blocks of cognition. In Rand J. Spiro, Bertram C. Bruce, and William F. Brewer, editors, *Theoretical Issues in Reading Comprehension*, pages 33–58. Lawrence Erlbaum Associates.
- N.L. Schroeder, J.C. Nesbit, C.J. Anguiano, and 1 others. 2018. [Studying and constructing concept maps: a meta-analysis](#). *Educational Psychology Review*, 30:431–455.
- T. Shealy, J. S. Gero, and P. Ignacio. 2022. [How the use of concept maps changes students' minds and brains](#). In *ASEE Annual Conference and Exposition, Conference Proceedings*.
- L. Sperotto. 2016. [The visual support for adults with moderate learning and communication disabilities: How visual aids support learning](#). *International Journal of Disability, Development and Education*, 63(2):260–263.
- John Sweller. 1988. [Cognitive load during problem solving: Effects on learning](#). *Cognitive Science*, 12(2):257–285.
- Teun A. van Dijk and Walter Kintsch. 1983. *Strategies of Discourse Comprehension*. Academic Press.
- Kai-Hsiang Yang, Hui-Chun Chu, Gwo-Jen Hwang, and Tzu-Jung Liu. 2025. [A progressive concept map-based digital gaming approach for mathematics courses](#). *Educational Technology Research and Development*.
- M. Yeari, E. Vakil, L. Schifer, and R. Schiff. 2018. [The origin of the centrality deficit in individuals with attention-deficit/hyperactivity disorder](#). *Journal of Clinical and Experimental Neuropsychology*, 41(1):69–86.
- Xiaoyu Zhang, Jianping Li, Po-Wei Chi, Senthil Chandrasegaran, and Kwan-Liu Ma. 2023. [ConceptEVA: Concept-based interactive exploration and customization of document summaries](#). In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23*, New York, NY, USA. Association for Computing Machinery.

Michal Zivan, Sasson Vaknin, Nimrod Peleg, Rakefet Ackerman, and Tzipi Horowitz-Kraus. 2023. [Higher theta-beta ratio during screen-based vs. printed paper is related to lower attention in children: An EEG study.](#) *Plos one*, 18(5):e0283863.

A Appendix

A.1 Detailed Extraction Performance

Table 3 presents the detailed performance metrics for relation extraction across the ten academic disciplines. Similar to concept extraction, we compared three distinct text processing approaches: section-level, paragraph-level, and paragraph-level pruned processing.

Disc.	Section			Paragraph			Para-Pruned		
	P	R	F1	P	R	F1	P	R	F1
CS	79.3	56.9	66.3	49.5	63.9	55.8	60.2	64.9	62.5
Bio	82.1	66.8	73.7	56.8	74.9	64.6	65.9	74.3	69.9
Hist	78.3	66.1	71.7	57.6	76.5	65.7	66.2	70.4	68.2
Phil	79.2	50.4	61.6	50.6	66.2	57.4	59.0	67.6	63.0
Pol	76.5	62.7	68.9	52.0	75.6	61.6	61.9	68.9	65.2
Ling	78.8	47.6	59.4	46.1	57.4	51.2	55.8	59.2	57.5
Art	77.6	59.0	67.1	52.8	68.3	59.6	60.4	66.2	63.2
Math	73.4	57.7	64.7	47.9	66.2	55.6	59.7	63.2	61.4
Med	77.4	64.9	70.6	49.2	68.3	57.2	63.6	67.7	65.5
Gen	83.5	65.5	73.4	56.9	73.6	64.2	67.4	70.5	68.9
Avg	78.6	59.8	67.7	52.0	69.1	59.3	62.0	67.3	64.5

Table 3: Performance of relation extraction by discipline with fuzzy matching (P = Precision, R = Recall, F1 = F1 Score).

A.2 Concept and Relation Categorization

Table 4 presents the normalized distribution of concept types across disciplines (per 1,000 words), highlighting discipline-specific knowledge organization patterns.

Concept	CS	BIO	HIST	PHIL	POL	LING	ART	MATH	MED	GEN
Core	2.2	1.2	0.6	6.5	4.7	2.6	1.6	2.2	3.4	1.7
Research	2.7	2.6	1.1	2.9	1.1	2.2	0.0	0.9	2.6	2.5
SocioCult	0.0	1.5	0.9	4.3	2.9	0.4	1.4	0.2	0.0	1.0
Process	2.5	3.8	1.0	2.2	4.4	4.8	1.2	1.9	6.9	2.5
Classif	1.7	2.6	0.0	2.2	0.4	0.0	0.0	1.4	0.0	0.0
Struct	0.0	2.3	0.0	2.2	3.6	0.0	0.0	1.7	2.2	0.0
Property	1.2	3.2	0.6	3.6	2.9	0.0	0.9	1.0	4.7	0.8
Environ	0.0	1.5	0.6	0.0	0.0	0.0	0.0	0.3	0.0	0.0
People	1.7	0.3	2.4	2.2	1.8	0.0	1.4	0.7	0.0	0.0
Docs	0.0	0.0	1.3	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Problems	3.1	0.0	0.0	0.0	0.4	0.0	0.0	1.9	0.0	0.6
Math/Comp	3.1	0.0	0.0	0.0	0.0	0.9	0.0	2.1	0.0	0.0
Impact	0.7	0.0	0.1	0.0	1.1	0.0	0.5	0.7	3.0	1.3
Politics	0.0	0.0	0.1	0.7	4.0	0.0	0.3	0.0	0.0	0.0
Medical	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	7.7	1.3
Media	0.3	0.0	0.1	0.0	0.0	0.0	0.5	0.3	0.0	0.0
Events	0.0	0.6	1.5	0.0	1.5	0.0	2.1	0.0	0.4	0.0

Table 4: Normalized concept distribution (per 1,000 words) across all disciplines.

Table 5 presents the normalized distribution of relation types across disciplines (per 1,000 words).

A.3 Example Prompt Templates

To facilitate reproducibility, we provide examples of the prompt templates used for concept and relation extraction:

Relation Type	BIO	LING	PHIL	HLTH	CS
Structural	6.74	4.40	11.57	12.89	6.95
Causal	2.34	4.40	10.85	10.31	5.96
Impact	4.98	3.96	10.12	9.88	6.29
Functional	6.45	1.32	0.00	9.02	5.79
Interaction	2.05	3.52	6.51	6.01	4.14
Cognitive	0.00	3.08	0.00	0.86	0.50
Linguistic	0.00	2.20	0.00	0.43	0.17

Table 5: Normalized relation distribution (relations per 1,000 words) for selected relation types across five representative disciplines.

A.3.1 Concept Extraction Prompt (Abbreviated)

A concept is defined as a significant term or phrase that represents a fundamental idea, entity, or phenomenon within a discipline. Extract key concepts from the provided text using the following guidelines.

Concept Layers: 1. **Core Concepts (Priority Layer):** - Primary theoretical concepts and fundamental principles - Key terminology and definitions essential to the topic - Major themes and overarching frameworks

2. **Supporting Concepts (Secondary Layer):** - Sub-processes and variations of core concepts - Related theories and complementary ideas - Component parts and organizational structures

3. **Contextual Elements (Tertiary Layer):** - Author names and their key contributions - Specific examples and case studies - Historical context and developments

Output Format: [List of JSON objects with entity, context, evidence, and layer fields]

A.3.2 Relation Extraction Prompt (Abbreviated)

Extract key relationships between these available concepts using the following guidelines. The extracted relations will be used for visualizations to aid educational comprehension.

Guidelines: - Ensure that the relations are clearly defined and relevant to the text’s main ideas. - Focus on capturing a variety of relationship types without restricting to specific categories. - Avoid speculative relationships; only include those with explicit or strong implicit textual support.

Output Format: [List of JSON objects with source, relation_type, target, and evidence fields]

Verb Placement Error Detection Datasets for Learners of Germanic Languages

Noah-Manuel Michael

Kiel University, Germany

Leibniz Institute for Science and
Mathematics Education, Kiel, Germany

Linköping University, Sweden
michael@ipn.uni-kiel.de

Andrea Horbach

Kiel University, Germany

Leibniz Institute for Science and
Mathematics Education, Kiel, Germany

horbach@ipn.uni-kiel.de

Abstract

Correct verb placement is difficult to acquire for second-language (L2) learners of Germanic languages. However, word order errors and, consequently, verb placement errors, are heavily underrepresented in benchmark datasets of NLP tasks such as grammatical error detection (GED)/correction (GEC) and linguistic acceptability assessment (LA). If they are present, they are most often naively introduced, or classification occurs at the sentence level, preventing the precise identification of individual errors and the provision of appropriate feedback to learners. To remedy this, we present **GermDetect**: Universal Dependencies-based (UD), linguistically informed verb placement error detection datasets for learners of Germanic languages, designed as a token classification task. As our datasets are UD-based, we are able to provide them in most major Germanic languages: Afrikaans, German, Dutch, Faroese, Icelandic, Danish, Norwegian (Bokmål and Nynorsk), and Swedish. We train multilingual BERT (mBERT) models on GermDetect and show that linguistically informed, UD-based error induction results in more effective models for verb placement error detection than models trained on naively introduced errors. Finally, we conduct ablation studies on multilingual training and find that lower-resource languages benefit from the inclusion of structurally related languages in training.

1 Introduction

Correct verb placement is difficult to acquire for L2 learners of Germanic languages. This is due to the placement depending on different factors such as finiteness and the clause type in which the verbs occur. Example (1) illustrates a Dutch sentence consisting of a single main clause.

(1) *Hij heeft een hond gekocht.*

he has a dog bought
S V2 O O V

‘He has bought a dog.’

Two characteristics can be observed here: The finite verb *heeft* is placed in the second position (V2) and the non-finite, participle verb *gekocht* is placed clause-finally. Example (2), consisting of a main and a subordinate clause, illustrates how the verb placement changes: Here, the finite *weet* occupies the V2 position in the main clause, but both the finite *heeft* and non-finite *gekocht* are placed clause-finally in the subordinate clause, following subject-object-verb (SOV) word order.

(2) *Ik weet dat hij een hond heeft gekocht.*

I know that he a dog has bought
S V2 _ S O O V V

‘I know that he has bought a dog.’

These and other context-dependent changes in verb placement in Germanic languages are not trivial for L2 learners. Therefore, being able to provide accurate feedback to learners about their verb placement is crucial.

However, word order errors are heavily underrepresented in both GED/GEC shared task datasets and LA datasets. If they are present, they are often introduced naively, which means they do not represent the kinds of errors L2 learners are likely to make. This leaves the capability of GED systems to detect naturalistic word order errors underexplored. Verb placement errors, as a subset of word order errors, are even more poorly represented, which, in the context of Germanic languages, is particularly critical. Additionally, LA datasets are often formulated as sentence-level binary classification or pair-wise ranking tasks. This does not allow for locating errors within a sentence and therefore offers

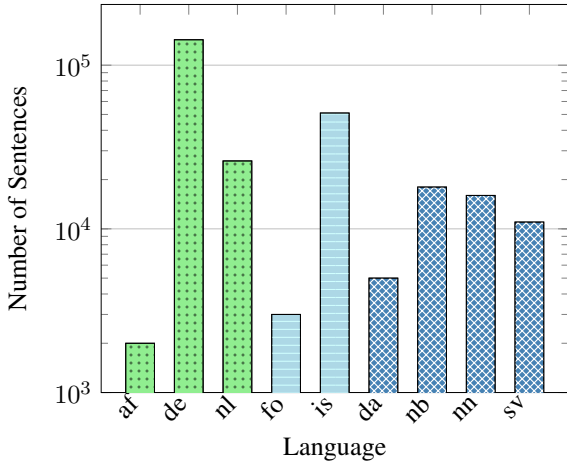


Figure 1: Number of sentences per language in GermDetect (log scale); language codes follow ISO 639-1.

little usability for providing feedback to learners.

To address this gap, we present GermDetect: UD-based, linguistically informed verb placement error detection datasets for learners of Germanic languages that can serve as a new benchmark to test GED systems’ capabilities in detecting naturalistic verb placement errors. The token-level classification design allows individual errors in verb placement to be located, and the datasets are available in most major Germanic languages:¹ Afrikaans, German, Dutch, Faroese, Icelandic, Danish, Norwegian (Bokmål and Nynorsk), and Swedish. Figure 1 presents a brief overview of the magnitude of our dataset, with a more detailed breakdown in Appendix A. We make the datasets and the code available at <https://github.com/noahmanu/gerlangmod>.

In the following sections, we provide a brief introduction to related work, followed by a description of our dataset creation algorithm. We then present the results of different mBERT configurations on our new benchmark.

2 Related Work

In this section, we introduce verb placement rules in Germanic languages, briefly talk about the frequency of verb placement errors in learner corpora, present relevant GED/GEC/LA datasets and their shortcomings with regard to the evaluation of word order errors, and survey popular GED/GEC tools’ capabilities in detecting verb placement errors.

¹Excluding English as English is the only modern Germanic language that does not follow V2.

2.1 Verb Placement Rules in Germanic Languages

Correct verb placement is challenging to acquire for L2 learners of Germanic languages (Orgassa, 2009; Schimke and Dimroth, 2018; Westergaard and Lohndal, 2019; Angantýsson, 2021). This is due to the placement depending on different factors such as finiteness and the clause type in which the verbs occur.

All Germanic languages covered in GermDetect follow V2 in main clauses, which means that the finite verb always occupies the second position. Example (1) illustrates how in West Germanic languages, SOV is the default syntax pattern for all other verbs, i.e., non-finite verbs are placed clause-finally in main clauses. In subordinate clauses, where V2 does not hold, all verbs follow SOV and are therefore found at the end of the phrase. Example (3) demonstrates by means of a Swedish sentence that, in North Germanic languages, all verbs follow SVO in main clauses.

- (3) *Han har köpt en hund.*
 he has bought a dog
 S V2 V O O

This masks the V2 constraint, as the surface word order in main clauses often resembles that of canonical SVO languages. Example (4) briefly presents how, when an element other than the subject takes the clause-initial position, V2 holds while the rest of the main clause follows SVO.

- (4) *Kanske har han köpt en hund.*
 maybe has he bought a dog
 – V2 S V O O
 ‘Maybe he has bought a dog.’

Additionally, all GermDetect languages form polar questions by inversion, placing the finite verb in the clause-initial position.² Examples (5) and (6) illustrate this structure for both German and Faroese. Non-finite verbs follow the respective default syntax patterns.

- (5) *Hat er einen Hund gekauft?*
 has he a dog bought
 V1 S O O V
 ‘Has he bought a dog?’

²In analogy to V2, we call this position V1.

Languages	Main–Finite	Main–Non-Finite	Subordinate	PolarQ–Finite	PolarQ–Non-Finite
af, de, nl	V2	SOV	SOV	V1	SOV
fo, is, da, nb, nn, sv	V2	SVO	SVO	V1	SVO

Table 1: Overview of the most unmarked syntax patterns in the languages covered by GermDetect; distinction between finite and non-finite verbs in main clauses and polar questions (PolarQ), and subordinate clauses.

Dataset	Task	% WOE	Languages
Dale and Kilgarriff (2011)	GEC	< 7.5	en
Ng et al. (2013)	GEC	0.0	en
Ng et al. (2014)	GEC	2.4	en
Napoles et al. (2017)	GEC	N/A	en
Bryant et al. (2019)	GEC	1.6	en
Warstadt et al. (2019)	LA	N/A	en
Warstadt et al. (2020)	LA	19.4	en
Nielsen (2023)	LA	50.0	<u>da</u> , <u>fo</u> , <u>is</u> , <u>nb</u> , <u>nn</u> , <u>sv</u>
Volodina et al. (2023)	GED	N/A	cs, <u>de</u> , en, it, <u>sv</u>
Masciolini et al. (2025)	GEC	N/A	cs, <u>de</u> , el, en, et, <u>is</u> , it, lv, ru, sl, <u>sv</u> , uk

Table 2: Percentage of word order errors (WOE) in different benchmark datasets; languages covered by GermDetect underlined.

(6) *Hevur hann keypt ein hund?*
has he bought a dog
V1 S V O O

Table 1 summarizes the most unmarked syntax pattern for both language groups, i.e., West Germanic and North Germanic. With verb placement being this complex in Germanic languages, learners are prone to make errors when trying to acquire the correct syntax patterns. However, learners typically do not make random errors in verb placement. Instead, the errors they make are often influenced by their previous language background. Therefore, certain error types are less likely to occur than others. Example (7) shows an unlikely example of an error where the learner splits the noun phrase *een hond* by misplacing a verb between the elements of this constituent. This is unlikely because nominal constituents are generally joint units, so this would most likely be considered an error in any of the languages the learner knows.

(7) **Ik weet dat hij een heeft hond.*
I know that he a has dog

Example (8), in contrast, illustrates a more likely error that could be produced by a first-language English speaker.

(8) **Ik weet dat hij heeft een hond.*
I know that he has a dog

It is evident that verb placement presents itself as a very complex system of rules for L2 learners of Germanic languages to acquire. Next, we briefly show how this is reflected in real learner corpora.

2.2 Verb Placement Errors in Germanic Learner Corpora

In the German part of the MERLIN corpus (Boyd et al., 2014),³ which comprises texts from all CEFR levels but is especially rich in texts from levels A2–C1,⁴ 34% of errors are annotated as “movement errors”, i.e., errors corrected by placing a token in a different position within the same sentence.⁵ Out of these, 12.1% involve the misplacement of a verb.⁶ The vast majority of verb placement errors in MERLIN are concentrated among the levels A2–B2 (94.9%).

In the FalkoEssayL2 v2.4 corpus (Lüdeling et al., 2008),⁷ which only comprises German L2 texts at the levels B2–C2, movement errors make up 29.6% of total errors. Out of these, 12.5% involve the misplacement of a verb. This suggests that even at advanced levels, learners continue to produce verb placement errors at rates comparable to those at

³ Accessible at: <https://commul.eurac.edu/annis/merlin/>, last accessed: 2025/06/03.

⁴ CEFR: Common European Framework of Reference for Languages.

⁵ We only count grammatical errors towards the total, i.e., spelling errors are excluded.

⁶ Appendix B contains the queries with which we determined the number of total grammatical error occurrences in MERLIN and Falko, the number of movement errors, and the number of movement errors involving the misplacement of a verb.

⁷ Accessible at: <https://korpling.german.hu-berlin.de/falko-suche/>, last accessed: 2025/06/03.

Tool	Languages	Verb Placement	Large-Scale Eval
GermDetect	af, de, nl, fo, is, da, nb, nn, sv	af, de, nl, fo, is, da, nb, nn, sv	✓
Grammarly	en	/	✗
LanguageTool	en, de, nl, da, sv, +21 others	de	✓
ProWritingAid	en	/	✓
Quillbot	en, de, nl, +3 others	de, nl	✗

Table 3: Overview of grammar checking tools with language support, verb placement error detection capabilities, and large-scale evaluation capabilities.

earlier stages, highlighting the pedagogical value of providing targeted feedback on such errors.

In the Icelandic Child Language Error Corpus and the Icelandic L2 Error Corpus (Ingason et al., 2021; Glisic and Ingason, 2022), there are two designated error categories for verb placement errors related to V2 violations.⁸ This indicates that verb placement is not trivial not only in L2 learning but also in first-language acquisition.

However, verb placement errors as a subset of word order errors are heavily underrepresented in all relevant benchmark datasets as we will point out in the following section.

2.3 Word Order Errors in GED/GEC/LA Benchmark Datasets

In recent years, several shared tasks have been organized in the field of GED/GEC. Additionally, LA tasks have been developed to test language models’ linguistic capabilities. Table 2 presents the most prominent benchmark datasets from all three domains covering Germanic languages and the percentage of word order errors they contain.

All of the datasets have in common that word order errors typically only make up a very small fraction of the errors present within them, or no information about the distribution of word order errors is provided at all. Germanic languages other than English have only recently seen their representation increase. However, with the exception of the ScaLA dataset (Nielsen, 2023), no information is available about the presence and distribution of word order errors in the datasets containing subsets of the languages covered by GermDetect. This leaves the capabilities of language models in detecting erroneous word order underexplored.

This situation is especially critical in the context

⁸Both corpora are accessible at: <https://github.com/icelandic-lt/iceErrorCorpusSpecialized/>, last accessed: 2025/06/03. Verb placement errors are annotated with the error code *v3*.

of Germanic languages, whose successful acquisition depends on mastering their complex verb placement rules. Verb placement errors, as a subset of word order errors, are consequently even more poorly represented in the datasets, thus limiting the development of GED systems capable of reliably detecting and providing feedback on such errors. Moreover, in the only Germanic dataset where information about the presence of word order errors is available – the ScaLA dataset – such errors were introduced using a naive corruption strategy that swaps adjacent tokens.⁹ As a result, the dataset neither specifically targets verb placement errors nor reflects the kind of word order errors that naturally occur in learner language. Thus, it provides little insight into whether language models can detect naturalistic word order errors that could be produced by L2 learners.

2.4 Survey GED/GEC Tools

While numerous writing assistants claim to provide feedback on grammatical errors, few are suitable for detecting verb placement errors in Germanic languages other than English. Among those capable of handling word order errors to some extent, many lack API access, which complicates large-scale evaluation.

Table 3 presents an overview of the most popular tools, the languages they support, and whether word order errors, particularly those involving verb placement, are among the phenomena they claim to be able to identify and provide feedback on. As the table shows, two of the most popular tools – Grammarly and ProWritingAid – do not support any of the relevant languages covered by GermDetect. While Quillbot claims to handle syntax errors in German and Dutch, and LanguageTool does so

⁹To ensure the corrupted sentences are indeed ungrammatical, Nielsen (2023) enforces a variety of part-of-speech restrictions prohibiting certain token swaps.

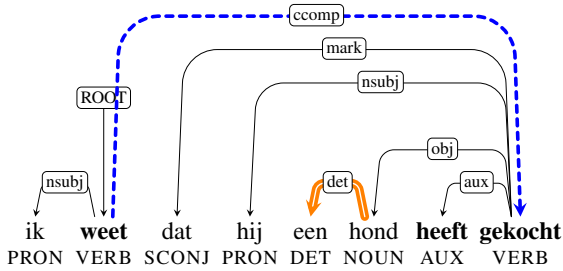


Figure 2: Dependency tree for the Dutch sequence “ik weet dat hij een hond heeft gekocht”; blue arc indicates where our algorithm splits verb phrases; orange arc indicates where our algorithm aggregates noun phrases into impermeable units.

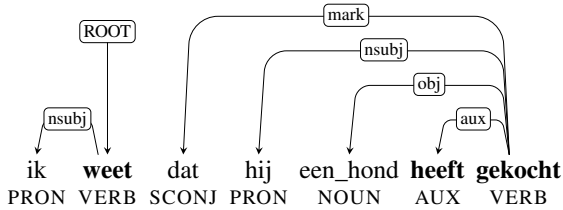


Figure 3: Dependency tree for the Dutch sequence “ik weet dat hij een hond heeft gekocht”; verb-headed phrases are aggregated as verbal heads plus their dependencies; noun-headed phrases are impermeable units.

for German, our manual evaluation on 100 GermDetect test sentences for each of the two languages found that their feedback on verb placement errors was largely unreliable, inaccurate, or incomplete.¹⁰ This highlights that effective tools for providing feedback on verb placement in Germanic languages are unavailable as of today. To remedy this, in the following section, we present our linguistically informed corruption algorithm that can introduce verb placement errors into any UD-annotated Germanic language subject to V2 verb placement.

¹⁰A locally hosted server of LanguageTool’s Python wrapper (Version 6.5; available at <https://pypi.org/project/language-tool-python/>, last accessed: 2025/06/03) flagged none of the 132 verb placement errors present in the German test sentences as word order errors; via their respective GUIs, LanguageTool was able to identify 41 and Quillbot was able to identify 38 out of 132 verb placement errors. For Dutch, Quillbot was able to identify 37 out of 130 verb placement errors. We counted errors as correctly identified even when they were not explicitly flagged as syntax errors, as long as the tool highlighted the relevant part of the sentence – whether as a missing or superfluous word or verb, or with a vague message such as “something is wrong” in combination with the appropriate error correction suggestion, among other generic error types.

3 Dataset Creation

Here, we briefly present the UD datasets as a basis for GermDetect, we present our preprocessing steps, and we describe our data corruption algorithm in detail.

3.1 Universal Dependencies

As the basis for our datasets, we use the UD Treebanks (Version 2.15; Zeman et al., 2024), including all available datasets for each of the GermDetect languages. Appendix A summarizes the datasets and their sizes after removing sentences without verbs.

Figure 2 illustrates how sentences are annotated for dependency relations in UD. Our algorithm currently operates both on the part-of-speech (POS)-tag level and the dependency arc level, but tokens are typically annotated with more linguistic information. We remove common punctuation tokens and lowercase all characters to ensure that models trained on GermDetect data cannot fall back on any orthographic information when determining the (in)correctness of verb placement and have to rely on their syntactic understanding only. This is especially relevant for sentences that begin with verbs, as the initial capitalization can reveal their original position. If such verbs are moved to another part of the sentence, the capitalization may serve as an unintended signal of misplacement.

3.2 Corruption Algorithm

Given a dependency parse tree T for a sentence $S = \{w_1, w_2, \dots, w_n\}$, we define an aggregation procedure to extract syntactic substructures centered around **full verbs** and **nouns**. The algorithm proceeds as follows:

Step 1 - Aggregation of Verb-Headed Phrases. Identify all tokens $v \in S$ such that $\text{POS}(v) = \text{VERB}$. These serve as the roots of *verb-headed phrases*.¹¹

For each full verb v , construct a *verb-headed phrase* $V_v \subseteq S$ defined as:

- V_v includes the full verb v itself,
- Plus all tokens in S that are **recursively governed by v** – i.e., all descendants of v in the subtree rooted at v ,

¹¹Due to inconsistent annotations, some UD treebanks use the VERB POS-tag for adjectives derived from verbs. We implemented a number of additional checks to prevent these tokens from heading their own verb-headed phrases.

- With the constraint that no token is included if the dependency path from it to v passes through another full verb (cf. verb phrase splitting in Figure 2).

This ensures that each verb-headed phrase captures the local syntactic scope of a full verb, while nested verbs and their subtrees are aggregated independently.

Step 2 - Aggregation of Noun-Headed Phrases within Verb-Headed Phrases. Identify all tokens $n \in V_v$ such that $\text{POS}(n) = \text{NOUN}$. These serve as the roots of *noun-headed phrases* within verb-headed phrases.

For each noun n , construct a *noun-headed phrase* $N_n \subseteq V_v$ defined as:

- N_n includes the noun n itself,
- Plus all tokens in V_v that are **recursively governed** by n – i.e., all descendants of n in the subtree rooted at n ,
- With the constraint that no token is included if the dependency path from it to n passes through another noun.

Each noun phrase N_n is treated as an **impermeable unit**: During any subsequent processing, i.e., the data corruption, the tokens in N_n must remain contiguous and in their original order. No external tokens may be inserted into the span of a noun-headed phrase.

Step 3 - Output. The final output is a collection of verb-headed phrases $\{V_{v_1}, V_{v_2}, \dots, V_{v_i}\}$, each rooted at a full verb. Within each V_v , zero or more noun phrases $\{N_{n_1}, N_{n_2}, \dots, N_{n_j}\} \subseteq V_v$ are identified as impermeable subunits. This structure supports the linguistically-informed manipulation of the sentence while preserving core syntactic boundaries. Figure 3 illustrates how, with the help of UD’s dependency structure, we are able to isolate syntactic structures that often correspond to main and subordinate clauses. This allows us to inject the data with more targeted corruptions, which aim to reproduce learner errors more closely.

Step 4 - Corruption of Verb-Headed Phrases. Each extracted verb-headed phrase V_v is corrupted by permuting the positions of a randomly selected subset of its verb tokens. Specifically:

- Identify all verb tokens $\tau \in V_v$ such that $\text{POS}(\tau) \in \{\text{VERB}, \text{AUX}\}$. Each such token is

selected for permutation with a probability of roughly $p = 0.5$, resulting in the dataset containing approximately as many correctly placed as incorrectly placed verbs.

- The selected verb tokens may be relocated to any position within V_v , either before or after any other token, **provided that the relative order of all non-verb tokens is preserved.**
- Noun-headed phrases $N_n \subseteq V_v$ remain contiguous and untouched; verb tokens may move around them but not split them.
- If $V_v = V_{v_1}$, i.e., it is the first verb-headed phrase in the sentence, no verb token may occupy the first position in V_v , unless a verb originally located in this position remains in it. This constraint is imposed to avoid the accidental generation of valid polar question syntax, which would undermine the goal of synthetically generating syntactically perturbed structures.
- All verb tokens that are moved from their original positions are automatically labeled as syntactically incorrect (F), enabling the generation of training data for GED models. This labeling strategy is justified by the relatively rigid verb placement rules found in Germanic languages, as explained earlier. Verbs that remain in their original position are labeled as correct (C), all non-verb tokens are labeled as other (O).

Example (9) illustrates how misplacing a verb in the first position is not permitted by our algorithm. This is to avoid generating well-formed polar questions that would be incorrectly labeled as ungrammatical (cf. Table 1).¹²

- (9) ik **weet** | dat hij een_hond heeft gekocht
weet ik* | dat hij een_hond heeft gekocht

Example (10) showcases all possible corruptions when we reduce the number of verbs in the subordinate clause of our running example to one, resulting in *ik weet dat hij een hond heeft* “I know that he has a dog”. There are 5 positions that *heeft* can theoretically take. One of them corresponds to the correct placement, while the last corruption in

¹²In addition to the standard *left asterisk indicating linguistically unacceptable examples, we use the asterisk on the right* side to indicate examples not permitted by our algorithm.

the example is not permitted due to the impermeable noun-headed phrase constraint. If a verb in the phrase is selected to be permuted, a permitted position is randomly chosen.

- (10) ik weet | dat hij een_hond **heeft**
 ik weet | ***heeft** dat hij een_hond
 ik weet | *dat **heeft** hij een_hond
 ik weet | *dat hij **heeft** een_hond
 ik weet | *dat hij een **heeft** hond*

Finally, example (11) illustrates what a labeled corrupted sentence could look like if we reintroduced the second verb. Note that the relative order of non-verb tokens always remains the same but the relative order of verb tokens to one another can change.

- (11) *ik weet* | **dat gekocht* *hij heeft een*
 O C | O F O F O
hond
 O

The GermDetect algorithm makes it possible to insert errors that specifically target verb placement while making sure to exclude misplacements that are unlikely, such as breaking up noun phrases, or that would result in a well-formed sentence despite the change of verb placement, such as in polar questions. The full extent of our dataset is summarized in Appendix A.

4 Benchmark Results

In this section, we present our experimental setup and analyze the results of evaluating various mBERT configurations trained on GermDetect.

4.1 Experimental Setup

Following the creation of the GermDetect dataset, we train and evaluate multiple mBERT models using various training dataset configurations and their combinations (Devlin et al., 2019). We use mBERT as our base model primarily due to computational constraints and methodological considerations. All experiments are conducted locally on a MacBook Pro with an Apple M4 Pro chip and 24 GB of RAM (macOS 15.4.1), which makes training larger models such as XLM-R impractical (Conneau et al., 2020). In addition to its lighter memory footprint, mBERT’s relatively lower performance ceiling provides a clearer basis for analyzing the impact of

training data composition. Specifically, we compare training on the target language alone with training on the target language and related Germanic languages, as explained in Section 4.3. Since stronger models like XLM-R often achieve robust performance regardless of training data composition, they may obscure more subtle transfer effects that are easier to detect with a smaller model. This choice also supports the growing emphasis on environmentally responsible NLP, as smaller models require significantly less energy to train and deploy.

To implement our approach, we add a BertForTokenClassification head to mBERT and fine-tune the sequence tagger using Hugging Face’s Trainer API. Appendix C summarizes the parameters we use to train our models. For each input sentence, the model generates one of three labels for each token: O for tokens that are not verbs, and C (correct) or F (false) for verb tokens, depending on whether their placement in the sentence is correct or incorrect.

We retain the original dataset splits provided by UD for training, development, and testing, and we do not make any further modifications beyond removing sentences without any verb tokens, as explained in Section 3.1. During training, the models are evaluated based on their loss on the development set. At inference time, in line with earlier works in GED (Bell et al., 2019; Yuan et al., 2021; Volodina et al., 2023), the models are evaluated using the macro-averaged $F_{0.5}$ score, which we compute exclusively over the C and F categories. This metric places greater emphasis on precision than recall, which aligns with standard practice in intelligent computer-assisted language learning applications where false positives, i.e., flagging correct structures as incorrect, are especially undesirable as they risk demotivating learners.

4.2 Monolingual Baseline Configurations

Table 4 presents the $F_{0.5}$ performance results for different configurations of the mBERT model on the GermDetect test data.

We evaluate three baseline configurations: target, random, and adjacent. The target configuration trains models exclusively on GermDetect data from the target language. The random configuration trains models on data in which half of the verbs assume any position within a sentence other than their original position, while the other half remain in their original positions. This corruption strategy represents generic verb placement errors,

mBERT Configuration	F _{0.5} Score by Language								
	af	de	nl	fo	is	da	nb	nn	sv
random	0.74	0.80	0.72	0.63	0.74	0.76	0.80	0.79	0.79
adjacent	0.71	0.67	0.69	0.68	0.64	0.72	0.74	0.75	0.75
target	0.82	0.94	0.88	0.72	0.82	0.85	0.90	0.89	0.86
all	0.88	0.94	0.89	0.85	0.84	0.90	0.93	0.92	0.91
all-balanced	0.86	0.94	0.89	0.79	0.83	0.88	0.92	0.91	0.90
west	0.87	0.93	0.88	–	–	–	–	–	–
west-balanced	0.84	0.94	0.88	–	–	–	–	–	–
north	–	–	–	0.84	0.83	0.89	0.92	0.91	0.90
north-balanced	–	–	–	0.78	0.83	0.86	0.92	0.91	0.89
island	–	–	–	0.82	0.83	–	–	–	–
island-balanced	–	–	–	0.75	0.83	–	–	–	–
mainland	–	–	–	–	–	0.88	0.91	0.90	0.89
mainland-balanced	–	–	–	–	–	0.87	0.92	0.90	0.89

Table 4: Macro-averaged F_{0.5} performance scores (computed over the C and F categories only) of different configurations of the mBERT model across the Germanic languages covered by GermDetect; models ablate the influence of training classifiers based on different data corruption strategies and by combining the training data of structurally related groups of languages; balanced indicates that the target language sets the upper limit for how many sentences of each language are used to train the classifier.

i.e., linguistically uninformed verb placement errors. For the adjacent configuration, half of the verbs switch positions with one of their adjacent tokens (both left and right swaps are equally frequent), approximating the word-order error induction mechanism described in Nielsen (2023), but applied to verbs only.

As shown in the results, the target configuration significantly improves performance across all languages compared to both the random and adjacent configurations, indicating that training directly on GermDetect data effectively enhances the detection of naturalistic verb placement errors.

4.3 Ablation of Multilingual Configurations

Next, we assess the effects of training models on configurations that include structurally related languages.

For the all configuration, one model is trained on all available data. Similarly, the west, north, island, and mainland configurations each include all data from the languages within their respective groups.¹³ In the balanced configurations, the number of sentences from each language is capped at the level of the target language, ensuring that the target language contributes at least as many sen-

tences as any other included language. This means that, e.g., in the north-balanced configuration with Faroese as the target language, if the Faroese dataset contains X sentences, then each of the other North Germanic languages can contribute at most X sentences to the training set, ensuring that Faroese is not underrepresented in the training data.

The results demonstrate that training on all available data yields the best-performing models, consistent with the expectation that more data generally improves performance. However, training solely on West Germanic data yields performance scores nearly equivalent to those obtained by using all data when tested on West Germanic languages, a trend also observed among North Germanic languages. Furthermore, training exclusively on mainland Scandinavian languages results in only a minor performance reduction in these languages, compared to training on all North Germanic languages. These observations suggest that, while incorporating more diverse data is generally beneficial, the models effectively exploit structural similarities among related languages, achieving performance scores close to the best-performing configurations. Balancing the representation of languages within the training sets does not provide additional benefits; thus, it is preferable to utilize all available data.

¹³West Germanic: af, de, nl. North Germanic: fo, is, da, nb, nn, sv. Island Scandinavian: fo, is. Mainland Scandinavian: da, nb, nn, sv.

It is equally unsurprising that German benefits the least from the inclusion of related languages in training, as it is by far the most well-represented language in GermDetect. Similarly, Icelandic and Dutch also exhibit only minor performance improvements. In contrast, Afrikaans and Faroese – being the languages with the least available data – benefit the most from the inclusion of data from related languages in training.

5 Conclusion

We have introduced **GermDetect**: UD-based, linguistically informed verb placement error **detection** datasets for learners of **Germanic** languages, designed as a token classification task. As our datasets are UD-based, we can provide them in most major Germanic languages: Afrikaans, German, Dutch, Faroese, Icelandic, Danish, Norwegian (Bokmål and Nynorsk), and Swedish. Unlike existing resources, GermDetect targets a specific and pedagogically relevant error type that is under-represented in current benchmark datasets and goes undetected by most existing GED/GEC tools. Our results show that multilingual models trained on data corrupted by the GermDetect algorithm outperform models trained on naively corrupted data. Furthermore, while training on all data consistently yields the highest performance, models trained on structurally related languages perform nearly as well – demonstrating the benefits of typological similarity. Crucially, the amount of available training data strongly influences the degree to which models benefit from multilingual training: high-resource languages such as German, Icelandic, and Dutch see marginal improvements, whereas low-resource languages such as Afrikaans and Faroese benefit substantially. These findings highlight the importance of both linguistic structure and data quantity in training robust GED/GEC models and suggest that targeted, linguistically informed error induction can support the development of systems capable of providing fine-grained feedback on complex syntactic phenomena such as verb placement in Germanic languages.

Limitations

In its current implementation, the verb placement error generation algorithm is subject to several limitations. It assumes UD sentences to be grammatically well-formed and their annotations to be accurate. However, this is not always the case, as data

quality can vary and annotation practices can differ across datasets. To address this, future iterations of the algorithm should include more robust checks to ensure that all tokens are treated correctly, even in the presence of inconsistent or imperfect annotations. Another limitation lies in the assumption that every corruption introduced by the algorithm results in an incorrect sentence. In theory, however, some corruptions can still yield well-formed or acceptable constructions. Although restrictions are currently in place to prevent the generation of polar question syntax, and the relatively rigid word order of Germanic languages minimizes the number of such cases, they are not entirely eliminated. A manual inspection of 100 German sentences revealed an error rate of 2.9% in which verbs were incorrectly labeled. For Dutch, this error rate was 2.1%. Future developments should aim to reduce these accidentally grammatically sound relocations of verbs even further. In the Dutch sentences we manually checked, we also noticed that, for very long sentences, the algorithm sometimes did not restore the correct order of the extracted verb-headed phrases. This affected the soundness of the respective sentence in 4% of the sentences we checked. In a future iteration, it should be examined whether this was caused by an algorithmic shortcoming or by insufficient data and annotation quality.

Additionally, while keeping noun phrases intact represents a step toward introducing linguistically informed errors, further restrictions could be added – for instance, preserving additional syntactic structures or avoiding verb placements that would misrepresent subordinate clause boundaries, such as placing verbs in front of subordinations. Future versions of the algorithm could also take further advantage of the rich linguistic information already present in UD annotations to better approximate a wider range of learner phenomena. Another issue arises when the root of a sentence is not a verb but a noun, which can occur in some UD treebanks when the main clause contains a copula verb. In such cases, the corresponding noun phrase is currently not impermeable. Moreover, although the current implementation provides feedback on the position of verb placement errors, it does not yet explain why a given verb placement is correct or incorrect. Providing this type of explanatory feedback in the form of a feedback message would offer more meaningful support to learners. Lastly, it goes without saying that it would be desirable to evaluate our models on actual learner data once

natural learner data annotated for verb placement errors become available.

Ethics Statement

We do not see any major ethical concerns in the context of this work. As always, to promote diversity, it would be desirable to extend the coverage of our datasets to more Germanic languages, in particular, lower-resourced ones, minority languages, and regional language varieties such as Luxembourgish, Frisian, Yiddish, Low German, Swiss German, etc. As our algorithm operates on UD data, this would be implementable as soon as sufficient UD-annotated data for these languages become available. Of course, implementing an API demonstration of our trained verb placement error detection tools is a natural next step to ensure that language learners can directly benefit from the models we developed.

Acknowledgments

This research was supported by TrustLLM funded by Horizon Europe GA 101135671. We would like to thank Marco Kuhlmann and Marcel Bollmann for their support and guidance during the earlier stages of this project.

References

- Ásgrímur Angantýsson. 2021. [English-like V3-orders in matrix clauses in Icelandic](#). *Working Papers in Scandinavian Syntax*, 106:17–46.
- Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. [Context is key: Grammatical error detection with contextual word representations](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 103–115, Florence, Italy. Association for Computational Linguistics.
- 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 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.
- 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.
- Robert Dale and Adam Kilgarriff. 2011. [Helping our own: The HOO 2011 pilot shared task](#). In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249, Nancy, France. 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.
- Isidora Glisic and Anton Ingason. 2022. [The Nature of Icelandic as a Second Language: An Insight from the Learner Error Corpus for Icelandic](#). *CLARIN Annual Conference*, pages 23–33.
- Anton Karl Ingason, Þórunn Arnardóttir, Lilja Björk Stefánsdóttir, and Xindan Xu. 2021. [The Icelandic Child Language Error Corpus \(IceCLEC\) Version 1.1](#). CLARIN-IS.
- Anke Lüdeling, Seanna Doolittle, Hagen Hirschmann, Karin Schmidt, and Maik Walter. 2008. Das Lernerkorpus Falko. *Deutsch als Fremdsprache*, 2:67–73.
- 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](#). University of Tartu Library.
- 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.

Dan Nielsen. 2023. [ScandEval: A benchmark for Scandinavian natural language processing](#). In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 185–201, Tórshavn, Faroe Islands. University of Tartu Library.

Antje Orgassa. 2009. *Specific Language Impairment in a Bilingual Context: The Acquisition of Dutch Inflection by Turkish-Dutch Learners*. Ph.D. thesis, University of Amsterdam. Published by UtrechtLOT.

Sarah Schimke and Christine Dimroth. 2018. [The influence of finiteness and lightness on verb placement in L2 German: Comparing child and adult learners](#). *Second Language Research*, 34(2):229–256.

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, Tórshavn, Faroe Islands. LiU Electronic Press.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananeey, 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.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. [Neural network acceptability judgments](#). *Transactions of the Association for Computational Linguistics*, 7:625–641.

Marit Westergaard and Terje Lohndal. 2019. *Verb Second Word Order in Norwegian Heritage Language: Syntax and Pragmatics*. Georgetown University Press.

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. [Multi-class grammatical error detection for correction: A tale of two systems](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Zeman et al. 2024. [Universal Dependencies 2.15](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A Dataset Specifications

Language	Dataset	Split	# Sents	# C	# F	# O
af	AfriBooms	Train	1,300	2,617	2,615	25,341
		Dev	192	404	402	3,966
		Test	425	776	776	7,637
	GSD	Train	9,710	9,728	9,726	129,050
		Dev	613	697	694	6,730
		Test	648	737	737	7,740
de	HDT	Train	104,322	120,674	120,674	1,391,353
		Dev	12,234	14,105	14,104	159,956
		Test	12,758	14,497	14,497	167,172
	LIT	Test	1,784	2,657	2,656	27,315
	PUD	Test	738	1,053	1,052	10,864
nl	Alpino	Train	11,604	14,167	14,165	132,951
		Dev	697	777	775	8,536
		Test	557	827	827	7,947
	LassySmall	Train	10,389	14,063	14,060	167,415
		Dev	1,252	1,790	1,788	19,352
		Test	1,272	1,719	1,718	19,241
fo	FarPaHC	Train	1,015	2,284	2,283	13,993
		Dev	292	728	726	5,676
		Test	299	760	760	5,845
	OFT	Test	1,203	770	770	6,869
	GC	Train	3,908	6,707	6,706	58,746
		Dev	495	941	938	7,972
		Test	529	860	861	7,755
is	IcePaHC	Train	32,505	62,068	62,066	475,106
		Dev	4,579	12,379	12,375	94,062
		Test	4,965	11,982	11,983	95,484
	Modern	Train	2,659	5,660	5,660	45,435
		Dev	383	745	746	6,141
		Test	432	883	882	7,620
	PUD	Test	995	1,484	1,481	14,091
da	DDT	Train	3,989	6,186	6,188	55,403
		Dev	518	825	823	7,167
		Test	532	803	803	6,870
nb	Bokmaal	Train	14,120	19,775	19,773	171,489
		Dev	2,178	2,966	2,966	25,653
		Test	1,810	2,526	2,526	21,095
nn	Nynorsk	Train	12,718	18,464	18,461	177,112
		Dev	1,685	2,429	2,425	22,475
		Test	1,337	1,858	1,858	18,115
sv	LinES	Train	3,008	4,672	4,669	39,408
		Dev	989	1,562	1,559	13,327
		Test	969	1,438	1,437	11,969
	PUD	Test	992	1,343	1,341	14,502
	Talbanken	Train	3,909	4,769	4,768	47,995
		Dev	478	709	708	7,263
		Test	1,140	1,573	1,571	14,798
Total			275,607	381,884	381,816	3,795,723

Table 5: Datasets per language and number of sentences (# Sents) as well as number of labels per data split.

B MERLIN and Falko Search Queries

Examples (12), (13), and (14) show the queries we used when looking for all errors, movement errors, and movement errors involving verbs in both MERLIN and Falko, respectively. Table 6 shows the exact error numbers. We show the MERLIN queries. The Falko queries are essentially equal with only minor differences in category names.

(12)

TH1Diff!=/(CHA)|(CHA\x2FSPLIT)/

(13)

TH1Diff=/(MOVS)|(MOVT)|(MOVS\x2FCHA)|
(MOVT\x2FMERGE)|(MOVS\x2FSPLIT)|
(MOVT\x2FCHA)/

(14)

TH1Diff=/(MOVS)|(MOVT)|(MOVS\x2FCHA)|
(MOVT\x2FMERGE)|(MOVS\x2FSPLIT)|
(MOVT\x2FCHA)/
& tok_pos=/(VAFIN)|(VAIMP)|(VAINF)|
(VMFIN)|(VMINF)|(VVFIN)|(VVIMP)|
(VVINF)|(VVPP)|(VVIZU)|(VAPP)/
& #1=_#2

Errors / Corpus	MERLIN	Falko
All grammatical errors	14,366	8,535
Movement errors	4,880	2,526
Movement errors with verb	591	315

Table 6: Movement errors across MERLIN and Falko corpora.

C Model Training Parameters

Hyperparameter	Value
Number of training epochs	3
Training batch size	16
Evaluation batch size	16
Weight decay	0.01
Learning rate	5e-5
Checkpoint saving strategy	Per epoch
Evaluation strategy	Per epoch
Max number of saved checkpoints	1
Metric for best model selection	Evaluation loss
Load best model at end	True

Table 7: Model training parameters.

Enhancing Security and Strengthening Defenses in Automated Short-Answer Grading Systems

Sahar Yarmohammadtoosky¹ Yiyun Zhou² Victoria Yaneva²

Peter Baldwin² Saed Rezaei² Brian Clauser² Polina Harik²

¹School of Data Science and Analytics, Kennesaw State University

²National Board of Medical Examiners (NBME)

yarmohamadishr@gmail.com

YYZhou@nbme.org, VYaneva@nbme.org, PBaldwin@nbme.org

SRezaeidemne@nbme.org, beclauser@gmail.com, PHarik@nbme.org

Abstract

This study examines vulnerabilities in transformer-based automated short-answer grading systems used in medical education, with a focus on how these systems can be manipulated through adversarial gaming strategies. Our research identifies three main types of gaming strategies that exploit the system’s weaknesses, potentially leading to false positives. To counteract these vulnerabilities, we implement several adversarial training methods designed to enhance the system’s robustness. Our results indicate that these methods significantly reduce the susceptibility of grading systems to such manipulations, especially when combined with ensemble techniques like majority voting and Ridge regression, which further improve the system’s defense against sophisticated adversarial inputs. Additionally, employing large language models such as GPT-4 with varied prompting techniques has shown promise in recognizing and scoring gaming strategies effectively. The findings underscore the importance of continuous improvements in AI-driven educational tools to ensure their reliability and fairness in high-stakes settings.

1 Introduction

As technology advances, automated scoring of free-text responses is transforming how we evaluate written answers, making the process faster and more consistent (Yannakoudakis et al., 2011). Early research in this area has focused on instance-based methods, treating the task as a supervised text classification problem (Burrows et al., 2015), (Bai and Stede, 2023). In this approach, models are trained using labeled data to predict labels for unseen data, such as predicting whether a short answer submitted to an Automated Short Answer Grading (ASAG) system is correct or incorrect (Bonthu et al., 2021). More recently, some ASAG systems have taken a similarity-based approach, where each

new response is assigned the label of the response it most closely matches from a sample of previously annotated responses. Neural similarity-based models have further advanced this field by learning rich response (or question-response) embeddings and matching them using cosine similarity, demonstrating superior performance in capturing meaning beyond surface-level text (Schneider et al., 2022).

Despite the significant potential demonstrated by similarity-based ASAG models, these models are especially vulnerable to scoring errors when presented with certain kinds of responses (Section 2). This creates an opportunity for examinees to exploit these vulnerabilities to earn undeserved credit, which can erode trust in automated grading and raise concerns over the responsible use of AI in educational assessments. Deliberate attempts by examinees to exploit ASAG systems in this way are known as “gaming strategies.”

The objective of this study is to identify and analyze potential gaming strategies that students may use to manipulate or deceive automated short-answer grading (ASAG) systems, particularly within medical education. To counteract these vulnerabilities, we propose a dual approach combining: (1) adversarial training and ensemble techniques—such as majority voting and Ridge regression—applied to a transformer-based ASAG system (ACTA), and (2) prompt engineering techniques applied to a large language model (GPT-4) to evaluate its ability to detect and mitigate gaming attempts. This dual framework allows us to examine the effectiveness of both system-level defenses and LLM-based scoring interventions in improving accuracy and reducing false positive rates (FPR) when presented with adversarial inputs. We evaluate the robustness of these methods before and after the proposed defenses are applied. This investigation is guided by three research questions:

1. How vulnerable are transformer-based grad-

Component	Description
Stem	A previously healthy 26-year-old man is brought to the emergency department because of a tingling sensation in his fingers and toes for 3 days and progressive weakness of his legs. He had an upper respiratory tract infection 2 weeks ago. He has not traveled recently. He was unable to get up from bed this morning and called the ambulance. Temperature is 37.3°C (99.1°F), pulse is 110/min, respirations are 22/min, and blood pressure is 128/82 mm Hg. Pulse oximetry on room air shows an oxygen saturation of 99%. Physical examination shows weakness of all four extremities in flexion and extension; this weakness is increased in the distal compared with the proximal muscle groups. Deep tendon reflexes are absent throughout. The sensation is mildly decreased over both feet.
Lead-in	What is the most likely diagnosis?
Sample Correct Answers	Guillain-Barré syndrome; acute immune-mediated polyneuropathy

Table 1: The parts of a short-answer question in the medical domain.

- ing systems to adversarial gaming strategies used by test takers?
2. What effect do adversarial training and ensemble methods have on system robustness?
 3. How effective are different prompt engineering strategies in identifying and mitigating adversarial inputs?

This study advances the ASAG field by addressing the critical issue of vulnerability to adversarial gaming strategies. By identifying such strategies and developing effective countermeasures, the robustness, integrity, and reliability of transformer-based short-answer grading systems can be improved. The reported findings have broad practical benefits including improving the trustworthiness of automated grading tools in educational settings and contributing to the security of AI-driven systems against adversarial attacks. The technical advancements that are reported are also complemented by theoretical insights into the challenge posed by gaming in the context of ASAG specifically as well as into the responsible use of AI in education more generally.

2 Related Work

With the advent of transformer models, neural similarity-based ASAG techniques have demonstrated improved accuracy and reduced data annotation requirements compared to instance-based methods (Bexte et al., 2023). However, these advancements have also introduced new challenges, particularly the susceptibility of similarity-based systems to adversarial attacks (Filighera et al., 2020). Such attacks can range from submitting random strings of letters (Ding et al., 2020) to adding irrelevant yet carefully chosen words to otherwise valid responses (Filighera et al., 2023), with the

goal of deceiving the model into misclassification. For example, Ding et al. (2020) found that a non-sensical string like "nswvtnvakgxpnm" could be classified as a correct response by an ASAG system.

Within the medical domain, Baldwin et al. (2025) have shown that several gaming strategies were successful in "deceiving" a similarity-based system. These strategies consisted of entering the following as responses to the short-answer questions: (1) random number of words selected at random from the stem¹, (2) random number of consecutive words selected at random from the stem, (3) random number of medical terms selected at random from the stem, (4) keywords selected from the stem by a content expert, and (5) a summary of the stem produced by GPT 3.5, as well as (6) listing multiple responses only one of which is correct. The results showed that the first five strategies lead to a success rate between 6% to 16%, while the last strategy led to a success rate of 57%, underscoring the need for addressing these vulnerabilities.

While prior work defined the problem of gaming strategies and quantified their effects on transformer-based scoring systems, this study focuses on systematically evaluating multiple adversarial training techniques and ensemble strategies to enhance system resilience within the clinical ASAG domain. Additionally, we explore the role of LLMs, such as GPT-4, in detecting and mitigating adversarial manipulation.

3 Methodology

This study investigates two approaches for defending against gaming strategies in automated short-answer grading (ASAG) systems. The first ap-

¹An item stem is the part of a test question that presents the problem or scenario to be answered or responded to, as shown in Table 1.

proach centers on the ACTA system, a transformer-based model that classifies short medical responses as correct or incorrect by leveraging sentence-BERT embeddings and a similarity-based matching mechanism. This approach is trying to enhance the robustness of the ACTA system through adversarial training and ensemble methods. The second approach involves a large language model (LLM)-based method, where GPT-4 is used with various prompt engineering techniques to independently score student responses and detect gaming strategies. This allows us to examine the effectiveness of both system-level defenses and LLM-based scoring interventions.

3.1 ACTA System Overview

Experiments were undertaken using the ACTA system (Analysis of Clinical Text for Assessment; Suen et al. (2023)), a transformer-based ASAG system designed to classify short responses to medical questions as correct or incorrect. To achieve this, ACTA utilizes sentence BERT (Reimers and Gurevych, 2019) and contrastive learning. When presented with a new response, ACTA matches it to the most similar response within a training set of human-scored responses and assigns it the matched response's label (correct or incorrect), provided their similarity exceeds a given operational threshold (for a detailed description of ACTA, see Suen et al. (2023)). While ACTA achieves near human-level performance with a binary F1 score of .98, previously reported weaknesses of transformer-based grading systems require an investigation of ACTA's susceptibility to gaming.

We evaluate the effectiveness of adversarial training by assessing the ACTA system's performance on gaming data both before and after the training is applied.

3.2 Prompt Engineering with GPT-4

Using large language models to score the real dataset has already shown promising results. This motivated the use of these models with different prompting techniques to evaluate whether large language models can accurately recognize and score gaming responses. Due to the consistently strong performance demonstrated by ChatGPT4 (Achiam et al., 2023) across various experimental settings, this model was selected as the primary tool for conducting this series of experiments.

To evaluate system robustness, we simulate gaming strategies that students might use to deceive

ASAG systems—detailed in Section 4. These adversarial examples are used both to adversarially train the ACTA model and to test the effectiveness of prompt engineering with GPT-4.

4 Experiment Design

4.1 Dataset

The dataset comprises 71 short-answer questions (SAQs) with 36,735 responses from 24,235 examinees. An example of an SAQ is shown in Table 1. Responses were collected during the administration of a Medicine Clinical Science subject exam distributed to a large number of medical schools in the US and Canada for use as a summative, end-of-semester exam.

4.2 Gaming Strategies Simulation

Following Baldwin et al. (2025), we simulate three gaming strategies meant to resemble how students *without* the requisite knowledge of a correct answer might nevertheless respond to an item. Data were generated as follows:

1. Simulate responses by randomly sampling words (excluding stop words) from a given item's clinical vignette. Variations of this strategy include consecutive words, non-consecutive words, and samples of words that appear in both the item description *and* a generic list of medical terms.
2. Utilize a summary of the clinical scenario as a response. Summaries were obtained using ChatGPT.
3. Utilize "mixed" responses that combine both correct and plausible incorrect answers into a single response, which, following operational guidelines, should be scored as incorrect.

For our data, the strategies generated an impractically large number of responses. To create a set of responses that could feasibly be used as part of an operational process, we randomly sample 5% from each strategy, resulting in 14,657, 573, and 584 simulated responses for strategies 1, 2, and 3, respectively. While simulated responses were largely nonce phrases or unequivocally incorrect, 3 simulated responses exactly matched (real) correct responses from the training data. Three misclassifications were deemed tolerable for our purpose, and all artificial responses were designated as incorrect.

Following a principal component analysis (PCA), Figures 2 and 3 plot the responses for two

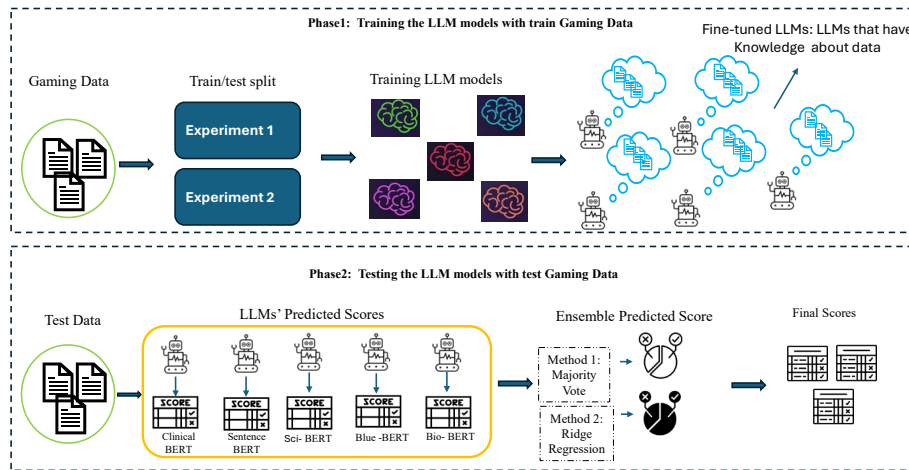


Figure 1: Adversarial Defense Workflow. Gaming data is combined with real data for training and testing purposes.

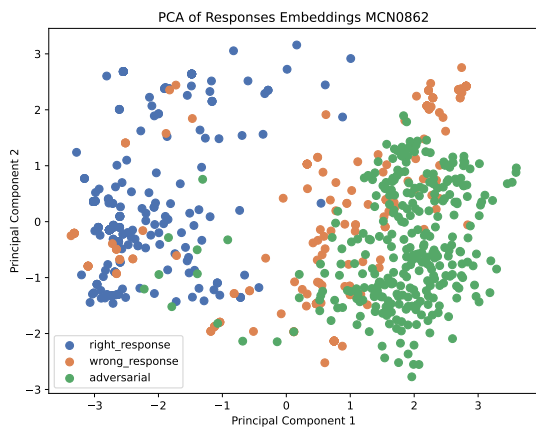


Figure 2: PCA of Response Embedding for Item 1

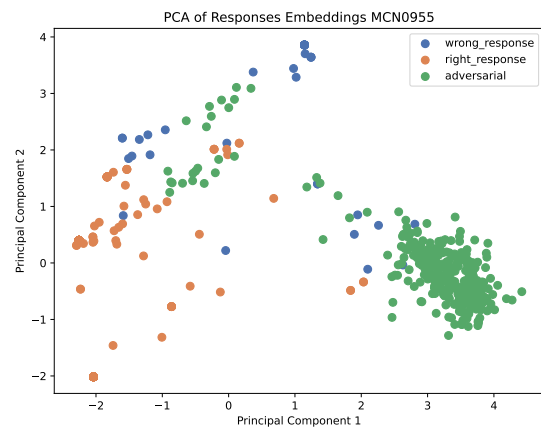


Figure 3: PCA of Response Embedding for Item 2

SAQs in the space defined by principal components 1 and 2. Differences in the identification of adversarial examples across items can be observed. For SAQ 1 (Figure 2), the distribution of gaming responses shares considerable overlap with the distribution of correct responses, suggesting that gaming responses may have a relatively high probability of being misclassified. In contrast, for SAQ 2 (Figure 3), gaming responses are comparatively isolated, suggesting that these responses may be more readily identified by an ASAG system.

4.3 Adversarial Training Setup for ACTA

To enhance the resilience of the ACTA system against gaming responses, two adversarial training experiments were undertaken to investigate (i) whether adversarial training based on all three types of gaming responses improves system robustness to these types of responses and (ii) whether adversarial training based on two types of gaming responses

improves robustness to a third type of responses. The general workflow is demonstrated in Figure 1. The first experiment entailed the inclusion of 70% of the simulated responses from each strategy into the training dataset (together with the authentic responses), with the remaining 30% of both artificial and authentic responses allocated to the test set. The objective of the second experiment was to assess the capacity of data derived from specific strategies to bolster the model's defenses against gaming strategies that were not identified during the training phase. This was achieved through the implementation of a 3-fold cross-validation method, where the model was trained on data from two gaming strategies and tested on the third. This approach enabled the evaluation of the model's enhanced ability to recognize unknown examples through exposure to known gaming adversarial examples.

To enhance the results and evaluate the efficacy

of various models, we employed five different models for response embeddings to predict whether a response was related to gaming: Clinical-BERT (Huang et al., 2019), Bio-BERT (Lee et al., 2020), Sci-BERT (Beltagy et al., 2019), and Blue-BERT (Peng et al., 2019). These models are pretrained on medical domain datasets, leveraging their specialized knowledge to aid in training the system. They were fine-tuned with the adversarial data in a 70/30 split (for experiment 1) and fine-tuned with two gaming strategies, and tested on the third one for experiment 2, as detailed above. These fine-tuned models are then used to classify responses as correct or incorrect. The embeddings generated by these models were then combined with the ACTA model using a majority vote method and ridge regression to determine if there was an improvement.

4.4 Prompt Engineering Setup for GPT-4

Three prompting techniques were employed in this experiment:

1. The model was provided with the item questions and the examinee’s response to the question. The model was then asked to score the response, given the question.
2. The model was given the questions along with examples of correct answers for each question. The model was then asked to score the examinee’s response.
3. The model was provided with examples of correct answers only, and then asked to score the examinee’s response.

Using ChatGPT-4, scores using each of these strategies were obtained. Due to resource limitations, 100 samples of each gaming data and real data were used for these experiments.

5 Results

5.1 ACTA Pre-Adversarial Training Results

We began by evaluating ACTA’s scoring of gaming responses prior to any adversarial training. The model was trained on 70% (26,095) of the real responses and evaluated on the remaining 30% (10, 890) combined with all artificial responses. Since the number of simulated gaming responses varies across strategies and experiments, we report two separate measures: F1 for real responses and false positive rate (FPR) for artificial responses. ACTA performed well when scoring real data (F1 = .9845); however, the gaming strategies deceived

ACTA into misclassifying many of the artificial responses as “correct.” FPRs for strategies 1, 2, and 3 were .061, .189, and .435, respectively, demonstrating the vulnerability of this system to examinee gaming (Table 2). Responses from strategy 3 were especially challenging to classify correctly, illustrating the potential for examinees with partial knowledge to game systems that have not been adversarial trained by simply listing as many plausible answers as possible.

5.2 ACTA Post Adversarial Training Results

The results from the experiments described above are shown in Figures 5 and 6 and Tables 3 and 4. In the first experiment, the model maintained a high F1 score, with substantial reductions in FPRs across various gaming strategies and embedding models. This demonstrates the efficiency of adversarial training in enhancing model accuracy. The FPR results for the gaming strategy “Information from the Stem” were consistently the lowest across models, indicating that even without adversarial training, this model recognized these responses better than the other two gaming strategies. The post-adversarial training gains for the “Mixed Responses” strategy are particularly encouraging, suggesting that training on simulated gaming responses is an effective countermeasure against the most successful gaming strategy. This highlights the significant benefits of adversarial training for defending against complex adversarial attacks.

The second experiment also maintained a high F1 score of 0.98 for real responses, while still providing some improvements with gaming detection. These results suggest that familiarity with known gaming strategies helps the model recognize responses based on unknown gaming strategies, enhancing overall robustness. The model’s resilience is significantly bolstered by training with ‘strong’ gaming examples (high FPR) instead of ‘weak’ ones. The model’s performance was least effective under strategy 3; however, incorporating this strategy into adversarial training markedly improved model efficiency against strategies 1 and 2. In contrast, training with the relatively weaker strategies 1 and 2 yielded lesser improvements in detecting strategy 3, reducing the FPR from 0.435 to 0.067, which is the smallest FPR for the ACTA model in the second experiment among all the models. This observation highlights the intricate relationship between the effectiveness of gaming strategies and the robustness of model training, suggest-

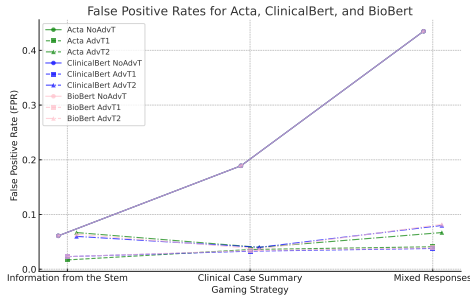


Figure 4: FPR Across Different Models - Part 1

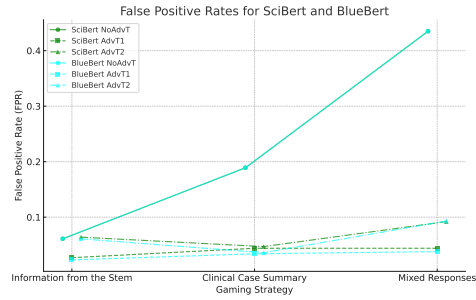


Figure 5: FPR Across Different Models - Part 2

Gaming Strategy	FPR Before Adv Training	FPR Adv Training #1	FPR Adv Training #2
Information from the Stem	.061	.017	.067
Clinical Case Summary	.189	.036	.04
Mixed Responses	.435	.041	.067

Table 2: False positive rates for the gaming responses before and after adversarial training

Gaming strategy	Acta Model			ClinicalBert		
	NoAdvT	AdvT1	AdvT2	NoAdvT	AdvT1	AdvT2
Information from stem	0.061	0.017	0.067	0.061	0.023	0.060
Clinical case summary	0.189	0.036	0.040	0.189	0.033	0.040
Mixed responses	0.435	0.041	0.067	0.435	0.038	0.080

Gaming strategy	BioBert			SciBert		
	NoAdvT	AdvT1	AdvT2	NoAdvT	AdvT1	AdvT2
Information from stem	0.061	0.023	0.064	0.061	0.027	0.064
Clinical case summary	0.189	0.034	0.037	0.189	0.044	0.047
Mixed responses	0.435	0.039	0.082	0.435	0.044	0.092

Gaming strategy	BlueBert		
	NoAdvT	AdvT1	AdvT2
Information from stem	0.061	0.023	0.061
Clinical case summary	0.189	0.034	0.036
Mixed responses	0.435	0.038	0.093

Table 3: False positive rates for gaming responses before and after adversarial training using various models.

Gaming strategy	Majority Vote Model			Ridge Regression		
	FPR (NoAdvT)	FPR (AdvT1)	FPR (AdvT2)	FPR (NoAdvT)	FPR (AdvT1)	FPR (AdvT2)
Information from stem	0.061	0.015	0.053	0.061	0.014	0.040
Clinical case summary	0.189	0.029	0.033	0.189	0.029	0.029
Mixed responses	0.435	0.035	0.076	0.435	0.035	0.068

Table 4: False positive rates for the gaming responses before and after adversarial training using Majority Vote and Ridge Regression models

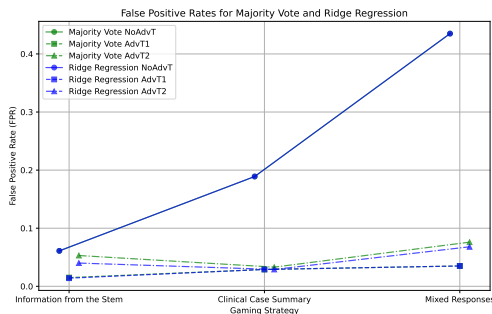


Figure 6: False Positive Rates for Majority Vote and Ridge Regression

Gaming strategy	Accuracy	TNR	FPR
Information from stem	0.89	0.89	0.11
Clinical case summary	0.97	0.97	0.03
Mixed responses	0.99	0.99	0.01

Table 5: ChatGPT results for the gaming responses

ing a positive correlation where more sophisticated adversarial training leads to improved robustness. Table 4 shows the FPR results after applying the embedding models’ results to Majority Vote and Ridge Regression (“AdvT2”). Ridge regression outperformed the majority vote with the FPRs for both experiments 1 and 2 (Figure 6). These findings suggest the effectiveness of these two models compared to considering an individual embedding model. Similar to the embedding model FPR results, gaming strategy 3 is more challenging to recognize and has a higher FPR than gaming strategies 1 and 2. However, FPR results for all gaming strategies are improved compared to each embedding model’s results.

5.3 Prompt Engineering Results

Summary results from the prompt engineering experiments are shown in Table 5. Because it performed best overall, only results from the first

prompting strategy, submitting a question and a response and requesting a score, are reported.

For the experiment with real data, the model maintained high performance, with an accuracy of 0.93, a precision of 0.97, and False Positive Rate (FPR) of 0.06. For the gaming data, the highest accuracy was achieved for the third gaming strategy, “submit multiple answers”, with an accuracy of 0.99, a TNR of 0.99, and the lowest FPR of 0.01. This suggests that in this experiment, ChatGPT-4 was more successful in recognizing and scoring responses generated using the third strategy compared to the adversarial training approaches reported above. The second gaming strategy, “summarize item vignette”, also performed strongly with an accuracy of 0.97 and an FPR of 0.028. The first strategy, “copy words from the item vignette”, had the lowest performance among the gaming strategies, with an accuracy of 0.89 and an FPR of 0.11. These results underscore the model’s effectiveness in handling various gaming strategies, with notable success in the third strategy, and relative ineffectiveness with responses from the first strategy.

6 Error Analysis

6.1 Error Analysis For Adversarial Training

The model’s performance was notably better for gaming strategies it was trained on compared to those it had not encountered during training. This points to a potential overfitting issue, where the model becomes too specialized in detecting known adversarial patterns but may struggle with novel or unseen strategies. Despite the reductions in FPRs, some gaming strategies, particularly Strategy 3 (“Mixed Responses”), remained challenging for the model to detect. This suggests that while adversarial training improves the model’s defenses, it may not fully mitigate all vulnerabilities, especially for more sophisticated or nuanced gaming strategies. The quality and representativeness of the adversarial examples used in training had a sig-

nificant impact on the model’s performance. Training with “strong“ adversarial examples (those with high FPRs) led to more substantial improvements in robustness, whereas training with “weaker“ examples provided less benefit. This underscores the importance of carefully selecting adversarial examples that accurately reflect the types of gaming strategies the model might encounter in real-world applications. The cross-validation experiments demonstrated that while training on multiple gaming strategies can enhance the model’s generalization capabilities, the process is complex and computationally expensive.

6.2 Error Analysis For Prompt Engineering

Upon reviewing the rationales across various datasets, several common patterns emerged that explain why certain responses were predicted wrong.

Summary of the clinical scenario: Many rationales indicate that a response “aligns with the intended correct pattern“ or “matches the expected correct response.“ This suggests that the system recognizes patterns it anticipates, regardless of the response’s accuracy. For instance, if a response correctly lists all the symptoms of a disease, the model may consider it correct simply because it aligns with the expected diagnosis related to those symptoms. Some rationales reveal that the presence of specific keywords in the responses triggers the system to mark it as correct, e.g., phrases like “man, 36, suffers sleepiness, ED, weight gain, hypertension“ match key descriptors associated with the correct answers, such as “sleep study.“

If a response mentions symptoms that suggest a disease, the model may consider it correct, even if the actual cause of the disease differs. An example would be a response stating, “Man on anti-malaria drugs shows signs of hemolysis,“ where the correct answer is “Hemolysis due to G6PD deficiency“. In this case, because the hemolysis disease was mentioned in the response, the model scored this response as correct.

Utilize mixed responses: Here, the rationales often point to specific phrases within the response, indicating that the model matches exact or nearly exact phrases it expects, regardless of whether the combination is logically sound. If a response includes correct elements alongside irrelevant parts that do not negate the correct diagnosis, the model may still consider it correct. For instance, “Rheumatic fever“ might be irrelevant, but it does not invalidate the correct diagnosis of

“systemic sclerosis (scleroderma).“ Sometimes, the model assesses the overall picture of the response; if a disease shares similarities with another mentioned in the response, it may still be considered correct. For example, the response “chronic obstructive pulmonary disease bronchiectasis“ might be deemed correct because “bronchiectasis“ was the intended correct answer, and it shares similarities with “chronic obstructive pulmonary disease bronchiectasis“.

Randomly sampling words: This strategy involves the use of random words; in cases where the model erroneously produces a correct score, the sample words are general and provide no specific clues about the disease. In such cases, the model relies on the question and uses the information provided to predict the disease, ultimately considering the response correct, although there was not any correct information in the response.

Real Dataset: If a response contains minor misspellings, the model may consider it incorrect, even if it matches the correct response. Conversely, the model may consider a vague term correct if it encompasses the specific diagnoses listed. For example, the response “heart disease“ might be accepted as correct, even if the correct answer is a specific type of heart failure or disease. The rationales sometimes rely on broad medical logic. The model might still consider it correct when a response refers to a general disease category without specifying details or subcategories. This suggests that the model applies standard medical reasoning but may lack the subtlety needed to distinguish between similar conditions. In some cases where the general concept is correct but details are slightly different, the model may still mark the response as wrong despite its correctness. These patterns indicate that the model prioritizes exact matches and penalizes variations, even when the overall concept is correct, highlighting its limitations in understanding nuanced or slightly varied responses.

7 Discussion

These results add new evidence related to exploitable vulnerabilities in transformer-based grading systems. Despite being artificially generated approximations of potential gaming behaviors, all three gaming strategies were successful in deceiving the non-adversarial trained system. This aligns with findings from previous research, which also reported that adversarial approaches could compro-

mise the integrity of automated systems, particularly when the system is not specifically trained to recognize such attacks (Baldwin et al., 2025). The first group of adversarial training experiments showed that data augmentation is a promising way to fortify ASAG systems against such attacks. The cross-validation experiments also showed that it is beneficial to train on examples across gaming strategies, suggesting a transfer of learning between strategies, which holds the potential to protect against unforeseen gaming tactics that may arise in practice.

The results show that incorporating embedding models into Majority Vote and Ridge Regression significantly reduced the false positive rates (FPR) in experiments, which is in line with findings from research on ensemble learning methods that demonstrate their superiority in reducing error rates (Naderalvojud and Hernandez-Boussard, 2023). Among the gaming strategies evaluated, strategy 3 proved to be the most challenging to recognize, yielding higher FPRs than strategies 1 and 2. Despite this, the FPR results across all gaming strategies showed improvement when compared to the results of each individual embedding model. This mirrors findings in previous studies, where certain adversarial strategies consistently posed greater challenges to detection systems.

The experiments demonstrated the first prompting strategy was effective, where the model was given questions and responses to score. With real data, the model showed high accuracy and precision and a low FPR, indicating robust performance in evaluating genuine responses. For gaming data, the best results were seen in the “submit multiple answers” strategy (consistent with Baldwin et al. (2025)). The “summarize item vignette” strategy also performed well; however, the “copy words from the item vignette” strategy performed relatively poorly.

In summary, while the non-adversarially trained system was susceptible to gaming, the defense mechanisms explored in this paper showed significant reduction in FPR both when training within strategy and across strategies. As the understanding of possible gaming strategies in the context of medical education matures, future work will include the simulation of new adversarial attacks for ASAG systems that are more closely aligned with human behaviors as well as further experimentation with adversarial training. Employing regularization techniques such as dropout, weight decay, and early

stopping can limit overfitting, which may improve a model’s generalizability. Furthermore, employing various prompt engineering techniques with LLMs also has the potential to enhance performance.

8 Limitations and Ethical Considerations

While this study provides promising directions for improving robustness in ASAG systems, several limitations must be acknowledged. First, the adversarial examples used in our experiments are simulated approximations of gaming strategies, rather than authentic, organically derived examples from real-world test-takers. As such, while the strategies are plausible and their effectiveness in gaming the scoring system was proven, they may not fully reflect the diversity and nuance of actual test-taker behaviors, particularly in high-stakes environments. Furthermore, the experiments were conducted within a single domain and dataset, and the generalizability of the findings to other domains—such as legal education, K-12, or general writing assessment—remains uncertain. Different domains may involve distinct response styles, expectations, learner populations, and gaming behaviors, which could impact the effectiveness of adversarial training strategies. Last but not least, this study explored the effects of these gaming strategies on the ACTA scoring system and on using GPT-4 to score responses via prompt engineering. The extent to which these results generalize to other transformer-based or few-shot scoring systems is an open question.

From an ethical standpoint, adversarial training raises important questions related to fairness, transparency, and trust in AI-based scoring. While improving robustness is a core goal, it is also critical to ensure that ASAG systems do not unfairly penalize legitimate test-taking strategies or linguistic variability, especially among non-native speakers or individuals from underrepresented groups.

It should also be recognized that research into gaming strategies inherently raises concerns about dual-use. While our intention is to strengthen the integrity of ASAG systems, the publication of methods for generating adversarial responses could inadvertently aid malicious actors. To mitigate this risk, we have intentionally abstracted implementation details and focused on generalizable insights rather than system-specific exploits.

On the positive side, adversarial examples can serve an additional purpose in enhancing explain-

ability. When used in conjunction with feature attribution methods, adversarial perturbations can help identify which aspects of a response most influence model predictions. For example, if minor lexical changes significantly affect scoring, it may indicate an over-reliance on specific keywords or surface features rather than deeper semantic understanding. For example, the error analyses of the prompt engineering approach revealed that the models tend to recognize anticipated patterns as a proxy to accuracy, which is what makes them particularly susceptible to gaming responses that follow the expected pattern of correct answers. These insights are critical for diagnosing model weaknesses, refining scoring rubrics, and improving transparency. In high-stakes assessment, the ability to explain and justify model decisions is essential for fostering user trust and ensuring accountability in automated assessment.

Overall, while adversarial training is a valuable tool for increasing the reliability of ASAG systems, its application must be guided by ethical principles that prioritize fairness, interpretability, and alignment with educational values.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- 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.
- Peter Baldwin, Victoria Yaneva, Kai North, Le An Ha, Yiyun Zhou, Alex J Mechaber, and Brian E Clouser. 2025. The vulnerability of ai-based scoring systems to gaming strategies: A case study. *Journal of Educational Measurement*.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.
- 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 *Machine Learning and Knowledge Extraction: 5th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2021, Virtual Event, August 17–20, 2021, Proceedings 5*, 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*, 25:60–117.
- Yuning Ding, Brian Riordan, Andrea Horbach, Aoife Cahill, and Torsten Zesch. 2020. Don’t take “nswvtnvakgxp” 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.
- Anna Filighera, Sebastian Ochs, Tim Steuer, and Thomas Tregel. 2023. Cheating automatic short answer grading with the adversarial usage of adjectives and adverbs. *International Journal of Artificial Intelligence in Education*, pages 1–31.
- Anna Filighera, Tim Steuer, and Christoph Rensing. 2020. Fooling automatic short answer grading systems. In *International conference on artificial intelligence in education*, pages 177–190. Springer.
- Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.
- 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.
- Behzad Naderalvojud and Tina Hernandez-Boussard. 2023. Improving machine learning with ensemble learning on observational healthcare data. In *AMIA Annual Symposium Proceedings*, volume 2023, page 521. American Medical Informatics Association.
- Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Johannes Schneider, Robin Richner, and Micha Riser. 2022. Towards trustworthy autograding of short, multi-lingual, multi-type answers. *International Journal of Artificial Intelligence in Education*, pages 1–31.
- King Yiu Suen, Victoria Yaneva, Janet Mee, Yiyun Zhou, Polina Harik, et al. 2023. Acta: Short-answer grading in high-stakes medical exams. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 443–447.

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.

EyeLLM: Using Lookback Fixations to Enhance Human-LLM Alignment for Text Completion

Astha Singh¹, Mark Torrance², Evgeny Chukharev¹

¹Iowa State University, ²Nottingham Trent University
asthas@iastate.edu, mark.torrance@ntu.ac.uk, evgeny@iastate.edu

Abstract

Recent advances in LLMs offer new opportunities for supporting student writing, particularly through real-time, composition-level feedback. However, for such support to be effective, LLMs need to generate text completions that align with the writer’s internal representation of their developing message, a representation that is often implicit and difficult to observe. This paper investigates the use of eye-tracking data, specifically lookback fixations during pauses in text production, as a cue to this internal representation. Using eye movement data from students composing texts, we compare human-generated completions with LLM-generated completions based on prompts that either include or exclude words and sentences fixated during pauses. We find that incorporating lookback fixations enhances human-LLM alignment in generating text completions. These results provide empirical support for generating fixation-aware LLM feedback and lay the foundation for future educational tools that deliver real-time, composition-level feedback grounded in writers’ attention and cognitive processes.¹

1 Introduction

Natural language processing (NLP) solutions exist for scaffolding students who are learning to produce effective text. These support both surface-level accuracy (grammar and spelling), and also compositional-level effectiveness, i.e. helping students produce text that communicates a coherent message (e.g., Franzke et al., 2005; Roscoe and McNamara, 2013). Recent advances in large lan-

guage models (LLMs) enable promising innovative applications for intelligent support of writing tasks. Specifically, there are potential advantages to providing composition-level feedback in real time, while the writer is still forming their message, rather than retrospectively, once their text is complete.

Achieving this is challenging, but is brought within reach by LLMs. These can generate plausible completions to text that a student is in the process of composing. However, providing feedback based on these plausible completions has limited learning benefit unless the LLM-generated completions are aligned with those that the student intended to produce. Human-LLM alignment will increase if the LLM captures important features of the writer’s current internal representation of their developing message. However, these mental representations are not directly observable. They are also likely to be implicit, at least in part: The writer might not have explicitly articulated their developing message even in their own internal representation (Torrance, 2016).

One possible clue to this implicit internal representation is provided by writers’ eye movement. During text production writers frequently hesitate, often very briefly, and look back within their own text. These lookback eye movements typically involve “hopping around” between isolated words and phrases rather than sustained reading (Chukharev-Hudilainen et al., 2019; Torrance et al., 2016). This eye movement is, however, targeted: Words are not fixated at random, but tend to be informationally rich. Previous work in cognitive psychology has suggested that lookbacks may be driven by the writer’s internal representation of the emerging message (Torrance, 2016; Torrance et al., 2016), but this hypothesis has not been systematically evaluated.

In this paper, we propose the use of eye-tracking cues to enhance LLM performance in predicting

¹Following the initial submission, we discovered an error in one of the analysis scripts that inadvertently introduced data contamination. To ensure the validity of the findings, all models were rerun using corrected code. This version reports the updated results. While specific numerical values have changed, the main conclusions of the study remain unaffected. Our code is available publicly at <https://go.chukharev.com/bea-2025>.

text completion. If eye fixations cue content for what the writer produces next, then lookback data can help provide completion suggestions that align more closely with the writer’s current thinking. To test this hypothesis, we use keystroke and eye movement data from human writers composing argumentative texts. We extract hesitation events: pauses when writers stopped and looked back into their text and then, without editing, continued writing (e.g., finishing the sentence that they were writing before the pause). We compare writers’ own completions with LLM completions generated on the basis of prompts that did and did not include the words and sentences that the writers fixated on during lookback. Increased overlap between LLM and writer completions when prompts incorporate information from lookbacks would be evidence for the potential value of eye-movement-informed message-level scaffolding of written composition.

The purpose of this paper is two-fold: First, we evaluate whether the information on the writer’s lookback fixations can enhance the alignment between the human and the LLM in the text completion task. Second, we investigate whether LLM text completions with and without eye movement data can provide evidence for the (cognitive) hypothesis about the role of lookbacks in human text production. This lays the necessary groundwork for designing novel educational applications wherein useful composition-level feedback can be provided to students in real time, before the text is completed by the student.

2 Related Work

Functions of reading during writing in humans.

Research in cognitive psychology suggests that writers often look back at their own text during pauses in production, particularly near sentence boundaries. These fixations frequently involve lexical processing rather than simple error-checking. Most are not part of sustained reading sequences but instead consist of gaze shifts among isolated words via forward and backward saccades. These lookbacks are thought to support the planning of upcoming text rather than merely identifying mistakes in previously written content (Chukharev-Hudilainen et al., 2019; Torrance et al., 2016).

Human–LLM Alignment. Recent efforts to enhance alignment between humans and large language models (LLMs) in writing support systems have focused on modeling writers’ intentions and

cognitive states (Zhang et al., 2024; Gero et al., 2022). However, these internal intentions are often difficult to directly observe. Looking back into existing text, in addition to supporting error monitoring, is likely to support ongoing text production, cuing both message and linguistic (lexical, syntactic) form for what the writer will say next (Chukharev-Hudilainen et al., 2019; Torrance et al., 2016). Knowledge of what words and sentences a writer fixates during these lookbacks, therefore, may provides insight into the writer’s evolving mental representation of their developing composition.

While some recent approaches have explored aligning LLM-generated suggestions with user intentions (Reza et al., 2025), most have not incorporated real-time behavioral signals. Our work builds on this line of inquiry by explicitly integrating gaze-based cues into prompting strategies, aiming to improve alignment between LLM completions and the writer’s unfolding mental model.

Eye-Tracking in NLP. Eye-tracking data has also been leveraged to improve NLP models across a variety of tasks. Prior studies show that incorporating gaze signals can enhance performance in named entity recognition (Hollenstein and Zhang, 2019), text comprehension (Reich et al., 2022), and question answering (Wang et al., 2024). More recently, (López-Cardona et al., 2025) introduced a reward model that uses eye-tracking data to optimize Human–AI alignment. Advances in LLMs have further spurred research into using neural and behavioral signals for better alignment. For instance, (Aw et al., 2023) demonstrate how instruction-tuning can align LLMs with human brain signals. In a similar vein, our study investigates whether reading fixations can serve as meaningful input for improving alignment between LLM text completion responses and writers’ cognitive processes.

3 Methodology

3.1 Data

Thirty undergraduate college students (22 women, 8 men, age range 18–22, mean age 19.7 years) composed two texts each using the CyWrite text editor (Chukharev-Hudilainen et al., 2019)², while their eye movements were recorded with an SR Research EyeLink 1000 Plus system in a monocular remote setup, calibrated using a 9-point procedure. CyWrite maps on-screen eye fixation coordinates to

²<https://github.com/chukharev/cywrite>

corresponding in-text locations – i.e., the specific words being fixated – accounting for scrolling, line wrapping, and text edits. The writing task appeared as the top paragraph in the editor, with participants composing their responses below it. There was no time limit for the writing tasks, the order of tasks was counterbalanced across participants, and a short break was provided between the two tasks for each participant. Participants were not allowed to consult any external sources. All texts were composed in English, and all participants reported that English was their first language.

CyWrite generates a time-aligned log file that records the timestamp for every key press, key release, and eye fixation. Fixations are classified into sustained reading (operationalized as sequences of at least three consecutive eye fixations on words within the same line of text progressing from left to right) and fixating isolated words (defined as fixations on text that are not part of sustained reading sequences). For this study, we define *hesitations* as pauses between successive keypresses during which the participant engages in sustained reading.

3.2 Language Models

We generate responses for four LLMs, namely, GPT-3.5, GPT-4, LLaMa3-8B and Mistral7B. We use gpt-3.5-turbo (OpenAI, 2023) and gpt-4.1 (Achiam et al., 2023) via the OpenAI API. The exact number of parameters for the GPT models have not been officially disclosed but gpt-3.5-turbo is expected to have approximately 20 billion parameters (Singh et al., 2023). The responses are generated at a temperature setting of 0.7. We use Llama3-8B and Mistral-7B through ollama (Ollama, 2023). LLaMa3-8B and Mistral-7B have 8 billion and 7 billion parameters, respectively. For all the models, the number of tokens to be generated is dynamically defined to be approximately equal to the number of tokens in the corresponding *completion*.

3.3 Prompt Design

We first create a baseline prompt that consists of the *pretext*, an instructional prompt, and the task description provided to the student. The two task descriptions are:

- Some people have said that finding and implementing green technologies, such as wind or solar power, should be the focus of our efforts to avert climate crisis. To what extent do you

agree or disagree with this statement? Try to support your arguments with appropriate evidence from, for example, your knowledge of scientific evidence, your own experience, or your observations and reading.

- Some people have argued that animals should be given similar rights to humans. To what extent do you agree or disagree with this statement? Try to support your arguments with appropriate evidence from, for example, your knowledge of scientific evidence, your own experience, or your observations and reading.

To contrast the LLM responses with fixations against those without fixations, we generate responses for a control condition where along with the baseline prompt we provide the LLM with a matched number of non-fixated non-stop words from the *pretext* (if there are fewer non-fixated words in the *pretext* than fixated words, we include all non-fixated words). Thus, we generate LLM responses in four conditions:

1. **Baseline.** Baseline Prompt only
2. **Words.** Baseline + fixated non-stop words
3. **Sentences.** Baseline + filtered fixated sentences
4. **Control.** Baseline + a matched number of non-fixated words

The prompts for each of the conditions are presented in Table 1.

3.4 Evaluation Metrics

To evaluate the performance of LLMs with and without fixations, we establish similarity between human and LLM-generated completions in each of the four conditions on both semantic and token-based (surface linguistic form) measures.

3.4.1 Semantic Similarity

We quantify the semantic similarity between human and LLM responses by computing the cosine similarity between embedding vectors generated from the completions using the text-embedding-ada-002 model via OpenAI API (OpenAI, 2024). This approach captures global semantic alignment between the different completions.

3.4.2 Token-based Similarity

We calculate two token-based similarity metrics: F1 Score and Jaccard Index.

Condition	Prompt
Baseline	This was the task description provided to a student: <task_description >. Please write a continuation of: <pretext >.
Words	This was the task description provided to a student: <task_description >. We have identified the following key words as particularly important: <fixated words >. Please write a continuation of: <pretext >.
Sentences	This was the task description provided to a student: <task_description >. We have identified the following sentences as particularly important: <fixated sentences >. Please write a continuation of: <pretext >.
Control	This was the task description provided to a student: <task_description >. We have identified the following key words as particularly important: <non-fixated words >. Please write a continuation of: <pretext >.

Table 1: Prompt for each condition

F1 Score: F1 score accounts for both precision and recall, making it useful for evaluating word overlap between the two texts. Precision measures the proportion of shared words in the second text (W_2), while recall measures the proportion of shared words in the first text (W_1).

$$\text{Precision} = \frac{|W_1 \cap W_2|}{|W_2|}$$

$$\text{Recall} = \frac{|W_1 \cap W_2|}{|W_1|}$$

The **F1 Score**, which balances precision and recall, is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Jaccard Index: Jaccard Index is a set-based measure that quantifies the overlap between two texts by comparing the size of their intersection with their union. This metric focuses on shared words without considering their relative frequency. It is defined as:

$$\text{Jaccard Similarity} = \frac{|W_1 \cap W_2|}{|W_1 \cup W_2|}$$

4 Approach

In this section, we present our approach to data extraction and LLM response generation. The approach is outlined in Figure 1.

4.1 Extract Hesitation Events

The first step in our approach is to obtain valid hesitation events from human text production data. We defined *hesitations* as inter-keypress intervals where writing is interrupted by a pause, during

which the writer engages in sustained reading. At the time of hesitation, we extract the span of text between the start of the current paragraph and the cursor location. We call this the *pretext*. In Figure 2, | represents the cursor location at the time of hesitation. We discard all hesitations where the pretext is empty.

Once we have a valid hesitation, we traverse the log file to extract the human *completion* of the pretext. To this end, we consider all keystrokes following the hesitation until the writer types a sentence-final punctuation symbol (./?/!). The completion is valid so long as the writer does not edit or delete any portion of the pretext at any point during the completion process. Valid human completions serve as the gold standard for comparison against the LLM-generated completions. However, we discard all invalid completions. We define a *hesitation event* as a valid hesitation followed by a valid completion. The process of extracting hesitation events is outlined in Algorithm 1.

Algorithm 1 Extract Hesitation Events

```

1: function GETHESITATIONS(data)
2:   for i in data do
3:     if len(data[i].pretext) > 0 then
4:       if sustained reading in data[i] then
5:         for j from i+1 to len(data) do
6:           if data[j] starts data[i] then
7:             if data ends in {.,?;!} then
8:               Extract  $e_n$ 

```

4.2 Extract Fixations

Once we have a set of valid hesitation events e_n (pretext, completion), we extract, from the available eye-tracking data, all eye fixations on the text that occurred during each hesitation event. We include both sustained reading fixations, and fixations on isolated words. We apply the following

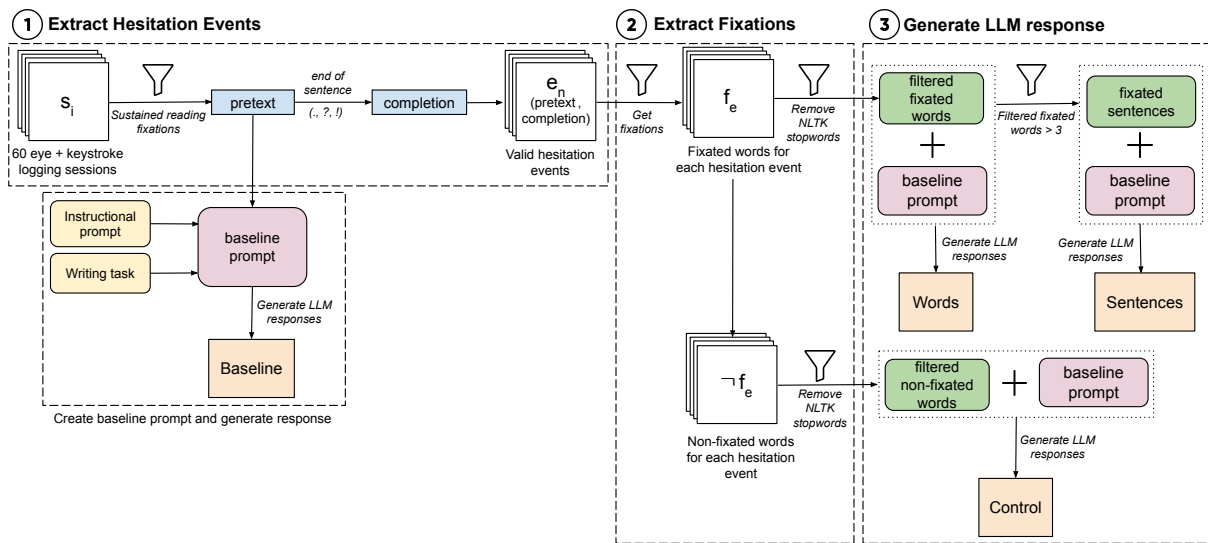


Figure 1: Overview of EyeLLM Approach

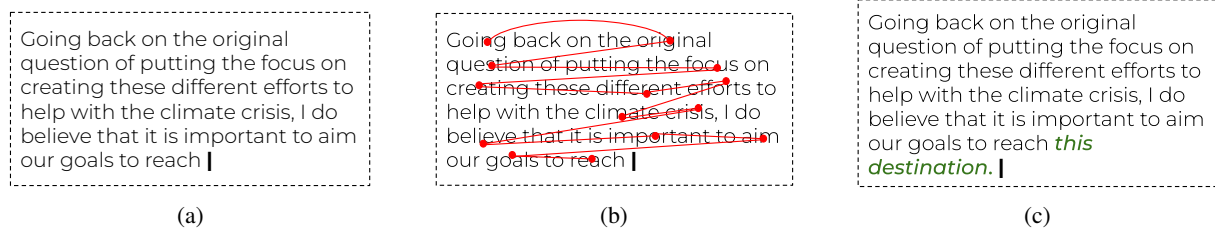


Figure 2: Example showing the extraction of pretext, fixations, and completion. (a) The writer pauses (hesitates) during text production. | indicates their cursor location at the point of hesitation. Everything between the start of the current paragraph and | is the *pretext*. (b) The writer fixates on the highlighted points as shown in the scanpath. Words containing eye fixations are *fixated words*. (c) The writer continues to produce text (highlighted in green). This is the *completion*.

Model	Similarity Scores											
	Semantic				F1				Jaccard			
	Control	Baseline	Words	Sentences	Control	Baseline	Words	Sentences	Control	Baseline	Words	Sentences
GPT-3.5	.8075	.8066	.8086*	.8090*	.1483	.1484	.1511*	.1480	.0823	.0824	.0841*	.0821
GPT-4	.8078	.8056*	.8089	.8101*	.1332	.1336	.1347	.1347	.0732	.0735	.0742	.0744
LLaMa3	.7945	.7958*	.7958	.7959	.1356	.1360	.1364	.1360	.0749	.0752	.0754	.0753
Mistral7B	.7935	.7919*	.7950*	.7945	.1228	.1244	.1262*	.1236	.0673	.0682	.0693*	.0678

Table 2: Average similarity scores across all LLMs. The highest score for each LLM is highlighted in **bold**. * marks scores that are significantly different from Control, $p < 0.01$.

filtering criteria:

1. We only include lookback fixations, i.e. fixations on the text before the cursor at the time of the hesitation.
2. We exclude fixations on words from the NLTK list of stop words, to ensure that only fixations on semantically important words are included.
3. For the Sentences condition, we identify *fixated sentences* as sentences that contain valid fixations on at least three words.

The process of extracting fixations is outlined in Algorithm 2. We only consider *hesitation events* that have at least 1 fixated word and at least 1 fixated sentence. After this filtering process, we get 822 valid hesitation events. The mean number of fixated words across all valid hesitation events is 11.11 (median 8), and the mean number of fixated sentences is 1.90 (median 1).

Algorithm 2 Extracting Fixations

```
1: function GETFIXATIONS(data, hesitation_events, n)
2:   for hesitation in hesitation_events do
3:     Extract fixations for current hesitation
4:     Remove fixations on stop words
5:     Store sentences with  $\geq n$  fixated words
6:     if valid fixations found then
7:       Append to results
8:     end if
9:   end for
10:  return results
```

4.3 Response Generation

After extracting fixation data for all hesitation events, we prompt several LLMs to generate completions. The full experimental setup is already described in Section 3.

5 Results

We run each model for 10 iterations. The results for all the models averaged over all iterations are presented in Table 2. We answer the following two research questions:

- **RQ1:** Does incorporating information about a writer’s lookback fixations improve the alignment between human and LLM-generated text completions?
- **RQ2:** How do different LLMs compare across conditions with and without lookback fixations?

5.1 RQ1: Impact of Lookback Fixations on the Similarity Scores

We perform inferential hypothesis testing to assess whether prompting condition had a statistically significant effect on similarity scores. In our analysis, we treat each similarity measure as a dependent variable. We fit linear mixed effects models (LMERs) with prompting condition (Control, Baseline, Words, Sentences) as the fixed factor. As detailed above, we generate completions 10 times for each *hesitation event*. LMERs therefore include random by-event intercepts and slopes for prompting condition.

First, we perform the analysis separately for each LLM. We fit LMERs for each measure (semantic similarity, F1, Jaccard), resulting in a total of 12 series of nested LMERs. In each series, we first fit an intercept-only model (M_0), and then add the prompting condition fixed effect (M_1). We compare model fit using the likelihood ratio test. We adopt a conservative significance threshold $p < .01$ to guard against Type I errors. When M_1 significantly improves model fit over M_0 , we evaluate the fixed-effect coefficients in M_1 to determine which prompting conditions show significant differences from the Control.

The results are shown in Table 2 and Figure 3. As expected, the Control condition does not outperform Baseline for F1 and Jaccard scores. For semantic similarity, however, Control provides significant performance gains over Baseline for GPT-4 and Mistral7B. This suggests that providing additional input to the LLM (even if it is unrelated to the eye-tracking signal) can improve human-LLM alignment in text completion.

Crucially, introducing eye-tracking signals (via Words and Sentences conditions) yields modest but statistically significant improvements over Control in all LLMs except LLaMa3. In terms of semantic similarity, Sentences generally outperform Words, except in Mistral7B. For token-based metrics (F1 and Jaccard), Words tend to perform better than Sentences, with GPT-4 being the exception.

To assess the overall effect of prompting condition across LLMs, we examine differences of average similarity scores between Control and other conditions (Table 3). To test for significance of these differences, we fit one LMER per similarity metric using data from all LLMs, treating LLM as a fixed effect, and including its interaction with prompting condition. Due to LMER convergence

issues, we simplify the random effects structure by removing the random by-event slopes. We then perform Tukey-adjusted pairwise comparisons of estimated marginal means across prompting conditions. We find that, for semantic similarity, all pairwise differences across conditions are statistically significant ($p < .0001$) except between Sentences and Words ($p = .426$). For F1 and Jaccard, Words significantly outperform all other conditions ($p < .01$), while differences among the remaining conditions are not significant ($p > .15$).

These findings support our hypothesis that including fixation-based information in prompts improves human–LLM alignment. Although LLM responses vary in sensitivity to the eye-tracking signal, overall we find that providing LLMs with fixated sentences enhances semantic alignment, while providing fixated words enhances both semantic and token-level alignment with human text completions.

Metric	Change relative to Control			
	Control	Baseline	Words	Sentences
Semantic	.8008	-.0008*	+.0013*	+.0016*
F1 Score	.1350	+.0006	+.0021*	+.0006
Jaccard	.0744	+.0004	+.0013*	+.0005

Table 3: Performance changes across prompting conditions, relative to the Control. Bold values indicate improvements. * indicates significant change ($p < .01$). Averages computed across all models.

5.2 RQ2: Differences among LLMs in Performance Across Prompting Conditions

To assess whether differences between LLMs are statistically significant, we extend the inferential tests from Section 5.1 by fitting a series of nested linear mixed-effects models (LMERs) for each similarity measure, using data from all four LLMs. We begin with a baseline intercept-only model (M_0), then sequentially add the fixed effect for prompting condition (M_1), the fixed effect for LLM (M_2), and finally the interaction between condition and LLM (M_3). Due to convergence issues, we remove by-event slopes from the random effects structure.

Model comparisons are conducted using likelihood ratio tests. For semantic similarity, successive models significantly improve the fit ($M_3 > M_2 > M_1 > M_0$, all $p < .0001$). The significant interaction term in M_3 for semantic similarity

indicates that the effect of prompting condition varies by LLM—that is, not only do the LLMs differ overall, but the way they respond to different prompting conditions also differs significantly, but only with respect to the semantic metric. On the other hand, for Jaccard and F1, M_3 does not provide further improvement of model fit over M_2 ($p = .018$; $p = .044$, respectively). This suggests no evidence for the condition-LLM interaction for the token-based metrics.

Pairwise comparisons between LLMs reveal significant differences throughout (all $p < .0001$ with Tukey adjustment), with the exception of the difference between GPT-3.5 and GPT-4 for semantic similarity ($p = .157$).

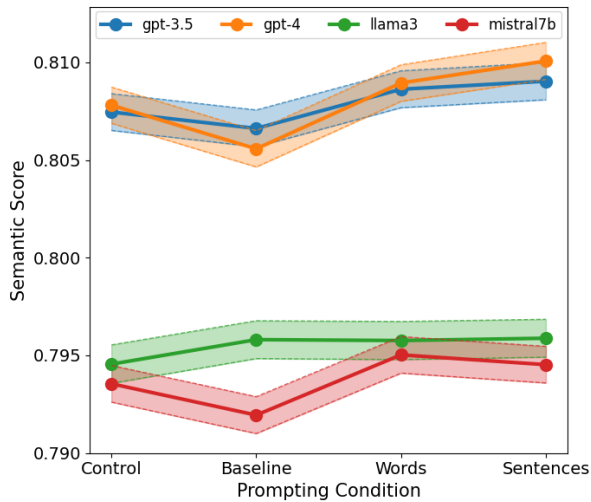
We present an overview of descriptive statistics for different similarity measures below.

Semantic Similarity: As shown in Figure 3a, GPT models consistently outperform LLaMa and Mistral in the semantic alignment across all conditions. With eye-tracking cues, all models show small relative improvements compared to the Control condition (between +.13% and +.29%), but only some of these improvements are statistically significant.

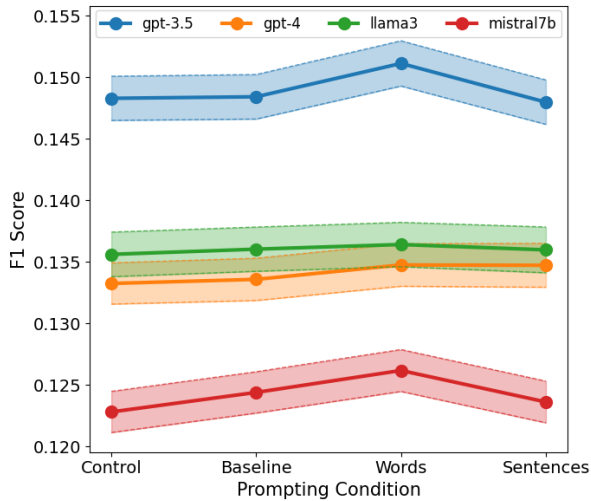
F1 Score: Figure 3b presents model comparisons based on the F1 score. Interestingly, GPT-3.5 outperforms all other models across all prompting conditions, showing the strongest token-level alignment with human completions. GPT-4 and LLaMa3 are closely comparable, while Mistral7B consistently underperforms relative to other models. Across prompting conditions, F1 score changes show greater variability. From Control to Words, only Mistral7B and GPT-3.5 show significant improvement (by +2.77% and +1.89%, respectively). GPT-4 and LLaMa3 show smaller improvements that do not reach significance threshold. All changes from Control to Sentences (ranging from +1.13% to -0.98%) are not statistically significant.

Jaccard Index: As shown in Figure 3c, Jaccard scores generally follow trends seen in F1 scores. From Control to Words, all models improve (between +0.67% and +2.97%), but only GPT-3.5 and Mistral7B show significant improvements. The shift to Sentences shows mixed changes (between +1.64% and -0.74%), none of which reach significance.

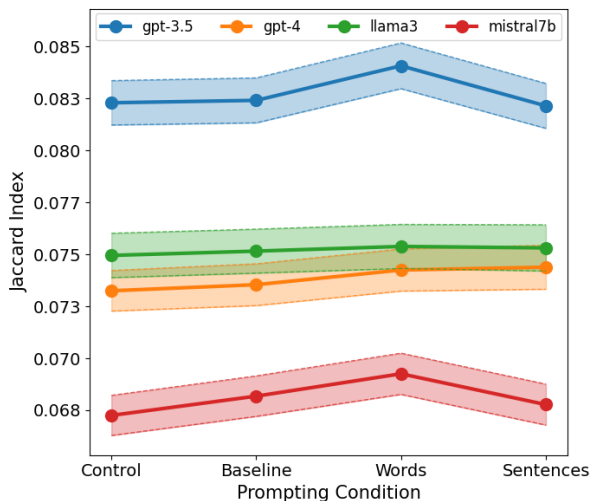
Overall, both GPT models show greatest semantic alignment with student text completions, while



(a) Semantic Score



(b) F1 Score



(c) Jaccard Index

Figure 3: Scores of different models across prompting conditions with 95% confidence intervals.

GPT-3.5 clearly leads on token-based similarity metrics. Eye-tracking cues are not sufficient to significantly change the relative performance of any two LLMs on any of the measures investigated.

6 Summary and Conclusion

To our knowledge, this paper is the first to investigate the impact of word- and sentence-level lookback fixation signal captured during writing pauses on LLM-generated text completions.

We first asked whether the eye-tracking cues improve the human-LLM alignment in the text completion task. By comparing different prompting conditions, we demonstrated that the addition of both the words and the sentences that a writer fixates resulted in small, but statistically significant improvement in the semantic alignment between LLM-generated text completions and what the writer themselves actually wrote. Adding fixated words (but not sentences) improves performance on token-based similarity metrics. This provides tentative (but, to date, best-available) evidence of the role of lookback in text planning and, again tentatively, suggests value in incorporating lookback data in intelligent, real-time tools for supporting and training written composition skills.

We then asked how different LLMs compare across prompting conditions. We found that GPT models outperform smaller open-source models on semantic metrics, while GPT-3.5 offers substantial advantages in token-based similarity. For semantic (but not token-based) metrics, significant statistical interaction between LLM and prompting condition suggests that different LLMs react differently to the eye-tracking signal.

Relative performance gains, while statistically significant, were small (in single-digit percent) across LLMs and similarity metrics. It remains to be seen whether improvements on this scale have practical value for developing educational technologies that support written composition. At the very least, they highlight the need for further research into how lookback information can be used to refine prompts.

Limitations

One limitation of our study lies in the scope of the data collected, which includes responses from 30 students. Nonetheless, we extracted 822 valid hesitation events across 60 composition sessions, reinforcing the robustness of our findings. Secondly,

while the variation in scores across conditions is statistically significant, it is relatively small and its practical significance will depend on the use case. Lastly, we do not present a complete end-to-end tool for providing LLM-generated writing assistance. However, this work establishes a strong foundation for future development in that direction.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 2016868 and 2302644. We gratefully acknowledge the insightful comments and constructive feedback from the reviewers, which significantly contributed to improving the quality of this paper. We are grateful to Dr. Emily Dux Speltz, Zoë DeKruif, and Jamie Smith for their assistance with data collection from human participants.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Khai Loong Aw, Syrielle Montariol, Badr AlKhamissi, Martin Schrimpf, and Antoine Bosselut. 2023. Instruction-tuning aligns llms to the human brain. *arXiv preprint arXiv:2312.00575*.
- Evgeny Chukharev-Hudilainen, Aysel Saricaoglu, Mark Torrance, and Hui-Hsien Feng. 2019. Combined Deployable Keystroke Logging and Eyetracking for Investigating L2 Writing Fluency. *Studies in Second Language Acquisition*, 41(3):583–604.
- Marita Franzke, Eileen Kintsch, Donna Caccamise, Nina Johnson, and Scott Dooley. 2005. Summary street@: Computer support for comprehension and writing. *Journal of Educational Computing Research*, 33(1):53–80.
- Katy Gero, Alex Calderwood, Charlotte Li, and Lydia Chilton. 2022. A design space for writing support tools using a cognitive process model of writing. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 11–24, Dublin, Ireland. Association for Computational Linguistics.
- Nora Hollenstein and Ce Zhang. 2019. Entity recognition at first sight: Improving NER with eye movement information. 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 1–10, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ángela López-Cardona, Carlos Segura, Alexandros Karatzoglou, Sergi Abadal, and Ioannis Arapakis. 2025. Seeing eye to ai: Human alignment via gaze-based response rewards for large language models. In *International Conference on Learning Representations (ICLR)*.
- Ollama. 2023. Ollama: Run llms locally. Accessed 2025.
- OpenAI. 2023. gpt-3.5-turbo-0613 announcement. Function calling, 16k context window, and lower prices.
- OpenAI. 2024. Openai api. Used to generate text embeddings via the OpenAI API.
- David Robert Reich, Paul Prasse, Chiara Tschirner, Patrick Haller, Frank Goldhammer, and Lena A. Jäger. 2022. Inferring native and non-native human reading comprehension and subjective text difficulty from scanpaths in reading. In *2022 Symposium on Eye Tracking Research and Applications*, New York, NY, USA. Association for Computing Machinery.
- Mohi Reza, Jeb Thomas-Mitchell, Peter Dushniku, Nathan Laundry, Joseph Jay Williams, and Anastasia Kuzminykh. 2025. Co-writing with ai, on human terms: Aligning research with user demands across the writing process. *Preprint*, arXiv:2504.12488.
- Rod D. Roscoe and Danielle S. McNamara. 2013. Writing pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105(4):1010–1025.
- Mukul Singh, José Cambronero, Sumit Gulwani, Vu Le, Carina Negreanu, and Gust Verbruggen. 2023. Code-fusion: A pre-trained diffusion model for code generation. *Preprint*, arXiv:2310.17680.
- Mark Torrance. 2016. Understanding planning in text production. *Handbook of writing research*, 2:72–87.
- Mark Torrance, Roger Johansson, Victoria Johansson, and Åsa Wengelin. 2016. Reading during the composition of multi-sentence texts: an eye-movement study. *Psychological Research*, 80(5):729–743.
- Bingbing Wang, Bin Liang, Lanjun Zhou, and Ruifeng Xu. 2024. Gaze-infused bert: Do human gaze signals help pre-trained language models? *Neural Computing and Applications*, 36(20):12461–12482.
- Shuo Zhang, Liangming Pan, Junzhou Zhao, and William Yang Wang. 2024. The knowledge alignment problem: Bridging human and external knowledge for large language models. *Preprint*, arXiv:2305.13669.

Span Labeling with Large Language Models: Shell vs. Meat

Phoebe Mulcaire

Duolingo

phoebe@duolingo.com

Nitin Madnani

Duolingo

nitin@duolingo.com

Abstract

We present a method for labeling spans of text with large language models (LLMs) and apply it to the task of identifying *shell language*, language which plays a structural or connective role without constituting the main content of a text. We compare several recent LLMs by evaluating their "annotations" against a small human-curated test set, and train a smaller supervised model on thousands of LLM-annotated examples. The described method enables workflows that can learn complex or nuanced linguistic phenomena without tedious, large-scale hand-annotations of training data or specialized feature engineering.

1 Introduction

Madnani et al. (2012) show that writers or speakers engaging in argumentative discourse do not simply enumerate their claims and evidence, but rather structure them in some manner for their argument to be convincing. Such discourse, therefore, might contain not only language expressing the core claims and evidence (the “meat” of the argument), but also language used to organize or support them (the “shell”). The authors also propose approaches to automatically detect shell language in real-world examples of argumentative discourse, such as test-taker responses and political debates.

To illustrate the difference between “meat” and “shell”, we provide a hypothetical test-taker response below discussing whether people learn better by being told what to do or shown what to do. Spans representing shell language are shown in bold while the core content of the argument is shown as plain text.

This is a very interesting topic for a debate. I would advocate the argument that being shown what to do is the better option because people are visual

learners. They learn better by watching than by just being listening to what someone else tells them. **While this may not apply to everyone, I think that** it certainly applies to the average joe. **For this reason, it is therefore clear that** being shown what to do is better.

In this paper, we build on the work of Madnani et al. (2012) by focusing more deeply on how shell language is used in responses written by test-takers for the Duolingo English Test (DET), a high-stakes English language proficiency test. Our goal is to build a finer-grained, accurate, and scalable shell detection pipeline for this use case, leveraging modern transformer-based approaches to power each stage.

We first detail our motivations for applying shell detection to test-taker responses (§2). Next, we discuss our annotation rubric for identifying shell language (§3) and our small-scale use of human annotation to validate and refine this rubric. We experiment with automatic annotation of test-taker responses using both non-reasoning & reasoning foundation models (§4), resulting in strong machine-human agreement rates. Then, we attempt to distill our annotations into a BERT model that would be cheaper & faster to deploy for operational use (§5), and provide additional discussion of our approach and error analysis (§6). Finally, we conclude with a comparison to related work (§7) and possible directions for future work (§8).

2 Motivation

English language assessments typically contain prompts asking test-takers to write open-ended responses as a demonstration of their writing proficiency. These prompts generally require arguing for/against a position with appropriate supporting

evidence, or relating a past event matching a high level-description (e.g. “talk about a time when...”). Given the nature of the writing tasks, these responses are likely to contain some amount of shell language, as illustrated by the sample response in the previous section.

A certain amount of shell is useful – necessary, in fact – to scaffold one’s arguments and produce a comprehensible and convincing argument. However, we have observed that many test-takers overuse such language to artificially inflate response length and vocabulary sophistication – both of which can impact the accuracy of automated essay scoring systems.

In this paper, we want to reliably identify (and categorize) spans of shell language in test-taker responses, independently of whether it is used appropriately to connect and organize the text or misused to pad it out with formulaic phrases. Some possible applications of reliable shell detection would include:

- Detecting the use of memorized response templates and other bad-faith patterns that rely on shell language overuse,
- Developing an independent measure of content development (the “meat”), and
- Gaining insights into stylistic variance that may arise even in the absence of shell overuse

Achieving these goals would allow us not only to improve the robustness of automated assessment to a common strategy employed by test-takers to fool automated scoring systems but also to improve measurement of content and coherence.

3 Annotation Rubric

The starting point of our pipeline, and the foundation of our approach, is an annotation rubric which defines & categorizes shell text. We use this rubric for manual annotation as well as to bootstrap automatic annotations using large language models. Since [Madnani et al. \(2012\)](#) do not share any annotation guidelines, we construct our own rubric for identifying shell language in test-taker responses.

We relied on multiple rounds of human annotation to start with an initial draft of our shell annotation rubric¹ and refine it into its final form.

¹To create the initial draft rubric, we employed few-shot prompting, supplying ChatGPT with a general description of shell language from ([Madnani et al., 2012](#)) along with 100 actual test-taker responses containing a range of shell language spans.

Specifically, the authors first collaboratively annotated 11 test-taker responses using the draft rubric and made major revisions based on the ensuing discussions. Next, the authors independently annotated 50 additional responses based on the revised rubric to determine any remaining discrepancies which were then resolved in a curation session. No changes were made to the rubric after this point. Annotation, review and curation were performed using INCEPTION ([Klie et al., 2018](#)).

Given that our goal is to enable finer-grained analyses of shell language, our final rubric defines multiple shell categories, as described in the subsections below.

3.1 Category A: Discourse Markers/Linking Expressions

The shell language spans in this category are defined to be words and phrases that are either serving an organizational or discursive purpose. For example, ones that link sentences or paragraphs with the goal of progressing between ideas. However, single-word coordinating conjunctions like “because”, “but”, “and”, etc. within sentences are not annotated as shell language. Examples of category A spans observed in test-taker responses include but are not limited to:

- To begin with ...
- In conclusion ...
- Firstly ...
- Secondly ...
- In addition ...
- There are three examples of ...
- For example, ...
- That is because ...
- Expanding on the previous discussion ...
- This is another reason
- ... in addition to the previous discussion

As the examples show, this category is mainly defined by short phrases and expressions, not entire sentences. A complete sentence of shell-like material is more likely to be category B, which we describe next.

3.2 Category B: General/Vague Statements

This category consists of phrases or statements that are formal and/or impersonal in nature and add emphasis, reflection, or consideration of the prompt or topic under consideration but *without* any real

content. Spans of this extremely productive category are often employed as padding in bad-faith responses. A very small subset of observed examples is shown below.

- It is imperative to recognize that ...
- ... would be very significant for us
- In today's age ...
- Today in society, there is a heated on-going discussion on the topic of ...
- If you ask me I would say that the statement has both pros and cons.
- In this burgeoning epoch of science and technology, we are dwelling in the 21st century.
- There is a widespread worry that this will lead to a myriad of concern in the world.

3.3 Category C: Prompt/Topic Restatement

This category contains sentences or chunks that simply restate the prompt or initial argument without any further development. We have observed that when a large part of the prompt is restated, the surrounding phrases are often from categories A or B. A few real-world examples are shown below with the corresponding prompt in parentheses. Note that only the category C spans are shown in bold; spans of any other categories are not shown.

- Today in the society, there is a heated on-going on discussion on the topic that **due to the invention of cell phones, people can communicate via text messages.**
(*Due to the invention of cell phones, people can communicate via text messages. Describe the ways texting has changed how we communicate.*)
- One of the most important trends in today's world is the sudden upsurge in the statement that **Acquiring new knowledge and skills doesn't always happen quickly.**
(*Acquiring new knowledge and skills doesn't always happen quickly. Do you think that patience is key when it comes to learning, or do you think it is possible to learn things quickly if you are motivated? Support your opinion with your personal experience and observation.*)

It must also be noted that not *all* mentions of the prompt should automatically be marked as shell. Specifically, we do not mark such a mention as shell if the response:

1. sufficiently restructures or paraphrases it (especially to use it as a topic claim) instead of just quoting or restating it, or
2. simply refers to entities or noun phrases from the prompt in context.

As an example consider the span ... being focused on a single thing is more likely to lead to higher productivity in a response to the prompt *Are you more productive when you are doing a few things at the same time, or are you more productive when you have only a single thing to focus on? What do you think helps you to be more productive?* We would not mark this as a category C shell span because the prompt topic has been paraphrased sufficiently to serve as a topic claim/thesis statement. This distinction is somewhat subjective, and while we achieved good inter-annotator agreement on this category, this point was likely a source of ambiguity for models.

3.4 Category D: Appeal to Authority

This category includes mentions of reports or studies that imply external validation or evidence. Examples include:

- A report from University of Maryland shows that ...
- Oxford University conducted a study that confirmed ...
- For example, a report published by The New York Times reveals that ...

3.5 Category E: Stance-taking

This category contains phrases or statements used to convey the writer's stance or position, whether in the first-person or third. Observed examples include:

- I feel/believe/think (that) ...
- From my point of view ...
- In my opinion ...
- Yes, I agree with the statement that ...

The exception for this category are phrases that are used to convey the writer's personal preference and do not serve a stance-taking role. For example,

consider the sentence I like good environment for touring because i loved with nature. Here, the phrase I like is used to convey the writer’s personal preference for a specific type of environment rather than their argumentative stance.

3.6 Rubric usage

Although our final rubric delineates five different categories of shell text, many shell language spans usually serve multiple purposes (e.g., the phrase More and more people believe that ... can be said to convey both general emphasis (category B) and the writer’s position (category E). In such cases, our practice is to consistently choose the category that seems more relevant in the context of the full response. For purposes of evaluation and error analysis, we also sometimes refer to a sixth category “O” consisting of all spans *not* labeled as shell (the “meat” of the response). We do not separately label this category manually or ask models to directly annotate O spans; it’s defined by the absence of annotations for other classes.

Once the final rubric was created, the authors independently annotated 92 additional responses followed by curation, for a final dataset of 142 responses with individual and curated shell annotations. Note that the first 11 annotated responses (used to make major revisions to the initial rubric) are *not* part of this final set, as their annotations do not reflect the final rubric. We then split this dataset into a "training set" of 40 responses, from which few-shot examples are drawn (see §4), and a test set of 102 responses.

4 Scaling Annotation with LLMs

Shell annotation is a complex task and traditional supervised approaches would require a much larger number of annotated examples to train, but human span annotation is time-intensive and tedious. In this section we evaluate the accuracy of annotations elicited with few-shot learning.

4.1 Method

For LLM-based annotation, we compare five models: DeepSeek-V3 (Chandra et al., 1981) GPT-4o (Hurst et al., 2024), DeepSeek-R1 (Liu et al., 2024), o1 (Jaech et al., 2024), and o3-mini (OpenAI, 2025). Note that the first two are non-reasoning models, while the latter three use self-prompting or reasoning techniques recently popularized by o1 and DeepSeek-R1.

```
<shell category="B"> In this modern world </shell>, artificial intelligence <shell category="B"> is so well known in the world </shell>, which is a kind of intelligence. <shell category="A"> Futhermore </shell>, <shell category="E"> I firmly agree with this given notion that </shell> intelligence has distinct types.
```

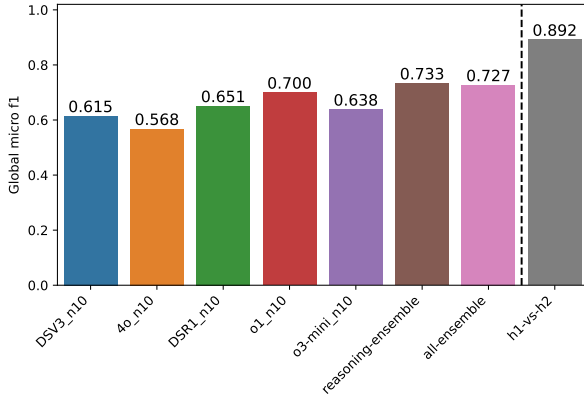
Figure 1: Markup format for LLM annotations.

Model	Success rate	Format errs	Generation errs
DeepSeek-R1	0.75	3	22
DeepSeek-V3	0.95	4	1
gpt-4o	0.93	7	0
o1	0.98	0	2
o3-mini	0.99	0	1

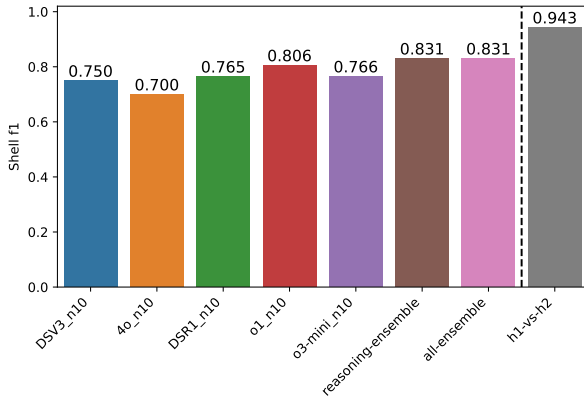
Table 1: Success rate of generating a valid annotation on the first try, by model (using 5 examples). Error counts are out of 102 responses. Note that DeepSeek-R1 had high rates of prematurely truncated responses seemingly unrelated to the task.

For each model, we use the same prompt containing the entire rubric, along with either 5 or 10 example responses annotated in an XML-like format with the shell category as an attribute (see Figure 1). We use this prompt to elicit span annotations on our eval set of 102 instances; the model is provided with the writing prompt and unannotated test-taker response, and responds with the annotated text in the same format as the examples. We chose this format based on its expressivity and convenience and did not experiment with any additional formats for now (see §7 for more discussion).

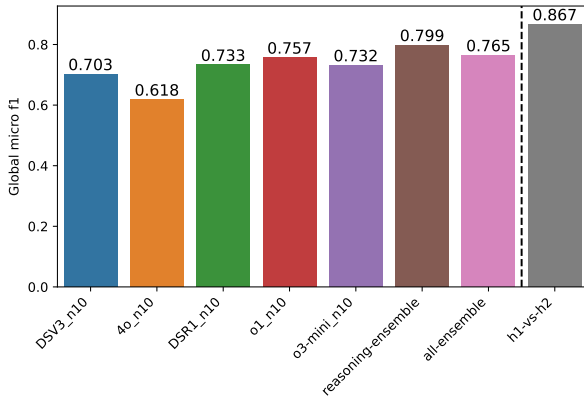
We validate the annotations by trying to automatically parse the XML, checking that there are no nested tags or task-unrelated tags, and that the text is unaltered from the original. The rate of validation failure varies by model. Furthermore, this failure rate is not necessarily constant for a given model; responses where annotation failed on the first round were more likely to also fail on a second try, suggesting that some examples are inherently hard to produce valid annotations for. The most common causes of error were incorrectly formatted XML (unclosed tags, nested tags or non-task-related tags) and missing sentences or phrases. Notably, we found almost no cases where the generated annotated text included unwanted “corrections” of grammatical or typographical er-



(a) Multiclass shell labeling.



(b) Binary shell labeling.



(c) Multiclass shell labeling (B and C categories excluded).

Figure 2: Token-level F_1 for three shell annotation tasks. Reasoning models outperform non-reasoning models, and the ensembles improve slightly over the best individual models. Exclusion of B and C categories improves micro-averaged F_1 for all models.

rors in the original test-taker response, with the exception of whitespace errors such as replacing multiple spaces with a single space or inserting a missing space after sentence-final punctuation. For purposes of evaluation, we automatically resolved these whitespace errors by editing all annotated versions of a text to match the original (to ensure

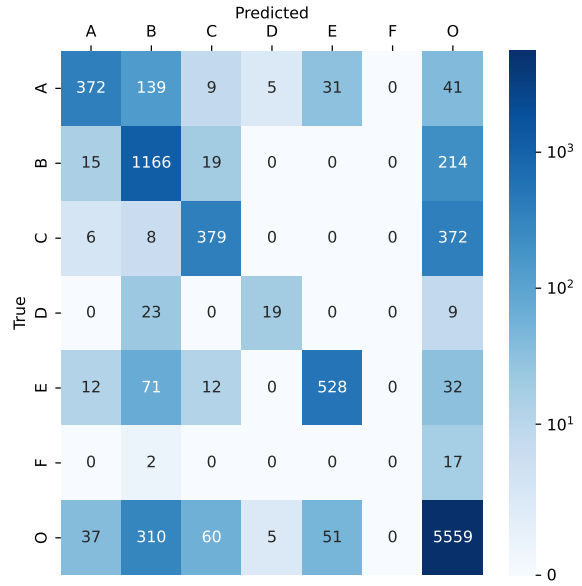


Figure 3: Confusion matrix for all-ensemble.

tokenization was compatible) before comparing annotations. For our provider of DeepSeek-R1, a significant fraction of long responses were cut off prematurely, increasing the error rate beyond what was attributable to formatting errors. Table 1 compares the LLMs by the fraction of responses that passed validation with a single request.

4.2 Results

Next, we compute token-level metrics for the LLM annotations using the curated human labels as the gold standard, and compare to the inter-annotator agreement for the human annotators. Figure 2 shows results for binary (shell vs non-shell) and multiclass evaluation.

We present the confusion matrix for the all-model ensemble in Figure 3 as a relatively representative example of the errors made by all models. The counts represent individual token counts. The confusion matrix provides insight into specific pairs of categories often confused; for example, we see that categories D and E have few false positives, i.e. they are rarely predicted when the true label is another category. We also observe that B and C are the categories with the most errors (both because they have high true token counts and because they require the most subjective decisions). For this reason, we also include F_1 results considering only a subset of shell category labels (all except B and C) in Figure 2.

Model	Prompt tokens	Completion Tokens	Total cost
DeepSeek-R1	431,316	127,414	\$2.19
DeepSeek-V3	431,737	15,566	\$0.56
GPT-4o	430,290	15,859	\$1.06
o1	425,061	257,592	\$20.28
o3-mini	431,234	492,148	\$2.54

Table 2: Comparing prompt tokens, completion tokens, and total cost when annotating our curated evaluation set of 102 responses using various LLMs.

4.3 Costs

Our method of shell annotation with large language models requires a lengthy rubric and several example texts to be provided in the prompt for every instance. This is a relatively costly approach. In addition, our use of reasoning models leads to high completion token counts.

In order to provide a useful comparison of model costs, Table 2 shows prompt/completion token counts and costs when annotating our curated evaluation set of 102 responses using the same set of LLMs we used in §4. Note that we do not retry any incorrectly formatted annotations for this specific set, so the error rates reported in Table 1 should also be considered when comparing these costs.

In addition, one would expect to incur significant upfront costs iterating and validating the annotation scheme and rubrics. In our case, we spent a total of \$3,665.45 across all experiments.

5 Supervised Learning to Detect Shell

In this section, we attempt to distill a large number of LLM-based annotations into a BERT variant, ModernBERT (Warner et al., 2024). There are several advantages to this approach: BERT models are cheaper, can be run locally (avoiding dependence on external APIs) and can directly produce per-token labels rather than generating the annotated text, removing a potential source of errors.

Based on the results in §4, we choose OpenAI’s o1 model as the best single model for the task. Using the same approach as previously described, we prompt o1 to annotate 7100 additional test-taker responses, and split the resulting dataset into a training set (6500 responses) and a validation set (600 responses). We then convert the annotations into BIO format (Ramshaw and Marcus, 1999) and finetune ModernBERT on three training samples with different sizes: 500, 1000, and the full 6500. For all finetuning runs, we set the batch size to 12 and learning rate to $7e^{-05}$ and train for 10 epochs with early stopping based on the performance on the

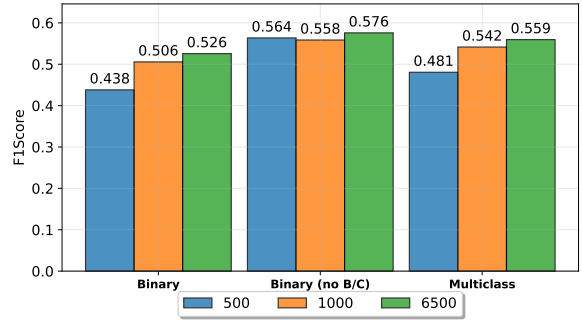


Figure 4: ModernBERT F_1 for each task on the human-labeled test set (102 examples). Notably, multiclass labeling performance is actually higher than binary labeling performance on equal data sizes.

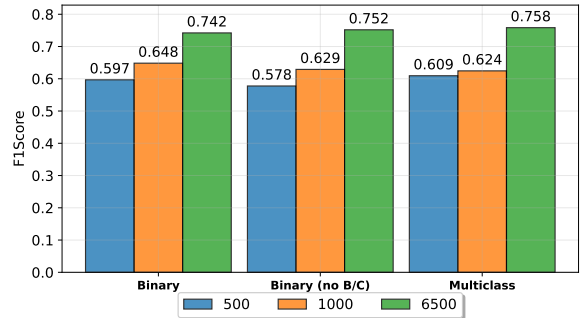


Figure 5: ModernBERT F_1 for each task on the o1-labeled dev set (600 examples). Performance is significantly higher than on the human-labeled test set, suggesting that the BERT model has learned o1-generated patterns that are misaligned to human raters.

validation set.²

We finetune and evaluate ModernBERT on three tasks: binary shell labeling, binary shell labeling with B and C categories excluded, and multiclass labeling (see Figure 4). ModernBERT substantially under-performs the LLM used to train it (o1 with 5 examples) on the human-labeled test set, never surpassing 0.6 F_1 even for the easiest task. However, on the o1-labeled validation set, ModernBERT trained on 6500 examples surpasses 0.75 F_1 for multiclass labeling (Figure 5).

²Learning rate was chosen based on a search over the validation set when training on 500 responses.

6 Discussion

6.1 o1 error analysis

Shell labeling is a difficult and to some extent subjective task. In this section, we present a qualitative analysis of the differences between the best-performing single LLM (o1) and curated human annotations, along with examples. To improve readability, we use `<X>...</X>` as a shorthand for `<shell category="X">...</shell>`.

The most common error categories in the confusion matrix in Figure 3 are missing tokens of categories B (general statements) and C (topic restatement), and various non-B tokens labeled as B. We observe a similar pattern when looking at whole-span errors³. The most common cases of whole-span errors are B-spans applied to O and A text. O spans applied to C text (i.e. missed C labels) are also common. This is perhaps to be expected given the rubric; B and C are the most contextual and nuanced categories, requiring consideration of what is specific to the prompt vs generic and what is restatement vs original.

In the following example, the model identified most of the sentence as B, possibly due to the positive emphasis ("outstanding", "wide knowledge") which is often seen in B spans.

Curated O vs. LLM B

```
<A>First of all</A> <E>it is true that</E> college and university can serve as an outstanding place to gaing wide knowledge and contact as you can meet with like minded individuals <A>First of all</A> <E>it is true that</E> <B>college and university can serve as an outstanding place to gaing wide knowledge and contact as you can meet with like minded individuals</B>
```

In another case, a partial reference to the prompt ("the second part of the statement") was mistakenly treated as a restatement of the prompt, as shown below. This may be a case where o1 talked itself into an otherwise unlikely error.

³Whole-span errors occur when a predicted span has no overlap with a human-annotated span of the same category. Boundary errors, by contrast, involve partial overlap but incorrect span length.

Curated O vs. LLM C

```
<E>I storngly prefer</E> the second part of the statement <A>for many reasons</A>. <E>I storngly prefer</E> <C>the second part of the statement</C> <A>for many reasons</A>.
```

Below we considered this declaration of "heated debate" to be an instance of B, but o1 did not:

Curated B vs. LLM O

```
<C>intercultural communication can be a valuable learning experience</C> <B>has sparked a heated debate.</B> <C>intercultural communication can be a valuable learning experience</C> has sparked a heated debate.
```

A common boundary error involved commas. During manual annotation, we settled on a convention of excluding trailing commas from shell spans but did not explicitly specify this in the rubric. O1 frequently took the opposite approach as shown below, causing a 1-token error.

Curated vs. LLM

```
<A>To sum up</A>, ... <A>To sum up,</A> ... <A>For my experience</A>, ... <A>For my experience,</A> ...
```

Finally, we excluded mentions of topics or entities from the prompt from annotations in sentences that were otherwise B. This was attested in a few examples, but not made explicit in the rubric, and o1 tended to include them, leading to boundary errors, as the example shows.

Curated vs. LLM

```
<B>A serious amount of worldwide attention has been drawn to</B> the intercultural communication. <B>Beacuse of the existence of evidencen in favour of as well as against the approval of</B> intercultural communication. <B>A serious amount of worldwide attention has been drawn to the intercultural communication.</B> <B>Beacuse of the existence of evidencen in favour of as well as against the approval of intercultural communication.</B>
```

6.2 Interpretation

Many error types above are consistent and systematic, which is a promising sign for improving the accuracy of automatic shell annotation. In a few cases, o1’s annotations were arguably more consistent with the intent of the rubric than the curated human annotations. For example, some spans of A and E were missed by human annotators, categories which were fairly reliably marked by o1. In other cases, o1 marked statements that broadly paraphrased statements from the prompt as C when human annotators judged the paraphrase as original in form, though not content.

As expected, the two step training procedure in §5 results in a model that suffers from two sources of errors: errors between o1 and human annotators, and errors between ModernBERT and o1. In fact, it seems that ModernBERT does not learn to correct any significant portion of o1’s errors, as the total error rate is not much better than if the two sources of error were entirely independent: we observe $0.559F_1$ for the largest ModernBERT model for multiclass labeling, vs. 0.531 expected ($0.7 \text{ o1 } F_1 \times 0.752 \text{ ModernBERT } F_1 \text{ on o1’s labels}$). This is consistent with LLMs consistently diverging from human annotations; ModernBERT is learning to imitate systematic error, rather than guessing in response to random noise.

7 Related work

The work most closely related to ours and the one we build upon is that of [Madnani et al. \(2012\)](#). However, there are also salient differences between our work and theirs. They rely on a small set of human annotations to train a binary, feature-based, discriminative classifier for shell language whereas we use a small, curated set of human annotations to bootstrap LLM-generated annotations at scale, and then distill them into an end-to-end transformer model used for finer-grained, multi-class, shell span classification. Additionally, while they do not share any information about their annotation process, we share a detailed rubric along with examples for each shell category. [Bejar et al. \(2013\)](#) apply the shell model developed by [Madnani et al. \(2012\)](#) to GRE essays to evaluate whether it agrees with expert raters’ judgments and whether the presence of shell language has an effect on the essay scores. [Du et al. \(2014\)](#) devise an unsupervised

HMM-LDA topic model for shell language and apply it to posts from online debate forums. Similarly, [Ó Séaghdha and Teufel \(2014\)](#) use a topic model to capture words & constructs used to express rhetorical function in scientific papers.

LLMs have been extensively used for a wide range of linguistic analysis tasks. Some of these tasks are fairly straightforward. For example, [Hao et al. \(2024\)](#) use ChatGPT to annotate conversation chat turns in a collaborative problem solving setting with a pre-defined set of labels. However, the decoder-only framework for text generation makes it difficult to represent more complex linguistic structures such as spans or dependency relations and their relationships to the annotated text, and, to our knowledge, there has been no consensus on the format to use for span annotation with LLMs (regardless of the particular application). [Blevins et al. \(2022\)](#) experimented with LLMs for sequence tagging tasks, including multi-token spans for chunking and NER. They framed the task as BIO tagging at the word level, regenerating the text with labels following each word. Since our spans are frequently even longer than syntactic chunks, and rarely as short as single words, we opt for a format that abstracts away from individual word labels.

More recently, [Kasner et al. \(2025\)](#) experimented with span annotation for evaluation of generated text by using structured decoding to get a list of spans with category labels in JSON format. This has the advantage of not requiring re-generation of the full input text. However, our application does not require full category names for individual annotations (only a single-character label) and we expect to label relatively densely (such that a significant fraction of the text would have to be copied in the output anyway). Future work should directly compare these output formats on a single task (or multiple tasks) and investigate the effects of output format on overall performance and error types.

8 Conclusions and Next Steps

We have shown that LLMs can be used for scalable span annotation and that reasoning models have a distinct advantage at the task of labeling shell text. However, both the original LLM annotation process and training a smaller model to imitate an LLM’s annotations remain error-prone. Based on the consistency of certain error types (§6), we believe that refinements to the annotation rubric could significantly improve the accuracy of LLM

annotation. For example, the distinction between a clear restatement of the prompt and its paraphrase is a bit subtle and can be made clearer to ensure a more consistent interpretation. Other directions for future work include:

- a more thorough hyperparameter search to improve supervised learning,
- finetuning a reasoning LLM either directly on the curated human data or a combination of human and LLM-annotated data (given the small size of the human data), and
- experimenting with other output formats from related work such as structured decoding for greater consistency.

While our supervised shell detection results certainly leave room for improvement, we hope that the work done in this paper can still serve as a source of useful information to other researchers working on shell language detection and, more broadly, LLM-based span annotation. We believe that the workflow proposed in this paper can be applied to other types of non-overlapping span-labeling tasks, assuming a rubric with clearly defined categories and reliable human-human agreement.

9 Acknowledgements

We thank the anonymous reviewers, as well as Kevin Yancey and Alina von Davier, for their valuable feedback.

Limitations

There are several limitations of this work. Most significantly, while the two-step annotation procedure we describe yields promising results, the resulting error rate of the final ModernBERT model may limit its application without additional refinements to the rubric and/or the training procedure (including improved hyperparameter tuning). For our LLM experiments, we compared several different models in §4 and found that an ensemble of multiple models performed best. However, our budget and time constraints limited the number of compared models, and, in the end, we had to pick the best single model (o1) to produce training data for ModernBERT instead of the ensemble. Due to limited space, our error analysis only covers errors made by o1, and does not show the extent to which the same patterns may be shared by other LLMs or

ModernBERT. Finally, our results are limited to the specific choice of span annotation format that we chose. As mentioned in §7, other formats may have different tradeoffs, which we hope future work will explore.

References

- Isaac I. Bejar, Waverly VanWinkle, Nitin Madnani, William Lewis, and Michael Steier. 2013. [Length of textual response as a construct-irrelevant response strategy: The case of shell language](#). *ETS Research Report Series*, 2013(1):i–39.
- Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2022. Prompting language models for linguistic structure. *arXiv preprint arXiv:2211.07830*.
- Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. [Alternation](#). *Journal of the Association for Computing Machinery*, 28(1):114–133.
- Jianguang Du, Jing Jiang, Liu Yang, Dandan Song, and Lejian Liao. 2014. Shell miner: Mining organizational phrases in argumentative texts in social media. In *2014 IEEE International Conference on Data Mining*, pages 797–802. IEEE.
- Jiangang Hao, Wenju Cui, Patrick Kyllonen, Emily Kerzabi, Lei Liu, and Michael Flor. 2024. [Scaling up the evaluation of collaborative problem solving: Promises and challenges of coding chat data with chatgpt](#). *Preprint*, arXiv:2411.10246.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Zdeněk Kasner, Vilém Zouhar, Patrícia Schmidtová, Ivan Kartáč, Kristýna Onderková, Ondřej Plátek, Dimitra Gkatzia, Saad Mahamood, Ondřej Dušek, and Simone Balloccu. 2025. Large language models as span annotators. *arXiv preprint arXiv:2504.08697*.
- Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. [The inception platform: Machine-assisted and knowledge-oriented interactive annotation](#). In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Nitin Madnani, Michael Heilman, Joel Tetreault, and Martin Chodorow. 2012. **Identifying high-level organizational elements in argumentative discourse**. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 20–28, Montréal, Canada. Association for Computational Linguistics.
- Diarmuid Ó Séaghdha and Simone Teufel. 2014. **Unsupervised learning of rhetorical structure with un-topic models**. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2–13, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- OpenAI. 2025. Openai o3-mini technical report. Technical report.
- Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.
- 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.

Intent Matters: Enhancing AI Tutoring with Fine-Grained Pedagogical Intent Annotation

Kseniia Petukhova, Ekaterina Kochmar

Mohamed bin Zayed University of Artificial Intelligence
{kseniia.petukhova, ekaterina.kochmar}@mbzuai.ac.ae

Abstract

Large language models (LLMs) hold great promise for educational applications, particularly in intelligent tutoring systems. However, effective tutoring requires alignment with pedagogical strategies – something current LLMs lack without task-specific adaptation. In this work, we explore whether fine-grained annotation of teacher intents can improve the quality of LLM-generated tutoring responses. We focus on MathDial, a dialog dataset for math instruction, and apply an automated annotation framework to re-annotate a portion of the dataset using a detailed taxonomy of eleven pedagogical intents. We then fine-tune an LLM using these new annotations and compare its performance to models trained on the original four-category taxonomy. Both automatic and qualitative evaluations show that the fine-grained model produces more pedagogically aligned and effective responses. Our findings highlight the value of intent specificity for controlled text generation in educational settings, and we release our annotated data and code to facilitate further research: <https://github.com/Kpetyxova/autoTree/tree/main/mathdial>

1 Introduction

Human tutoring is a cornerstone of educational development, playing a vital role in empowering learners and fostering societal progress. One-on-one tutoring has long been recognized as highly effective (Bloom, 1984); however, its widespread implementation is constrained by the limited availability of qualified tutors. Recent advancements in LLMs have shown great promise in educational contexts (Wang et al., 2024; Gan et al., 2023), leading to the emergence of LLM-powered intelligent tutoring systems (ITS) (Pal Chowdhury et al., 2024; Liu et al., 2024) and the use of LLMs as tutors via advanced prompting strategies (Denny et al., 2024; Mollick and Mollick, 2024). These AI tutors serve

a range of educational objectives (Wollny et al., 2021), with one of the most prominent being the remediation of student mistakes and confusion – an area that continues to drive the development of AI tutoring systems (Macina et al., 2023; Wang et al., 2023).

While LLMs do well both at generating human-like conversations and at addressing various reasoning tasks, such as commonsense reasoning and basic mathematical reasoning (Achiam et al., 2023; Kojima et al., 2022; Laskar et al., 2023; Yang et al., 2024), they cannot be directly deployed in educational systems without significant adaptation. Effective tutoring requires more than fluent conversation – it involves guiding learners to discover answers on their own. Rather than simply providing solutions, a good tutor employs strategies such as giving hints, asking questions in a Socratic dialog (Carey and Mullan, 2004), and encouraging active problem-solving. As such, LLM-based tutors should ideally align with human tutoring strategies (Nye et al., 2014) and active learning practices shown to enhance student outcomes (Freeman et al., 2014).

In order to have such models, we need dialog tutoring datasets. MathDial (Macina et al., 2023) is one such dataset, comprising tutor-student dialogs centered around math reasoning tasks. Each teacher utterance is labeled with one of four pedagogical move types from Macina et al. (2023): *Focus* (guiding task progress), *Probing* (encouraging conceptual exploration), *Telling* (providing help when students are stuck), or *Generic* (non-pedagogical conversational turns). These annotations were provided by teachers during data collection to better scaffold student learning. While this four-category taxonomy offers a helpful high-level structure, it lacks the fine-grained detail needed for advanced applications such as controlled response generation, pedagogical analysis, and behavior modeling in AI tutors. At the same time,

finer-grained annotations may enable better interpretability, improved pedagogical alignment, and greater flexibility in guiding student learning experiences.

Although MathDial’s original taxonomy includes only four broad categories, the authors also provide an expanded set of eleven fine-grained intents, which could offer greater control and variety in AI-generated tutoring responses. Building on this, in this work, we apply a fully automated framework for conversational discourse annotation (Petukhova and Kochmar, 2025) to **re-annotate a portion of the MathDial dataset using the finer-grained eleven-intent taxonomy**. This annotation framework uses LLMs to automatically construct a decision tree from the taxonomy and use it to label utterances, providing a scalable alternative to manual annotation. This approach has demonstrated superior performance compared to crowdworkers in annotating dialog with speech functions taxonomy (Eggins and Slade, 2004).

Our goal in this work is to assess whether such more detailed annotations can improve the quality of LLM-based tutoring through fine-tuning models on both the original and re-annotated data. Specifically, we fine-tune Mistral-7B-Instruct on the original coarse-grained as well as the new fine-grained annotation, and compare the generated tutor responses using automatic metrics and human evaluation. Our results demonstrate that **the fine-grained model produces more pedagogically aligned and effective responses**. To facilitate further research and development, we release a public repository containing both the code and the re-annotated dataset.¹

2 Background & Related work

2.1 The MathDial Dataset

We build on the foundational work of Macina et al. (2023), whose dataset provide an invaluable basis for advancing pedagogically aligned dialog systems. MathDial is a large-scale, high-quality dialog tutoring dataset focused on multi-step math reasoning problems. Unlike previous datasets that suffer from low pedagogical quality, small size, or lack of grounding, MathDial provides rich annotations grounded in realistic student confusions and pedagogical strategies. The authors introduce a novel semi-synthetic data collection framework

¹Available at <https://github.com/Kpetyxova/autoTree/tree/main/mathdial>.

that pairs expert human teachers with LLMs simulating students and their errors, enabling scalable and controlled creation of educational dialogs that closely mimic authentic tutoring scenarios. This approach effectively addresses privacy concerns and quality issues associated with crowdsourcing or classroom recordings.

The authors’ methodology consists of a Wizard-of-Oz-inspired framework (Kelley, 1984), where expert teachers engage in one-on-one tutoring dialogs with LLMs acting as students. These student models are carefully prompted with student profiles and frequently occurring conceptual errors generated using temperature sampling over diverse reasoning paths produced by LLMs. The math word problems (MWP) used are sourced from GSM8K (Cobbe et al., 2021). Teachers are instructed to scaffold student understanding using a taxonomy of four pedagogical moves: *Focus*, *Probing*, *Telling*, and *Generic*, with additional fine-grained intents (see Table 1).

Crucially, before writing a response, teachers must annotate the pedagogical move being employed, encouraging more intentional strategy use. The dialogs are also grounded in metadata, including the specific confusion, full problem, step-by-step solutions, and whether the confusion was resolved, thus offering rich signals for training AI tutors.

Empirical evaluation demonstrates that models fine-tuned on MathDial significantly outperform both zero-shot and instruction-tuned larger LLMs like ChatGPT in terms of correctness and equitable tutoring (Macina et al., 2023). Notably, fine-tuned open-source models achieved similar rates of student problem-solving success while reducing the incidence of “telling” – prematurely giving away solutions. Human evaluations confirmed that these fine-tuned models were more coherent, correct, and pedagogically effective than large prompted models.

2.2 Annotation Framework

While manual discourse annotation is costly and time-consuming, advances in LLM-based annotation present a promising alternative with demonstrated improvements in speed, consistency, and cost-effectiveness (Gilardi et al., 2023; Hao et al., 2024). Petukhova and Kochmar (2025) have recently proposed an open-source pipeline for fully automated discourse annotation using LLMs. Specifically, this pipeline automates the construc-

Category	Intent	Example
Focus	Seek Strategy	<i>So what should you do next?</i>
	Guiding Student Focus	<i>Can you calculate ... ?</i>
	Recall Relevant Information	<i>Can you reread the question and tell me what is ... ?</i>
Probing	Asking for Explanation	<i>Why do you think you need to add these numbers?</i>
	Seeking Self Correction	<i>Are you sure you need to add here?</i>
	Perturbing the Question	<i>How would things change if they had ... items instead?</i>
	Seeking World Knowledge	<i>How do you calculate the perimeter of a square?</i>
Telling	Revealing Strategy	<i>You need to add ... to ... to get your answer.</i>
	Revealing Answer	<i>No, he had ... items.</i>
Generic	Greeting/Farewell	<i>Hi ... , how are you doing with the word problem? Good Job! Is there anything else I can help with?</i>
	General Inquiry	<i>Can you go walk me through your solution?</i>

Table 1: Teacher moves with examples of utterances and their intents from MathDial (Macina et al., 2023).

tion of hierarchical tree annotation schemes and the annotation of utterances within dialogs, making it a promising and scalable approach for enriching the MathDial dataset with more detailed teacher intent annotations.

Petukhova and Kochmar (2025) explore multiple configurations for tree construction and annotation, including binary and non-binary structures, frequency-based grouping, and optimal split strategies, and report that the frequency-guided optimal split selection using GPT-4o outperforms crowdworkers on dialog annotation tasks based on the taxonomy of speech functions (Eggin and Slade, 2004), while reducing total annotation time from over 30 hours to under 1.5 hours. Therefore, in our work, we adopt this configuration using the publicly available implementation.²

2.3 Controlled Generation

Controlled text generation (CTG) aims to direct language models to produce outputs that adhere to specific attributes or constraints, such as sentiment, style, or intent. A prevalent method in CTG involves fine-tuning models with prompts that include explicit intent labels, enabling the generation of text aligned with desired behaviors (Liang et al., 2024).

Instruction fine-tuning has emerged as an effective strategy for this purpose. By training models on datasets where prompts are augmented with natural language instructions or intent labels, models learn to condition their outputs accordingly. For instance, the InstructCTG framework demonstrates how conditioning on natural language descriptions and demonstrations of constraints allows models to generate text that satisfies various requirements without altering the decoding process (Zhou et al., 2023).

²<https://github.com/Kpetyxova/autoTree>

This approach is particularly beneficial in educational contexts, where aligning generated content with pedagogical strategies is crucial. By fine-tuning models with prompts that specify instructional intents, AI tutors can provide more effective and tailored support to learners (Jia et al., 2025).

3 Re-annotating the MathDial Dataset

3.1 Tree Creation

To construct a tree for the extended taxonomy proposed in MathDial, we used the best framework configuration from Petukhova and Kochmar (2025) – frequency-guided optimal split selection and backtracking. This method iteratively selects among the candidate splits by scoring them and choosing the highest-ranked one, with backtracking employed if a viable partition cannot be formed. Additionally, the approach biases the tree construction toward more frequent classes, making them quicker to reach and producing trees that better reflect real-world class distributions. The tree was generated based on eleven intent names and their corresponding examples (see Table 1). The resulting tree, presented in textual form in Figure 1, has a depth of two with five branches emerging from the root node.

Interestingly, most intents are grouped according to high-level categories defined in Macina et al. (2023), except for the *Probing* intents, which are split into two separate groups: (1) *Asking for Explanation* and *Seeking Self-Correction*, and (2) *Perturbing the Question* and *Seeking World Knowledge*. While this split was not predefined, it is interpretable: the first group centers on prompting students to reflect on and assess their reasoning, whereas the second group encourages them to explore broader or external concepts beyond the immediate problem.

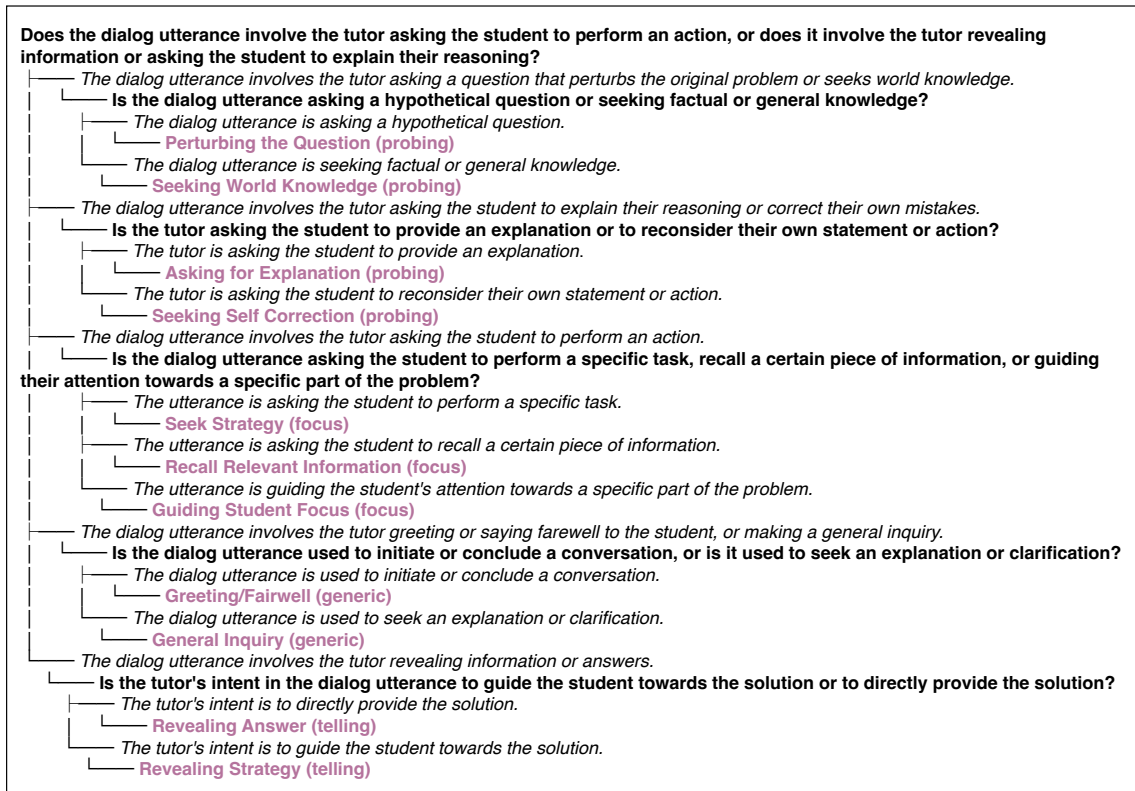


Figure 1: Tree created for the extended taxonomy of the MathDial dataset using the framework from [Petukhova and Kochmar \(2025\)](#). Questions corresponding to tree nodes are in **bold**, possible answers that represent branches are in *italics*, and leaf nodes, representing the eleven intents, are in **purple bold**.

3.2 Annotation

Data Preprocessing Out of 2,861 dialogs, we randomly selected 500 dialogs for training, 100 for validation, and 100 for testing.

An example of the original tutor intent annotation in MathDial is shown in Figure 2. A single label is applied to each teacher utterance in the original annotation, which, while effective for high-level analysis, may limit flexibility in downstream applications requiring finer-grained control. For instance, an utterance [I see.]₁ [But we’re dealing with individual pies here, rather than slices.]₂ [If you had a birthday cake, and lots of guests at your party, you couldn’t just keep producing slices of cake.]₃ [Can you think of another way to figure out how everyone gets a piece?]₄ in MathDial is annotated as *Probing*. However, this utterance comprises several discourse units with distinct functions: segment [1] appears to be *Generic*, segment [2] aligns with *Focus* (specifically, *Guiding Student Focus*) as it redirects the student’s attention, segment [3] fits the *Probing* category, and segment [4] corresponds to *Focus* (*Seek Strategy*) because

it prompts the student to think of an alternative solution.

In contrast, in other cases, the assigned label appears to follow the final part of the utterance. For example, the utterance [But there are always 4 slices in a shepherd’s pie, so using the total number of slices might not be helpful.]₁ [Are there any other quantities you could use to divide by the slices in the pie?]₂ is labeled as *Focus*. Here, while *Focus* applies to the second sentence [2], it would be more appropriate to label the first sentence [1] as *Telling*. This inconsistency – where labels are sometimes based on the first segment and other times on the last – underscores the potential benefits of a more fine-grained and consistent annotation approach for certain downstream tasks.

Ideally, annotation should be performed at the elementary discourse units (EDUs) level rather than entire utterances. EDUs are segments of text that typically correspond to clauses ([Jurafsky and Martin](#)). Therefore, in this work, we preprocess the data by first splitting teacher utterances into EDUs.

Since no state-of-the-art method currently ex-

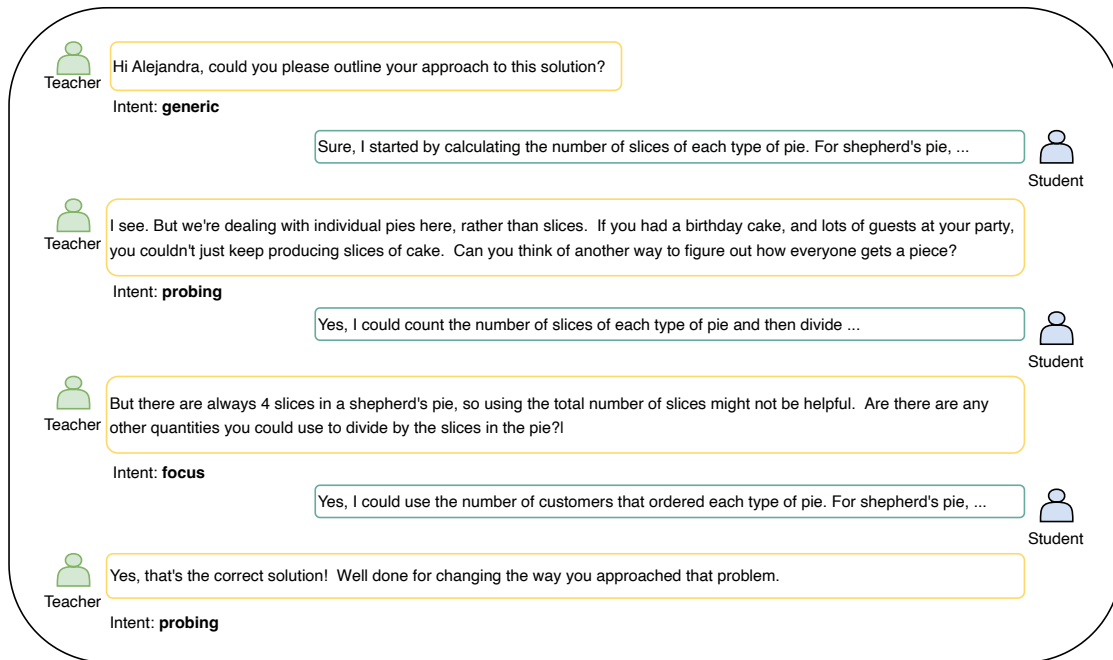


Figure 2: An example of teacher utterances and their annotated intents from MathDial.

ists for automatically dividing the text into EDUs, we use the following strategy: (1) Punctuation Removal: first, we remove all punctuation from the utterances; (2) Punctuation Restoration: next, we restore the punctuation using a model trained for this task;³ (3) Comparison and Segmentation: finally, we compare the original utterance with the punctuation-restored version. If the restored punctuation replaces a comma in the original text with a period, question mark, or exclamation mark, we split the utterance at that comma, thereby creating separate EDUs. By default, we also split different sentences into separate EDUs. Each EDU that resulted from the original utterance through this process inherits the original label assigned to the full utterance in MathDial (i.e., one of the four high-level categories).

After the data is split into EDUs, the number of resulting teacher utterances in the **train** split is **5,174**. The **validation** and **test** sets are similarly segmented into EDUs and limited to **100** teacher utterances each.

Annotation Using the generated tree, a GPT-4o-based annotation pipeline from [Petukhova and Kochmar \(2025\)](#) is applied. Since the tree’s structure aligns with the hierarchical intent relationships

³<https://huggingface.co/oliverguhr/fullstop-punctuation-multilang-large>

proposed by the authors of MathDial, we can reasonably expect that annotation based on this tree will reflect those relationships. For instance, if the annotation using the tree assigns the label *Perturbing the Question*, the original annotation should correspondingly contain *Probing*, and so on. Based on this alignment, we can evaluate the annotation quality, at least in terms of consistency with the original higher-level annotations.

Table 2 presents weighted precision (P_w), recall (R_w), and F1 ($F1_w$) as well as macro F1 (F1) scores when comparing lower-level intent annotations on the training set to the original high-level teacher move categories. The low scores are expected, given that the original teacher utterances were split into EDUs while retaining the same label. As discussed earlier, different EDUs within the same utterance would often be of distinct types, which was not accounted for in the original annotation in MathDial.

P_w	R_w	$F1_w$	F1
0.40	0.38	0.36	0.27

Table 2: Evaluation of 11-label annotation on the training set, comparing the new alignment with the original 4-label annotation from MathDial, using the annotation framework from [Petukhova and Kochmar \(2025\)](#).

Among the 5,174 teacher utterances, 1,319 remained unchanged from the original dataset, as

they originally consisted of a single EDU. Annotation results for these utterances are presented in Table 3. While these metrics are higher than those in Table 2, they still indicate relatively poor performance.

P_w	R_w	$F1_w$	$F1$
0.48	0.45	0.43	0.31

Table 3: Evaluation of 11-label annotation on the training set utterances that remained unchanged (i.e., originally consisted of a single EDU), comparing the new alignment with the original 4-label annotation from MathDial, using the annotation framework from Petukhova and Kochmar (2025).

However, a manual analysis revealed significant inconsistencies in the original annotation. Consider the following illustrative examples:

- A student initially identifies 14 as the correct final answer to the task. However, during the discussion, the student incorrectly restates the final solution as $10 + 10 + 4 = 24$. The teacher responds, *Is that 14?* — referring back to the earlier moment when 14 was correctly identified as the expected answer (see the full dialog in Appendix A). The tree-based annotation classifies this teacher utterance as *Seeking Self-Correction*, corresponding to *Probing*. However, in the original dataset, it is labeled as *Telling*, which we believe is not accurate.
- The tutor says, *You need to add brackets to (8-2) and remember the order of operations.* The student responds, *Yes, I understand now. The correct equation should be $6 + (8 + 8) - 2 = 22$ new books.* The teacher replies, *No, I said it's 8-2, not 8+8.* Although the tree-based annotation assigns this utterance to *Revealing Answer (Telling)*, the original annotation labels it as *Generic*, possibly reflecting a different interpretation or contextual judgment.
- A student states, $6 + 8 + (8 - 2) = 22$. The teacher responds, *Please explain how you got 22.* The tree-based annotation categorizes this utterance as *Asking for Explanation*, which corresponds to *Probing*. However, in the original dataset, it is labeled as *Generic*, which does not align well with the intent of the utterance.

Given the prevalence of such unclear or ambiguous cases in the original annotation of the dataset,

we cannot conclude that the tree-based annotation is inaccurate. Instead, these inconsistencies in the original annotation suggest that the discrepancies in the evaluation metrics may be due, at least in part, to ambiguities in the original dataset.

The distribution of the eleven predicted intents across all dataset splits (train, validation, and test) is shown in Figure 3.

4 Controlled Generation

To demonstrate the benefits of an extended taxonomy with annotations collected using the framework from Petukhova and Kochmar (2025), we fine-tune an LLM to predict the next teacher utterance. The model is trained using the math task description, its gold solution, the student’s solution, the dialog history, and the teacher’s next utterance intent as predicted by the annotation framework.

Additionally, we fine-tune a second version of the same model using the original four-intent annotation. We then compare the performance of these two fine-tuned models with each other, as well as with the same LLM in its zero-shot setting.

Model We use Mistral-7B-Instruct as the base model for fine-tuning, specifically its 4-bit quantized version from Hugging Face.⁴ The maximum sequence length is set to 1,600. We fine-tune the model using QLoRA (Quantized Low-Rank Adaptation) (Hu et al., 2022), a parameter-efficient method that applies low-rank adapters with quantization to reduce memory and compute costs. We use a rank of $r = 32$ and scaling factor $\alpha = 32$. Fine-tuning is conducted for one epoch with a learning rate of $2e^{-5}$, batch size 8, and gradient accumulation of 4. We employ the AdamW optimizer (Loshchilov and Hutter, 2019), linear scheduling with a warmup (0.1), weight decay of 0.1, and evaluate every 50 steps using SACREBLEU (Post, 2018).

Data Preprocessing We convert the annotated samples into pairs of prompts and gold outputs, where each prompt consists of an instruction, the math task, the gold solution for the task, the student’s solution, the dialog history, and the intent of the following teacher utterance (which is available from the annotated data). While the intent is available as an annotation during both training and evaluation – since we have access to the gold next

⁴<https://huggingface.co/unsloth/mistral-7b-instruct-v0.3-bnb-4bit>

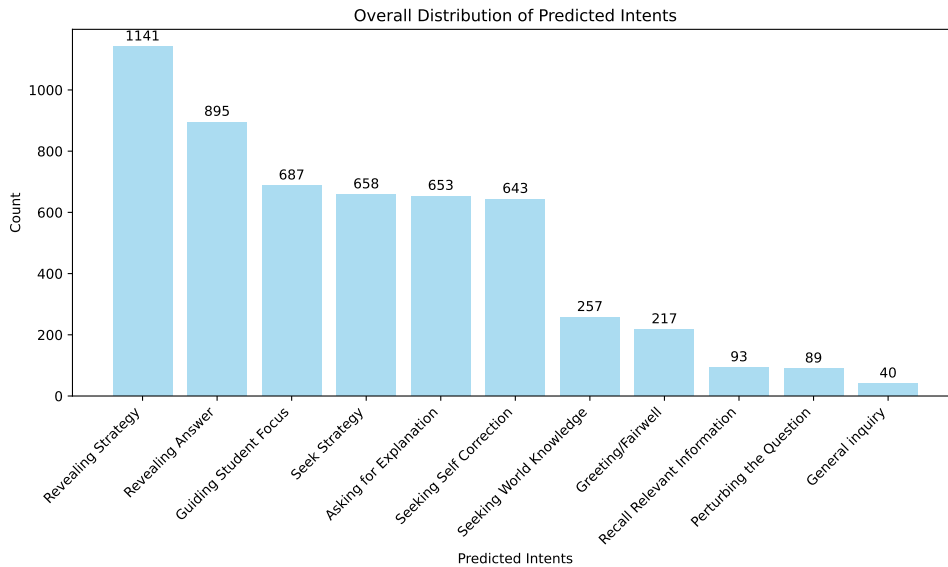


Figure 3: Overall distribution of the eleven predicted intents in the re-annotated dataset.

teacher utterance and can classify its intent – for real-world applications this intent would need to be predicted by a separate model as part of a controlled generation pipeline. The prompt template is shown in Appendix B.

Evaluation We conduct an automatic evaluation of generated outputs using reference-based metrics, including CHRF++ (character n-gram F -score) (Popović, 2017), SACREBLEU (a weighted geometric mean of n -gram precision scores), and ROUGE-1, ROUGE-2, and ROUGE-L (recall-oriented measures of n -gram overlap) (Lin, 2004). In addition, we conduct a small-scale human evaluation.

Results Table 4 presents the generation results for both zero-shot and fine-tuning settings, comparing two annotation schemes: the original four teacher intents provided in the MathDial dataset and the extended set of eleven intents. As expected, the fine-tuned LLM outperforms the zero-shot baseline, and the model trained on the more fine-grained, eleven-intent annotation consistently achieves higher scores across all metrics.

In addition to automated metrics, we conducted a human evaluation with four annotators, each holding at least a Master’s degree in Natural Language Processing. We randomly selected seven dialogs from the test set, resulting in 30 response pairs – one from the model fine-tuned on four intents and one from the model fine-tuned on eleven intents. Each annotator was shown these pairs and asked to decide which response was better or whether

both were equally good or poor (see Figure 4). Based on majority voting, responses from the FT-11 model were preferred in 56.7% of cases.⁵ The inter-annotator agreement, measured using Fleiss’ Kappa, is $\kappa = 0.33$, indicating fair agreement.

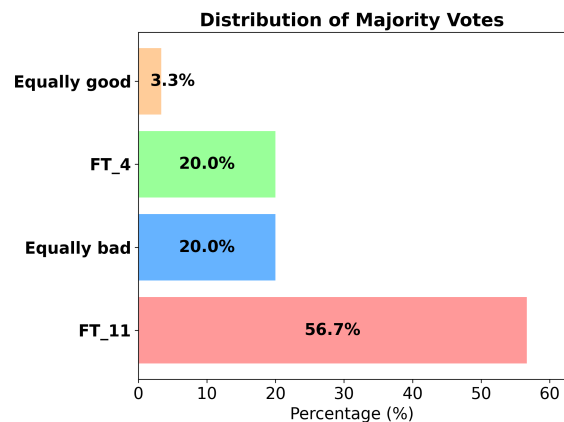


Figure 4: Results of human evaluation. Four annotators were asked to choose the better response or indicate if both were equally good or bad. Final decisions were determined via a majority vote.

Manual analysis (see Figure 5) indicates that the responses generated by the model fine-tuned on eleven intents (FT-11) are superior to those from the model fine-tuned on four intents (FT-4), based on the following observations:

- FT-11 consistently demonstrates a deeper understanding of conversational strategies, such

⁵There were no ties in the majority votes – each example received a clear decision.

Configuration	CHRF++	SACREBLEU	ROUGE-1	ROUGE-2	ROUGE-L
Zero-Shot, 4 intents	16.50	0.93	8.93	2.19	7.10
Zero-Shot, 11 intents	17.11	0.73	8.72	1.95	6.87
Fine-Tuning, 4 intents	16.82	2.67	17.13	5.61	15.95
Fine-Tuning, 11 intents	18.06	4.59	20.73	7.39	19.28

Table 4: Evaluation of controlled generation on the test set from MathDial in a zero-shot setting and with fine-tuned Mistral, comparing using the original four intents from MathDial with eleven intents annotated using the framework from [Petukhova and Kochmar \(2025\)](#).

as using more effective questioning techniques. For instance, when addressing the incorrect prom couples calculation (see the first example in Figure 5), FT-11 explicitly prompts the student to reconsider the original conditions (*So, if we know that there were 123 students at the prom, how many couples were there?*). In contrast, FT-4 merely restates the incorrect scenario (*So, if we have 120 couples, how many students attended the prom?*), which is less effective in guiding the student to realize their mistake.

- FT-11 more directly addresses student misconceptions. In the second example in Figure 5, FT-11 directly questions the student’s arbitrary assumption (*How did you get 100 cows?*), aligning closely with the teacher’s gold standard (*Claire, why did you assume that the farmer had 100 cows?*). FT-4 is less focused, requesting the student to explain calculations instead of addressing the root cause of misunderstanding.
- FT-11 responses tend to be concise yet relevant, prompting students to reflect critically on their reasoning rather than reiterating previous statements. For example, in the third scenario in Figure 5, FT-11 succinctly acknowledges correctness (*Correct.*), aligning well with the actual teacher response (*That’s right.*), while FT-4 unnecessarily repeats previous questions, demonstrating less effective dialog management.

5 Conclusions

In this work, we examined the impact of fine-grained annotation on controlled response generation in the MathDial dataset. By expanding the original taxonomy of teacher moves from four broad categories to eleven more specific intents and using the framework for automated tree creation and annotation from [Petukhova and Kochmar](#)

(2025), we demonstrated that this approach enhances the performance of a fine-tuned LLM in generating meaningful responses.

The results confirm that fine-tuning on a dataset with high-granularity labels leads to better alignment with expected teacher responses, outperforming both the zero-shot setting and fine-tuning on the original four-category annotation. This suggests that the specificity of intent labels is crucial for enhancing the model’s ability to generate targeted and effective tutoring responses.

Furthermore, the manual analysis of the annotation process revealed inconsistencies in the original dataset, highlighting the importance of high-quality, well-structured intent annotation. The tree-based annotation framework proposed in [Petukhova and Kochmar \(2025\)](#) provides a systematic way to refine such datasets, making them more suitable for training controllable generation models.

Limitations

Due to resource constraints, we re-annotate only a subset of the MathDial dataset and utilize a small language model for fine-tuning.

These limitations suggest several directions for future work: (1) re-annotating the entire dataset to enable training of higher-quality models, and (2) exploring larger open-source models for improved fine-tuning performance.

In addition, we acknowledge that our conclusions are based on the use of automated metrics and small-scale human evaluation. In order to demonstrate the real-world impact of these findings and assess the pedagogical value of the generated tutor interventions, future work should consider verifying these conclusions with actual teachers and students.

Ethical Considerations

As this work is exploratory and the outputs of the models used in this research have not been tested with real students, we do not anticipate any significant risks associated with this work or the use

Task	Student's Incorrect Solution	Dialog
Chalktown High School had their prom last weekend. There were 123 students who attended. If 3 students attended on their own, how many couples came to the prom?	Let's call the number of couples that attended the prom "c". We know that there were 123 students total, and that 3 of them attended on their own. This means that the rest of the students attended as couples. So we can set up the equation: couples + 3 = 123 Subtracting 3 from both sides: couples = 120 So there were 120 couples that attended the prom. 120	<p>Teacher: Hey Heidi, do you know what a couple is?</p> <p>Student: Yes, a couple is two people who are together, usually a romantic couple.</p> <p>Teacher: So if we had 120 couples attending the prom, how many students would that be in total?</p> <p>Student: 120 couples would be 240 students in total, since each couple is made up of two people.</p> <p>Teacher: Great.</p> <p>Teacher: would you like to check your solution again, bearing in mind what youve just told me, please?</p> <p>Student: Sure. So if we have 120 couples, that would mean there are 240 students in total. We know that 3 students attended on their own, so if we subtract 3 from 240, we get 237. So there were 120 couples and 237 students in total at the prom.</p> <p>Teacher: We need to read the question again.</p> <p>Teacher: the question tells us how many people were at the prom.</p> <p>Next Teacher Utterance: Gold: lets start over from the beginning and ill help you if needed. FT-4 (focus): So, if we have 120 couples, how many students attended the prom? FT-11 (Seek Strategy): So, if we know that there were 123 students at the prom, how many couples were there?</p>
40% of a farmer's cattle are males. The rest are females. If a female cow produces 2 gallons of milk a day, how much milk will the farmer get a day if he has 50 male cows?	If 40% of the cattle are males, then 100% - 40% = 60% are females. Let's say the farmer has a total of 100 cows, then 40% of 100 = 40 cows are males. So 60 cows are females. If each female cow produces 2 gallons of milk a day, then 60 x 2 = 120 gallons of milk will be produced a day. Since the farmer has 50 male cows, they don't produce milk, so the total milk produced a day is 120 gallons.	<p>Teacher: Hi, could you please walk me through your solution?</p> <p>Student: Sure. I started by figuring out that 40% of the cattle were males, so 100% - 40% = 60% were females. Then I assumed that the farmer had a total of 100 cows, so 40% of 100 = 40 cows were males and 60 cows were females. I then multiplied the number of female cows (60) by 2 gallons of milk a day to get 120 gallons of milk produced a day. Since the farmer only has 50 male cows, they don't produce any milk, so the total milk produced a day is 120 gallons.</p> <p>Next Teacher Utterance: Gold: Claire, why did you assume that the farmer had 100 cows? FT-4 (generic): Hi, could you please explain how you got 120 gallons of milk from 60 cows? FT-11 (Seek Strategy): How did you get 100 cows?</p>
It takes 3 men an hour to complete a job. If they are hired to work on 5 such similar jobs by a company that pays each of them \$10 per hour, calculate the total amount of money the three will earn when they complete the jobs?	It takes 3 men an hour to complete a job, so it will take them 3 x 5 = 15 hours to complete the 5 jobs. If they are paid \$10 per hour, the total amount of money each of them will earn when they complete the jobs is \$10 x 15 = \$150. Therefore, the total amount of money the three will earn when they complete the jobs is \$150 x 3 = \$450.	<p>Teacher: Hi Brenda.</p> <p>Teacher: lets take this step by step.</p> <p>Teacher: How much would one man earn for one job?</p> <p>Student: Hi, one man would earn \$10 for one job.</p> <p>Teacher: How many jobs are there?</p> <p>Student: There are 5 jobs.</p> <p>Teacher: How much would one man doing the 5 jobs earn?</p> <p>Student: One man doing the 5 jobs would earn \$10 x 5 = \$50.</p> <p>Teacher: Great, How many men are there?</p> <p>Student: There are 3 men.</p> <p>Teacher: So each man is paid 50 total for the 5 jobs and there are 3 men.</p> <p>Teacher: so how much is the total for the 3 men?</p> <p>Student: The total for the 3 men is \$50 x 3 = \$150.</p> <p>Next Teacher Utterance: Gold: Thats right. FT-4 (generic): So how much is the total for the 3 men? FT-11 (Revealing Answer): Correct.</p>

Figure 5: Examples of responses generated by a model fine-tuned on the original four-intent annotation set (FT-4) and by a model fine-tuned on data annotated with eleven intents using the framework from [Petukhova and Kochmar \(2025\)](#) (FT-11), alongside actual responses from teachers (Gold).

of the re-annotated dataset. At the same time, we acknowledge that this work uses LLMs, and such models may present risks when applied in real-life educational scenarios, as they may generate outputs that, despite being plausible, are factually inaccurate or nonsensical, which in turn may lead to misguided decision making and propagation of biases. While we do not foresee any immediate risks associated with the research presented in our paper, if future work based on this research ap-

plies presented approaches to real-life scenarios, appropriate safeguards should be applied.

Acknowledgments

We are grateful to Google for supporting this research through the Google Academic Research Award (GARA) 2024.

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*.
- 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.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>, 9.
- 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. *arXiv preprint arXiv:2402.01580*.
- Suzanne Eiggins and Diana Slade. 2004. *Analysing casual conversation*. Equinox Publishing Ltd.
- 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.
- 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.
- Jing Hao, Yuxiang Zhao, Song Chen, Yanpeng Sun, Qiang Chen, Gang Zhang, Kun Yao, Errui Ding, and Jingdong Wang. 2024. Fullanno: A data engine for enhancing image comprehension of mllms. *arXiv preprint arXiv:2409.13540*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Linzhao Jia, Changyong Qi, Yuang Wei, Han Sun, and Xiaozhe Yang. 2025. Fine-Tuning Large Language Models for Educational Support: Leveraging Gagne’s Nine Events of Instruction for Lesson Planning. *arXiv preprint arXiv:2503.09276*.
- Daniel Jurafsky and James H Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.
- John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.
- 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*.
- Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and 1 others. 2024. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*.
- Chin-Yew Lin. 2004. **ROUGE: A package for automatic evaluation of summaries**. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J Malan. 2024. Teaching CS50 with AI: leveraging generative artificial intelligence in computer science education. In *Proceedings of the 55th ACM technical symposium on computer science education V. 1*, pages 750–756.
- Ilya Loshchilov and Frank Hutter. 2019. **Decoupled Weight Decay Regularization**. In *International Conference on Learning Representations*.
- 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*.
- 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*.
- 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.

- Kseniia Petukhova and Ekaterina Kochmar. 2025. A Fully Automated Pipeline for Conversational Discourse Annotation: Tree Scheme Generation and Labeling with Large Language Models. *arXiv preprint arXiv:2504.08961*.
- Maja Popović. 2017. *chrF++: words helping character n-grams*. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- Matt Post. 2018. *A call for clarity in reporting BLEU scores*. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- 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 Drachslers. 2021. Are we there yet?-a systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.
- 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.
- Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pages 42602–42613. PMLR.

A Output of the Tree Creation Step

Task: Harry decided to buy some balloons for his friend's birthday party. One balloon costs \$0.5, and a pack of 10 balloons is cheaper and costs only \$3. Finally, Harry wants to bring to the party exactly 14 balloons. How much did he need to pay for them?

Correct Solution: To buy 14 balloons, Harry should buy one pack of 10, and $14 - 10 = 4$ separate balloons.

The price of 4 balloons is $4 \times 0.5 = \$2$.

In total, the pack and separate balloons would cost Harry in total $3 + 2 = \$5$.

5

Student's incorrect answer: Harry needs to buy 2 packs of 10 balloons, which will give him 20 balloons in total. He will then need to buy 4 more individual balloons to get to 14.

2 packs of 10 balloons cost $2 \times \$3 = \6 .

4 individual balloons cost $4 \times \$0.5 = \2 .

Therefore, Harry needed to pay $\$6 + \$2 = \$8$ for 14 balloons.

8

Dialog:

Teacher: (generic)Hi Michael, could you please walk me through your solution?

Student: Sure. I calculated that Harry needed to buy 2 packs of 10 balloons, which cost $2 \times \$3 = \6 . Then, he needed to buy 4 individual balloons, which cost $4 \times \$0.5 = \2 . So, the total cost was $\$6 + \$2 = \$8$.

Teacher: (focus)So can you please tell me why he needs two packs of 10 (which is 20) and another 4 on top? What's $20 + 4$?

Student: Harry needs two packs of 10 balloons because each pack costs \$3 and buying two packs is cheaper than buying 20 individual balloons. Then, he needs to buy 4 more individual balloons to get to 14 balloons in total. $20 + 4 = 24$, so Harry has enough balloons for the party.

Teacher: (focus)Please could you tell me how many balloons Harry requires in the math problem.

Student: Harry needs 14 balloons for the party.

Teacher: (focus)So is that 24?EOM!Student: No, 14. He needs two packs of 10 balloons and 4 individual balloons, which gives him a total of 24 balloons, but he only needs 14.

Teacher: (focus)So how many packs of 10, plus single balloons does he really need to equal 14?

Student: Harry needs two packs of 10 balloons and 4 individual balloons to equal 14.

Teacher: (probing)so what's $10+10+4$?

Student: $10+10+4 = 24$.

Teacher: (telling)Is that 14?

Student: No, 24 is not 14. Harry needs two packs of 10 balloons and 4 individual balloons, which gives him a total of 24 balloons, but he only needs 14.

Teacher: (telling)So Harry needs one pack of 10, which equals \$3. Plus he needs four separate balloons at \$0.5 each, can you tell me how much he needs to spend?

Student: Harry needs to spend \$3 for one pack of 10 balloons and \$2 for four individual balloons, so the total cost is $\$3 + \$2 = \$5$.

Teacher: (generic)Brilliant, can you see where you were duplicating the packs of 10

Student: Yes, I was calculating that Harry needed to buy two packs of 10 balloons, which cost $2 \times \$3 = \6 . Then, I was calculating that he needed to buy 4 individual balloons, which cost $4 \times \$0.5 = \2 . So, the total cost was $\$6 + \$2 = \$8$, which was incorrect because I was duplicating the packs of 10.

Figure 6: An example dialog from the MathDial dataset.

B The Prompt Template used for Fine-tuning

```
[INST]
You are a math tutor. You will receive a task, a correct ("gold") solution, a student's
incorrect solution, and an ongoing dialog between the student and the tutor.
Your objective is to generate the next tutor response based on the given intent.

Task:
{task}

Gold Solution:
{gold_solution}

Student's Incorrect Solution:
{student_incorrect_solution}

Dialog:
{dialog}

Intent for the Next Tutor Utterance:
{intent}

[/INST]
### Tutor:
```

Figure 7: The prompt template used for fine-tuning on MathDial.

Comparing Behavioral Patterns of LLM and Human Tutors: A Population-level Analysis with the CIMA Dataset

Aayush Kucheria
Aalto University
aayush.kucheria@gmail.com

Nitin Sawhney
Aalto University
nitin.sawhney@uniarts.fi

Arto Hellas
Aalto University
arto.hellas@aalto.fi

Abstract

Large Language Models (LLMs) offer exciting potential as educational tutors, and much research explores this potential. Unfortunately, there's little research in understanding the baseline behavioral pattern differences that LLM tutors exhibit, in contrast to human tutors. We conduct a preliminary study of these differences with the CIMA dataset and three state-of-the-art LLMs (GPT-4o, Gemini Pro 1.5, and LLaMA 3.1 450B). Our results reveal systematic deviations in these baseline patterns, particularly in the tutoring actions selected, complexity of responses, and even within different LLMs.

This research brings forward some early results in understanding how LLMs when deployed as tutors exhibit systematic differences, which has implications for educational technology design and deployment. We note that while LLMs enable more powerful and fluid interaction than previous systems, they simultaneously develop characteristic patterns distinct from human teaching. Understanding these differences can inform better integration of AI in educational settings.

1 Introduction

Large Language Models (LLMs) offer unprecedented capabilities for creating educational technologies that can interact with students. Unlike traditional intelligent tutoring systems (ITS), which were often limited by constrained interfaces and rigid interaction patterns (Alkhatlan and Kalita, 2019; Mousavinasab et al., 2021), LLMs provide natural-language interactions that draw on extensive linguistic and contextual training (Brown et al., 2020; Bommasani et al., 2022). This allows LLMs to respond to learner inputs in ways that more closely resemble human tutors, presenting new possibilities for personalized learning experiences.

Despite their potential, important questions remain about how closely LLM tutoring interactions

align with human tutoring practices. Existing literature on human tutoring and ITSs emphasize strategies such as scaffolding, immediate feedback, and adaptive questioning to meet the learners' needs (Chi et al., 2001; VanLehn, 2011). However, the conversational and pedagogical behaviors of LLMs in tutoring scenarios remain underexplored.

The current work addresses this research gap. Utilizing the CIMA dataset of language teaching dialogues (Stasaski et al., 2020), which contains multiple responses of human tutors to the same students in an Italian language learning context, we systematically examine and compare the structural pedagogical patterns of human tutors and several state-of-the-art language models, GPT-4o (OpenAI et al., 2024), Gemini Pro 1.5 (Team et al., 2024), and LLaMA 3.1 405B (Grattafiori et al., 2024). We identify and characterize behavioral patterns of LLM tutors and human tutors, focusing on action preferences and response complexity.

Our analysis reveals several key findings:

1. Both human and AI tutors show similar high-level preferences in action selection, with hints comprising approximately 45% of all tutoring actions.
2. Human tutors strongly prefer single-action responses (approximately 72% of interactions), while LLM tutors consistently combine multiple pedagogical actions in their responses.
3. Each LLM exhibits its own characteristic pattern, highlighting the need for LLM-specific tailoring.

As these systems continue to evolve and be deployed in diverse learning contexts, recognizing their distinctive behavioral patterns becomes increasingly important—not to eliminate differences, but to use them more effectively in creating educational experiences that complement human instruction.

2 Related Work

2.1 Evolution of Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) have evolved significantly over decades, from rule-based systems with limited interaction capabilities to increasingly sophisticated architectures. Traditional ITS platforms like Cognitive Tutors (Anderson et al., 1995) and knowledge-based tutors (Akkila et al., 2019) demonstrated effectiveness in specific domains but were constrained by rigid interaction patterns and limited adaptability. These systems typically operated within carefully engineered knowledge frameworks, making them powerful but inflexible (VanLehn, 2011; Ma et al., 2014).

The field has progressively sought more natural and adaptive educational technologies. Dialog-based tutoring systems (Graesser et al., 1999; Rus et al., 2013) attempted to incorporate conversational elements but remained limited by predefined pathways. Recent advances in NLP have enabled more sophisticated systems capable of processing and generating natural language interactions (Rus et al., 2013; Nye et al., 2014), setting the stage for the current generation of LLM-based educational tools.

2.2 Language Models in Educational Applications

Large Language Models represent a fundamental shift in educational technology, offering unprecedented fluidity in natural language interaction coupled with broad knowledge coverage. Recent research has explored various applications of LLMs in education, including personalized learning (Park et al., 2024), assessment (Wang et al., 2024), and tutoring (Kumar et al., 2024).

Studies have demonstrated LLMs' potential to support complex learning processes through adaptive dialogue (Schmucker et al., 2023) and to generate contextually relevant explanations (Naik et al., 2024). LLMs' performance as educational tools has primarily been studied through various metrics such as learning gain (Pardos and Bhandari, 2023) or through assessing the quality or correctness of LLM responses (Kumar et al., 2024).

However, while these systems enable more natural interaction, they simultaneously operate according to statistical patterns learned during training rather than pedagogical principles explicitly encoded by designers (Brown et al., 2020; Bommasani et al., 2022). This tension between fluid

interfaces and underlying fixed statistical patterns remains underexplored in educational applications of LLMs.

2.3 Tutoring Patterns and Behaviors

Research on human tutoring has extensively documented the patterns that characterize effective teaching interactions. Chi et al. (2001) identified interactive patterns like scaffolding and feedback loops that support student learning. VanLehn (2011) further explored the balance between different pedagogical moves, noting that expert tutors dynamically adjust their approach based on student needs. Feedback, specifically, has been widely studied, with Hattie and Timperley (2007) emphasizing its critical role in facilitating student learning through targeted interventions.

In comparing AI and human tutoring behaviors, early work by Graesser et al. (1999) examined differences between human tutors and AutoTutor, finding systematic differences in questioning strategies and elaboration patterns. More recent work by Stasaski et al. (2020) with the CIMA dataset highlighted the diversity of valid teaching approaches human tutors employ, noting the low agreement rate (18.1%) between different tutors responding to the same student input. This underscores the complexity of establishing normative patterns for tutoring behavior.

2.4 Interaction Patterns in Language Models

Research on conversational behavior and dialogue generation in LLMs has identified patterns related to turn-taking, conversational coherence, and response complexity (Sandler et al., 2024; Shaikh et al., 2023). These studies highlight that while LLMs produce coherent interactions, the underlying statistical nature can lead to repetitive patterns and superficial dialogues – this behavior has, in part, also led to LLMs being labeled as “stochastic parrots” (Bender et al., 2021).

The few studies that have examined instructional patterns in AI systems have typically focused on direct comparisons of specific responses rather than population-level analysis of behavioral distributions (Puech et al., 2024). These findings emphasize the need to systematically analyze LLM interaction patterns to better understand their educational utility and identify areas for improvement.

2.5 Research Gap

Our research addresses the need to systematically analyze LLM interaction patterns by conducting a detailed comparison of human and LLM tutoring patterns across multiple dimensions of analysis, focusing on action distributions, response complexity, and teaching dynamics. This population-level approach provides a new perspective on how LLMs function in educational contexts compared to human tutors, with implications for both educational technology design and pedagogical theory.

3 Methodology

3.1 Research Questions

This study investigates differences between how language models and humans approach the tutoring task. We examine the underlying patterns in how these systems engage with learners compared to human tutors. This focus can be broken down into specific questions in light of ITS and AI:

1. How do artificial tutoring systems function when given the same context as human tutors?
2. What systematic differences emerge in how AI and human tutors structure their teaching interactions?

These questions address core theoretical interests about the nature of LLMs as ITS while avoiding assumptions about what constitutes “correct” or “effective” tutoring. By focusing on behavioral patterns rather than performance metrics, we aim to understand fundamental differences in how artificial and human tutors approach the teaching task.

3.2 Design Principles

Our methodology is shaped by several key principles:

Population-Level Analysis: Rather than attempting direct turn-by-turn comparisons between human and LLM responses, we focus on analyzing aggregate behavioral patterns across the entire dataset. This approach is particularly important given the low agreement rate (18.1%) observed between human tutors in the CIMA dataset.

Reference Distribution Approach: We aggregate human tutor responses to create reference distributions that capture the characteristic patterns of human tutoring behavior. These distributions serve as a baseline for comparative analysis.

Model Comparison: We maintain separate distributions for different LLM configurations, enabling us to distinguish between model-specific behaviors and general LLM characteristics.

This approach reorients our research question from “Does this LLM respond like a human tutor would?” to “Does this LLM’s pattern of action choices align with the patterns we observe in human tutors?”.

3.3 Dataset

Our analysis utilizes the CIMA (Conversational Instruction with Multi-responses and Actions) dataset (Stasaski et al., 2020), which provides tutoring dialogues focused on teaching Italian prepositional phrases to English speakers. The dataset is particularly valuable for our study as it captures multiple valid tutoring responses for each student interaction, reflecting the reality that there is rarely one “correct” way to respond in a tutoring context.

Key features of the dataset include:

- **Multiple Valid Approaches:** For each student utterance, three different tutors provide responses, showing distinct but equally valid tutoring strategies.
- **Action Labeling:** Each response is annotated with pedagogical actions (Hint, Question, Correction, Confirmation, Other).
- **Progressive Learning:** The dataset captures how concepts build across exercises.

The dataset contains 391 completed exercises across 77 students, with each exercise grounded in both visual and conceptual representations. The mean response lengths (6.82 words for students, 9.99 words for tutors) indicate substantive interactions. This richness, combined with explicit action labeling, provides a strong foundation for analyzing how different tutors structure their teaching interventions.

3.4 Dataset Enhancement with AI Tutors

To enable direct comparison between human and artificial tutoring patterns, we enhanced the CIMA dataset by generating parallel responses from state-of-the-art language models. We selected three advanced instruction-tuned models:

- GPT-4o 2024-08-06 (OpenAI) (OpenAI et al., 2024)

- Gemini Pro 1.5 (Google) (Team et al., 2024).
- LLaMA 3.1 405B instruct nitro (Meta) (Grattafiori et al., 2024)

This selection from different providers, each with distinct architectural choices and training approaches, allows us to distinguish between behaviors fundamental to language models in general versus those specific to particular implementations.

For response generation, we developed a structured prompting system that provides each model with equivalent context to what human tutors received in the original dataset. Each interaction uses a prompt template that specifies:

You are a language tutor teaching Italian. Available actions:

- Question: Ask student for clarification or to elaborate
- Hint: Provide indirect guidance
- Correction: Point out and fix errors
- Confirmation: Acknowledge correct responses
- Other: Any other type of response

Context:

- Target phrase (IT): {target_phrase['it']}
- Target phrase (EN): {target_phrase['en']}
- Grammar rules: {grammar_rules}
- Conversation history: {conversation_history}

Please provide a response as a tutor to the student's last message. Respond in JSON format with: { "response": "your response text", "actions": ["your action types"] }

This approach ensures consistent action categorization and response formats across all interactions.

3.5 Analysis Framework

Our analysis examines two key dimensions of tutoring behavior:

- **Action Distribution Analysis:** We examine the relative frequency of fundamental tutoring actions across different populations. This analysis compares the baseline distribution derived from human tutors against Language Model behavior, identifying systematic preferences or avoidances in action selection.

- **Action Combination Analysis:** We investigate patterns in how actions are combined within individual responses, including the typical number of actions per response and the balance between single-action and multi-action responses.

3.6 Methodological Limitations

Our analysis framework operates within several important constraints:

- **Dataset Characteristics:** The study utilizes a dataset limited to Italian preposition instruction with crowdsourced rather than professional tutors.
- **Structural Constraints:** The prescribed JSON response format may influence natural interaction patterns, and the restricted action vocabulary limits expressive range.
- **Model Implementation:** Analysis is limited to three model variants with a single prompt template approach and no model fine-tuning.
- **Scope of Conclusions:** While we can identify alignment or deviation from human behavioral patterns, we cannot evaluate the optimality of tutoring choices or assess the quality of specific responses.

Our focus on action distributions represents a deliberate methodological choice, prioritizing the analysis of strategic-level behavioral alignment over response-level quality assessment.

4 Analysis

Our analysis revealed systematic differences in how language models and human tutors approach the educational task, with patterns emerging across multiple dimensions of analysis.

4.1 Action Distributions

Both human and AI tutors demonstrate a strong preference for hints as their primary teaching action, with hints comprising approximately 45% of all actions across both human and LLM sessions (Figure 1). This suggests fundamental alignment in basic tutoring strategy, possibly reflecting the effectiveness of scaffolded guidance over direct instruction.

However, examining the broader action distributions reveals key differences in pedagogical approaches. Human tutors show a more balanced

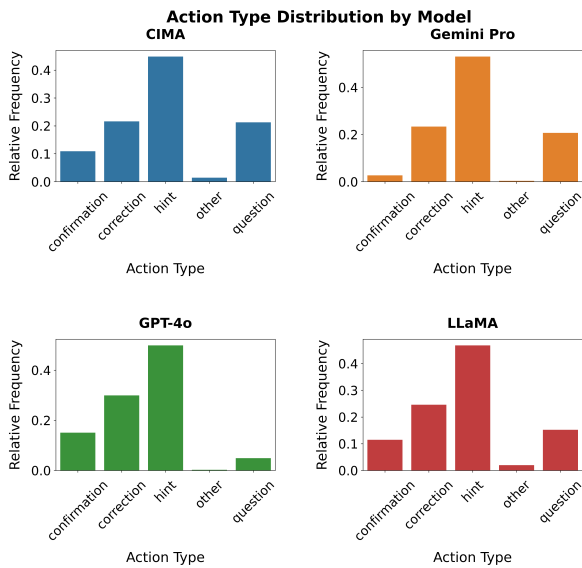


Figure 1: Distribution of actions by different tutors, showing the relative frequency of different pedagogical strategies.

distribution between corrections (20.3%) and questions (21.5%), suggesting a diverse approach. In contrast, AI systems exhibit model-specific patterns - while all maintain the primacy of hints, they differ in secondary strategies. GPT-4o and Gemini Pro 1.5 demonstrate a stronger tendency toward corrections (28.7% and 29.4% respectively) compared to questions (7.3% and 6.8%), while LLaMA 3.1 maintains a more balanced profile closer to human tutors.

Statistical analysis confirmed that the observed differences in action distributions between human and AI tutors were significant ($\chi^2 = 495.17$, $p < .001$, Cramer's $V = 0.124$), indicating a weak to moderate effect size. This suggests an interesting pattern: while there is fundamental alignment in primary teaching strategies (the preference for hints), significant differences emerge in how secondary strategies are deployed. This nuanced finding reveals that LLMs have captured core aspects of tutoring behavior while diverging in other dimensions.

4.2 Response Complexity

The most striking difference between human and AI tutors emerges in response complexity (Figure 2). Human tutors demonstrate a strong and consistent pattern for single-action responses, with approximately 71.8% of responses containing just one action, 24.6% containing two actions, and only 3.6% containing three or more. This pattern sug-

gests a teaching strategy focused on clear, targeted interventions.

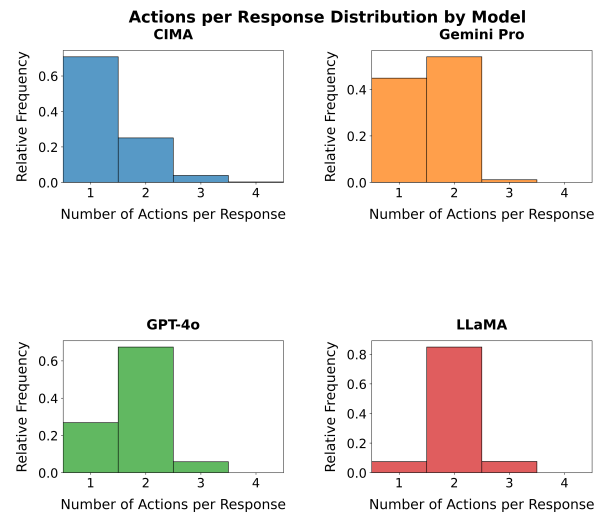


Figure 2: Distribution of the number of pedagogical actions per response in tutoring sessions.

In contrast, AI tutors consistently combine multiple actions in their responses, though with interesting variations between systems. LLaMA shows the strongest preference for dual-action responses (82.3%), while GPT-4o and Gemini Pro display a more balanced distribution. GPT-4o uses single actions in about 31.5% of responses and dual actions in 64.7%, while Gemini Pro shows a more even split between single (42.8%) and dual actions (54.9%).

A Kruskal-Wallis test revealed significant differences in the number of actions per response across the four tutor types (human and three LLMs) ($H = 1507.37$, $p < .001$). Post-hoc pairwise comparisons with Bonferroni correction showed significant differences between humans and each LLM ($p < .001$ for all comparisons), as well as between all pairs of LLMs ($p < .001$). This confirms that not only do AI tutors differ from human tutors in response complexity, but each AI model exhibits its own statistically distinct pattern in how it structures responses.

5 Discussion

5.1 Summary of Research Findings

Our study provides a comparative population-level view of how LLM tutors and human tutors approach the teaching task.

Our first finding is that both human tutors and LLM tutors share a high-level strategy, where hints are the main tutoring action (approximately 45%

of all actions each). This suggests that LLMs have learned to prioritize guidance much like human experts. The secondary actions show some differences. Human tutors use a somewhat balanced mix of questions and corrections in the interactions (roughly 20% each), indicating an approach that alternates between direct feedback and prompting student thinking. For the secondary actions, LLM tutors show skewed distributions; for example, GPT-4o and Gemini 1.5 rely more heavily on corrections, whereas LLaMa 3.1 maintained a more human-like balance.

These differences in action preference suggest that while LLMs have captured the primary tactic of hinting, they diverge in how they follow up, either by explaining or correcting or by probing the learner.

The second finding is the strong contrast in response complexity between human tutors and LLMs. Human tutors strongly prefer a concise, single-action response (roughly 70% of human tutor responses in the dataset had only one pedagogical action). In comparison, LLM tutors frequently combine multiple actions in a single response; the difference was the strongest for LLaMa 3.1, where over 80% of the responses had two actions). Statistical tests confirmed that these differences in response complexity are significant.

The third finding is that LLMs have unique behavioral signatures. Although the three evaluated LLMs have been trained with large masses of data, each had their distinct tutoring style. This highlights that the way how an LLM interacts reflects the model’s design choices or fine-tuning. These results extend the prior observations by Graesser et al. (1999), who noted systematic differences in tutoring style between a classical ITS (AutoTutor) and human tutors. We find that LLM-based tutors likewise deviate from human tutors.

5.2 Pedagogical and Practical Implications

The differences identified in our analysis have implications for educational practice and the design of AI tutoring systems. First, the alignment of primary strategy in terms of heavy use of hints highlights that LLMs have converged on a generally effective tutoring practice. This is encouraging from a pedagogical point of view, as hints are known to facilitate learning by prompting student thinking (Chi et al., 2001).

However, the way how LLM tutors use secondary strategies could affect learning in subtle

ways. For example, the LLMs were more likely to provide corrections, and asked prompting questions less frequently than human tutors. Asking questions is often used to encourage active learning – if an LLM tutor predominantly gives corrections, the student might become more passive in the learning process.

On the other hand, providing rapid corrections can be also be beneficial, depending on the scenario. The pedagogical implication is that LLM tutors should be tailored to the contexts and objectives: if the objective is to foster student reasoning, LLMs should be tweaked to ask more open-ended questions rather than providing quick fixes. Furthermore, compound responses might overwhelm the learner, and to avoid this, LLMs should be adapted to match the user competences. That is, there is room for improvement in the pedagogical quality and ability of LLM-driven tutors.

Broadly speaking, our results emphasize that LLM tutors, despite the fluent dialogue, have embedded biases in how they tutor. This resonates with the tension noted by Horvitz between fluid and natural interfaces and the rigid patterns of automated systems (Horvitz, 1999).

5.3 Limitations

Our work comes with a set of limitations, which we acknowledge. Firstly, our study focuses on a single dataset and domain, i.e. the CIMA dataset of Italian language learning dialogues (Stasaski et al., 2020). The tutoring patterns that we focused on (for both humans and LLMs) may be specific to language teaching or even to particular prompts and tasks in CIMA, and it is possible that the balance of actions and complexity would be different to other datasets. This means that the generalizability of the results should be assessed with additional contexts and datasets.

Secondly, our analysis focused on population-level comparisons, but it does not capture how a tutor might adapt over a tutoring session. Human tutors often dynamically adjust their strategy based on students’ progress, but we do not know to what extent this holds for LLM tutors, and our current analysis misses these dynamics.

Thirdly, our annotation strategy was automatic, and relied on the existing categories in the CIMA dataset. It is possible that LLM (or human) actions do not always neatly fall into specific categories. We sought to mitigate this by using clear definitions, but acknowledge the presence of noise. Ad-

ditionally, we relied on LLMs' self-reported action classifications without manual validation. While our population-level patterns are robust to some classification noise, future work should validate the accuracy of these self-classifications. Furthermore, our analysis does not capture subtler nuances in responses; as an example, a human tutor might provide a more encouraging response than an LLM, even if both responses are categorized as hints.

Additionally, as the CIMA dataset was released in 2020 and our tested LLMs were trained on data through 2023-2024, it is possible that the dataset appeared in their training corpora. While this does not invalidate our behavioral analysis—the patterns we observe reflect how these models approach tutoring tasks regardless of prior exposure—it should be considered when interpreting the alignment between LLM and human tutoring strategies.

Finally, we cannot deduce the efficiency of the tutoring. This is a limitation of the practical significance of the work. Despite these issues, our work fills a gap by systematically comparing the baseline behaviors of human and LLM tutors.

6 Conclusion

In this paper, we studied LLM-based tutors differ from human tutors in their interaction patterns, conducting a population-level analysis using the CIMA tutoring dataset. Our focus was on the behavioral structure of the tutoring, composed of what actions the tutors take and how they deliver them.

By generating parallel tutoring responses using three state-of-the-art LLMs and comparing them against human tutor responses, we observe the following: (1) LLM tutors and human tutors have similar high-level tactics with a shared emphasis on giving hints, which indicates that current LLMs have learned or been tuned to adopt some of the existing practices of humans; (2) When going beyond the high-level tactics, there are significant differences in how LLM tutors balance their actions and in how complex the responses are; (3) The differences are not uniform across the LLM tutors, which highlights that each LLM has its own “personal” style of tutoring. These findings were made possible by analyzing the aggregate patterns over many tutoring responses, moving beyond anecdotal or one-to-one comparisons.

Recognizing and understanding these patterns is important when seeking to make informed decisions on how to effectively integrate LLMs into

learning environments. The differences that we highlight suggest areas where LLM tutors might benefit from additional tailoring (e.g. tailoring LLMs to the context and objectives, and to match the user competences).

In conclusion, we present a foundation for treating behavioral patterns of LLM tutors as a subject of study by its own right, parallel to how one might study different teaching styles among human tutors. We have also shown how to quantitatively characterize how an LLM “teaches”. Such a characterization can help in aligning LLM tutor behavior towards educational best practices, while also benefiting from the existing capacities such as consistency and breadth.

References

- Alaa N. Akkila, Abdelbaset Almasri, Adel Ahmed, Naser Masri, Yousef Abu Sultan, Ahmed Y. Mahmoud, Ihab Zaqout, and Samy S. Abu-Naser. 2019. [Survey of Intelligent Tutoring Systems Up To the End of 2017](#). *International Journal of Engineering and Information Systems (IJEAIS)*, 3(4):36–49. Publisher: IJARW.
- Ali Alkhatlan and Jugal Kalita. 2019. [Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments](#). *International Journal of Computer Applications*, 181(43):1–20. 70 citations (Semantic Scholar/DOI) [2024-09-14].
- John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray. Pelletier. 1995. [Cognitive Tutors: Lessons Learned](#). *Journal of the Learning Sciences*, 4(2):167–207. 1991 citations (Semantic Scholar/DOI) [2024-09-14].
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2022. [On the Opportunities and Risks of Foundation Models](#). *arXiv preprint. ArXiv:2108.07258* [cs].
- 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](#). *arXiv preprint*. ArXiv:2005.14165 [cs].
- Micheline T.H. Chi, Stephanie A. Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G. Hausmann. 2001. [Learning from human tutoring](#). *Cognitive Science*, 25(4):471–533.
- Arthur C Graesser, Katja Wiemer-Hastings, Peter Wiemer-Hastings, and Roger Kreuz. 1999. [Auto-Tutor: A simulation of a human tutor](#). *Cognitive Systems Research*, 1(1):35–51.
- 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](#). *arXiv preprint*. ArXiv:2407.21783 [cs].
- John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research*, 77(1):81–112.
- Eric Horvitz. 1999. [Principles of Mixed-Initiative User Interfaces](#). pages 159–166.
- Harsh Kumar, Ilya Musabirov, Mohi Reza, Jiakai Shi, Xinyuan Wang, Joseph Jay Williams, Anastasia Kuzminykh, and Michael Liut. 2024. [Impact of Guidance and Interaction Strategies for LLM Use on Learner Performance and Perception](#). *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW2):1–30. ArXiv:2310.13712 [cs].
- Wenting Ma, Olusola O. Adesope, John C. Nesbit, and Qing Liu. 2014. [Intelligent tutoring systems and learning outcomes: A meta-analysis](#). *Journal of Educational Psychology*, 106(4):901–918. 460 citations (Semantic Scholar/DOI) [2024-09-14].
- Elham Mousavinasab, Nahid Zarifsanaiy, 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. 240 citations (Semantic Scholar/DOI) [2024-09-14] Publisher: Routledge _eprint: <https://doi.org/10.1080/10494820.2018.1558257>.
- Atharva Naik, Jessica Ruhan Yin, Anusha Kamath, Qianou Ma, Sherry Tongshuang Wu, Charles Murray, Christopher Bogart, Majd Sakr, and Carolyn P. Rose. 2024. [Generating Situated Reflection Triggers about Alternative Solution Paths: A Case Study of Generative AI for Computer-Supported Collaborative Learning](#). *arXiv preprint*. ArXiv:2404.18262 [cs].
- 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. 207 citations (Semantic Scholar/DOI) [2024-09-14].
- OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, A. J. Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. [GPT-4o System Card](#). *arXiv preprint*. ArXiv:2410.21276 [cs].
- Zachary A. Pardos and Shreya Bhandari. 2023. [Learning gain differences between ChatGPT and human tutor generated algebra hints](#). *arXiv preprint*. ArXiv:2302.06871 [cs].
- Minju Park, Sojung Kim, Seunghyun Lee, Soonwoo Kwon, and Kyuseok Kim. 2024. [Empowering Personalized Learning through a Conversation-based Tutoring System with Student Modeling](#).
- Romain Puech, Jakub Macina, Julia Chatain, Mrinmaya Sachan, and Manu Kapur. 2024. [Towards the Pedagogical Steering of Large Language Models for Tutoring: A Case Study with Modeling Productive Failure](#). *arXiv preprint*. ArXiv:2410.03781 [cs].
- Vasile Rus, Sidney D’Mello, Xiangen Hu, and Arthur Graesser. 2013. [Recent Advances in Conversational Intelligent Tutoring Systems](#). *AI Magazine*, 34(3):42–54. Number: 3.
- Morgan Sandler, Hyesun Choung, Arun Ross, and Prabu David. 2024. A linguistic comparison between human and chatgpt-generated conversations. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 366–380. Springer.
- Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2023. [Ruffle&Riley: Towards the Automated Induction of Conversational Tutoring Systems](#). *arXiv preprint*. ArXiv:2310.01420 [cs].
- Omar Shaikh, Kristina Gligorić, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. 2023. [Grounding gaps in language model generations](#). *arXiv preprint arXiv:2311.09144*.
- Katherine Stasaski, Kimberly Kao, and Marti A. Hearst. 2020. [CIMA: A Large Open Access Dialogue Dataset for Tutoring](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context](#). *arXiv preprint*. ArXiv:2403.05530 [cs].

Kurt VanLehn. 2011. [The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems](#). *Educational Psychologist*, 46(4):197–221.

Rose E. Wang, Ana T. Ribeiro, Carly D. Robinson, Susanna Loeb, and Dora Demszky. 2024. [Tutor CoPilot: A Human-AI Approach for Scaling Real-Time Expertise](#). *arXiv preprint*. ArXiv:2410.03017 [cs].

Temporalizing Confidence: Evaluation of Chain-of-Thought Reasoning with Signal Temporal Logic

Zhenjiang Mao¹, Artem Bisliouk^{1,2}, Rohith Reddy Nama¹, Ivan Ruchkin¹

University of Florida¹, University of Mannheim²
{z.mao,a.bisliouk,namarohithreddy,iruchkin}@ufl.edu

Abstract

Large Language Models (LLMs) have shown impressive performance in mathematical reasoning tasks when guided by Chain-of-Thought (CoT) prompting. However, they tend to produce highly confident yet incorrect outputs, which poses significant risks in domains like education, where users may lack the expertise to assess reasoning steps. To address this, we propose a structured framework that models stepwise confidence as a temporal signal and evaluates it using Signal Temporal Logic (STL). In particular, we define formal STL-based constraints to capture desirable temporal properties and compute robustness scores that serve as structured, interpretable confidence estimates. Our approach also introduces a set of uncertainty reshaping strategies to enforce smoothness, monotonicity, and causal consistency across the reasoning trajectory. Experiments show that our approach consistently improves calibration metrics and provides more reliable uncertainty estimates than conventional confidence aggregation and post-hoc calibration.

1 Introduction

Large language models (LLMs) are increasingly applied in educational contexts such as concept explanation, question answering, and personalized tutoring, especially in STEM domains like mathematics (Kasneji et al., 2023). These models exhibit strong capabilities in solving complex problems; however, they also tend to produce answers that are fluent and seemingly confident, yet factually incorrect. In educational settings, such outputs can be particularly problematic, as students may lack the expertise to distinguish between correct and incorrect reasoning (Polyxeni Paulina Kastania, 2024), and may be misled by responses that appear trustworthy. This mismatch between confidence and correctness raises critical concerns about the reliability of LLM-generated answers and highlights

the importance of integrating uncertainty estimation into educational AI systems.

Although prior work has explored various uncertainty estimation techniques, such as predictive entropy, sampling-based variance, and confidence calibration, most studies focus on general NLP tasks rather than educational scenarios (Zhao et al., 2021; Jiang et al., 2021). In educational contexts like mathematics learning, well-calibrated uncertainty can be especially useful for guiding student attention, supporting teacher oversight, and improving feedback systems. However, existing uncertainty metrics often show poor alignment with actual correctness (Zhu et al., 2025).

In this work, we address this challenge by proposing a novel approach to *estimate uncertainty in LLM-based chain-of-thought (CoT) reasoning* for high school mathematics problems. Our method models the sequence of reasoning steps as a temporal confidence signal and evaluates its structural properties using Signal Temporal Logic (STL) (Fainekos and Pappas, 2006). Instead of modifying the LLM or directly penalizing its outputs, we quantify undesirable confidence behaviors, such as abrupt increases following uncertain steps, by computing robustness scores against formal STL constraints. This yields a constraint-aware aggregation scheme that captures how confidence is expected to evolve over time, offering a structured and interpretable view of the reasoning process while improving calibration. We evaluate our method on a curated dataset of Chinese Gaokao mathematics multiple-choice questions (Zhang et al., 2023). Experimental results show that our method significantly improves calibration, reducing Expected Calibration Error (ECE) compared to baseline uncertainty aggregation methods.

This paper’s contributions are: (1) a novel perspective that treats stepwise confidence in chain-of-thought reasoning as a temporal signal amenable to formal analysis, (2) a constraint-aware modeling

approach that reshapes confidence trajectories and quantifies their structural quality using STL robustness and (3) empirical validation of our method’s effectiveness on Chinese Gaokao mathematics. Fig 1 provides an overview of our pipeline: starting from stepwise CoT confidence, we apply uncertainty reshaping followed by STL-based temporal logic evaluation. This transformation results in interpretable, structure-aware confidence scores.

2 Related Work

Applications of LLMs in Education: LLMs have been widely adopted in educational settings for tasks such as grammar correction, content generation, problem explanation, and intelligent tutoring. Prior studies highlight their potential to support learners across domains like language writing (Kasneji et al., 2023), mathematics (Gan et al., 2023), and personalized feedback (Zhou et al., 2025; Wang et al., 2024). For instance, LLMs like MathGPT and Khanmigo have been used to generate step-by-step math explanations aligned with curriculum standards (Shah et al., 2024), while ChatGPT has shown promise in automated feedback for student essays and short answers (Kasneji et al., 2023). Despite these advances, concerns remain around academic integrity, hallucinated outputs, and students over-relying on unverified responses (Benítez et al., 2024). Moreover, few works explicitly quantify how reliable these educational outputs are, or how uncertainty signals can be used to guide learners or inform teachers. As LLMs become integral to education technology, recent surveys have called for deeper investigations into trust, transparency, and uncertainty in educational applications (Idris et al., 2024).

Uncertainty in LLMs: LLMs exhibit remarkable fluency across diverse NLP tasks, yet their outputs often suffer from overconfidence and miscalibration (Kadavath et al., 2022), especially in impact-sensitive domains such as education. Existing work has explored various uncertainty estimation techniques, including entropy-based methods like predictive entropy and confidence gaps (Zhu et al., 2025), as well as sampling consistency across outputs (Lyu et al., 2025). Confidence calibration is another active area, revealing that LLMs tend to be overconfident, particularly in zero-shot or out-of-domain tasks (Desai and Durrett, 2020; Zhao et al., 2021). Recent studies also propose using uncertainty signals to guide reasoning, such as

Uncertainty-Guided CoT prompting (Zhu et al., 2025), active prompting for data selection, and consistency-based calibration (Diao et al., 2023; Lyu et al., 2025). However, most of these methods have been evaluated on general NLP or code generation tasks, with limited attention to structured educational settings like math problem solving, where reliable uncertainty estimates can help students assess model-generated reasoning and assist teachers in diagnosing student understanding..

CoT and STL: Recent advances in CoT prompting have significantly improved the multi-step reasoning ability of LLMs, yet they also introduce new layers of uncertainty, such as error propagation across intermediate steps and unfaithful explanations (Zhang et al., 2022; Wang et al., 2022; Tanneru et al., 2024). In this work, we propose a novel perspective that treats CoT steps as discrete temporal signals, enabling the use of STL to formally specify and evaluate reasoning quality over time (Rescher and Urquhart, 2012). STL allows for expressive specifications like *eventually correct* or *always consistent*, and provides quantitative robustness scores that capture the degree of satisfaction or violation (Fainekos and Pappas, 2006). This formalism has been successfully applied in domains such as motion planning (van Huijgevoort et al., 2024), reinforcement learning (Li et al., 2017), and control synthesis, and offers a promising path toward interpretable and rigorous evaluation of LLM-generated reasoning trajectories. Applying STL to CoT would not only enable structured detection of flawed reasoning patterns but also facilitate the development of confidence-aware feedback and scoring systems in education and other applications.

3 STL-Guided Confidence Estimation

Our approach consists of three stages: (1) generating a stepwise confidence signal via CoT prompting, (2) applying uncertainty reshaping strategies to promote temporal consistency, and (3) evaluating the reshaped sequence using STL. This pipeline is illustrated in Fig 1(b), showing how each reshaping strategy transforms a sample confidence trajectory. Compared to the original signal, our smoothing strategies effectively suppress abrupt spikes and produce a more temporally coherent confidence trajectory, while preserving the overall trend of reasoning. This behavior is crucial for downstream STL-based evaluation, which benefits

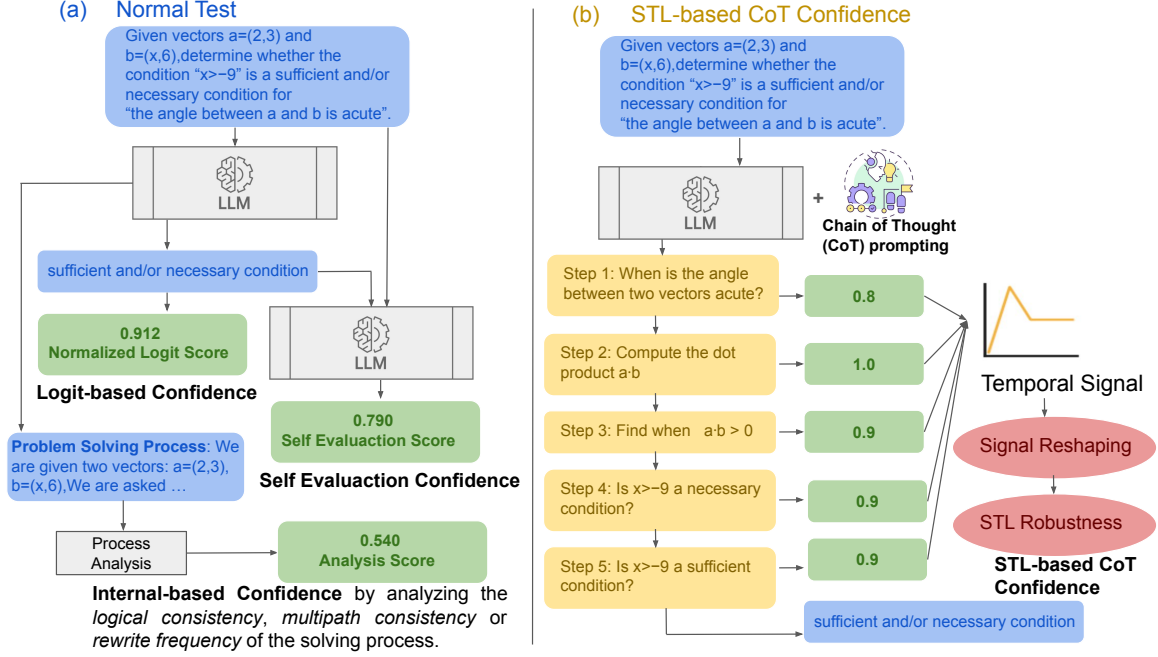


Figure 1: (a) Conventional methods output global confidence via logit, self-evaluation, or internal analysis. (b) Our method models step-wise CoT confidence as a temporal signal, applies signal reshaping, and evaluates robustness using STL to obtain a temporally consistent confidence score.

from smoother and more causally consistent input signals.

3.1 Problem Setup

We model LLMs as autonomous agents tasked with solving high school mathematics problems. Given an input question $q \in \mathcal{Q}$, the agent generates a final answer $a \in \mathcal{A}$ along with a scalar uncertainty score $u \in [0, 1]$, representing its confidence in the answer. Ideally, high confidence should correspond to high correctness probability, and vice versa (Guo et al., 2017).

To evaluate calibration, we employ the *Expected Calibration Error (ECE)* (Naeni et al., 2015; De-sai and Durrett, 2020), a widely-used metric that quantifies the mismatch between confidence and accuracy. Formally, we partition predictions into M bins based on their confidence values and compute:

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|, \quad (1)$$

where B_m denotes the set of predictions falling into the m -th confidence bin, $\text{acc}(B_m)$ is the empirical accuracy defined as

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}[\hat{a}_i = a_i], \quad (2)$$

and $\text{conf}(B_m)$ is the average predicted confidence. Here, n is the total number of examples, and $\mathbf{1}[\cdot]$ is the indicator function that returns 1 if the prediction is correct, and 0 otherwise. Since the task is formulated as multiple-choice classification, both model predictions and ground-truth answers are represented as one of a finite set of discrete options (e.g., A, B, C, D). This allows correctness to be determined via exact match of the selected option label, avoiding ambiguities arising from natural language variation. rect , and 0 otherwise.

In addition to ECE, we also report the *Brier Score (BS)* as a complementary calibration metric. The Brier Score measures the mean squared error between predicted confidence and ground-truth correctness, defined as:

$$\text{BS} = \frac{1}{n} \sum_{i=1}^n (c_i - y_i)^2, \quad (3)$$

where $c_i \in [0, 1]$ is the model’s predicted confidence for example i , and $y_i \in \{0, 1\}$ is the binary correctness label. Lower Brier Scores indicate better calibrated and more reliable confidence estimates.

Unlike conventional classification tasks, reasoning in LLMs often unfolds over multiple steps (Wei et al., 2022). This raises an additional challenge: confidence should not only be calibrated across

Question	Logit	Self-Eval	Internal
Given vectors $\mathbf{a} = (2, 3)$ and $\mathbf{b} = (x, 6)$. Determine whether the condition “ $x > -9$ ” is a sufficient and/or necessary condition for “the angle between \mathbf{a} and \mathbf{b} is acute.” Options: A. Sufficient but not necessary B. Necessary but not sufficient C. Sufficient and necessary D. Neither sufficient nor necessary <i>Model answer: C (incorrect), True answer: B</i>	0.99	0.98	0.99
Set parabola $C : y^2 = 4x$. The focus is F . A line passes through $(-2, 0)$ with slope $\frac{2}{3}$, intersecting C at points M and N . Compute $F\vec{M} \cdot F\vec{N}$. Options: A. 5 B. 6 C. 7 D. 8 (correct) <i>Model answer: D (correct)</i>	0.98	0.95	0.97

Table 1: Examples of high school mathematics questions and confidence scores from three estimation strategies: Logit-based, Self-evaluation, and Internal consistency. While the model is highly confident in all three views, the first question is incorrectly answered, leading to significant miscalibration. The **Expected Calibration Error (ECE)** for Logit, Self-Eval, and Internal confidences are **0.485**, **0.465**, and **0.480**, respectively.

examples, but also evolve smoothly and consistently over the reasoning trajectory (Zhu et al., 2025). Hence, the problem extends to generating temporally coherent uncertainty sequences that reflect both local confidence (per step) and global correctness (final answer). Our objective is thus to design a framework where uncertainty estimates are not only well-calibrated across examples, but also evolve in a temporally consistent manner—exhibiting properties such as smooth progression, causal coherence, and alignment with the underlying reasoning process.

3.2 Uncertainty Reshaping Strategies

To model reasoning-time uncertainty, we use CoT prompting to elicit a sequence of intermediate reasoning steps $\{s_1, \dots, s_T\}$, each associated with a confidence score $c_t \in [0, 1]$. We treat the resulting confidence sequence $\mathbf{c} = \{c_1, \dots, c_T\}$ as a temporal signal (Rescher and Urquhart, 2012). However, due to the inherent causal nature of reasoning, abrupt increases in confidence, especially after initially low-confidence steps, can be misleading (Zhu et al., 2025). To take advantage of this insight, we propose several signal reshaping functions that induce smoother and causally consistent confidence evolution. While some of these strategies are conceptually related to smoothing methods in time-series analysis, they are, to the best of our knowledge, novel in the context of modeling stepwise confidence in LLM-based reasoning.

- **Causal Minimum Smoothing (CMS):** Limits future confidence based on past minimum values plus a small fixed margin δ :

$$\tilde{c}_t = \min \left(c_t, \min_{i < t} c_i + \delta \right)$$

- **Exponential Decay Smoothing (EDS):** Applies exponential smoothing by blending the current value with the average of past values:

$$\tilde{c}_t = \alpha \cdot c_t + (1 - \alpha) \cdot \frac{1}{t} \sum_{i=1}^{t-1} c_i$$

- **Monotonic Penalty Smoothing (MPS):** Dampens confidence spikes if the previous step is below a fixed threshold τ . This focuses on upward spikes, which are more likely to mislead following uncertain steps:

$$\tilde{c}_t = \begin{cases} \frac{c_{t-1} + c_t}{2} & \text{if } c_{t-1} < \tau \text{ and } c_t > c_{t-1} \\ c_t & \text{otherwise} \end{cases}$$

- **Guarded Smoothing (GS):** Caps sudden jumps beyond threshold τ plus tolerance ϵ :

$$\tilde{c}_t = \begin{cases} \tau + \epsilon, & \text{if } c_{t-1} < \tau \text{ and } c_t > \tau + \epsilon \\ c_t, & \text{otherwise} \end{cases}$$

The reshaped sequence $\tilde{\mathbf{c}} = \{\tilde{c}_1, \dots, \tilde{c}_T\}$ is passed to a formal temporal logic evaluation module described next as shown in Fig. 1(b).

3.3 STL-Based Temporal Evaluation

Rather than relying solely on the final-step confidence c_T or averaging all stepwise confidences, we propose a STL-based framework to evaluate the temporal structure of the confidence trajectory $\tilde{\mathbf{c}} = \{\tilde{c}_1, \dots, \tilde{c}_T\}$ (Fainekos and Pappas, 2006). STL enables formal specification of desired temporal properties of confidence during reasoning, such as smooth progression or eventual certainty.

Each STL formula encodes a specific temporal pattern, and its associated robustness score

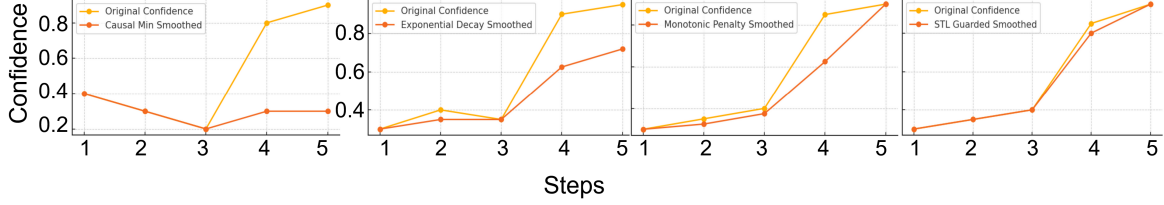


Figure 2: Visualization examples of stepwise confidence signals before and after applying different uncertainty reshaping strategies. Each subplot compares the original confidence trajectory (yellow) against a smoothed version (orange) using one of the following methods: (left to right) Causal Minimum Smoothing (CMS), Exponential Decay Smoothing (EDS), Monotonic Penalty Smoothing (MPS), and Guarded Smoothing (GS). These transformations produce smoother and more temporally consistent confidence profiles while preserving the overall trend.

$\rho_i = \rho(\tilde{c}, \text{STL}) \in \mathbf{R}$ quantifies how well the reshaped signal satisfies that property (Donzé and Maler, 2010). A positive score indicates the satisfaction margin, while a negative score represents the magnitude of a violation.

We define three STL specifications, each yielding a separate confidence score:

- **Eventually Confident:** Confidence should eventually rise above a threshold τ :

$$\text{STL1} = \diamond_{[t_1, t_2]}(\tilde{c}(t) > \tau)$$

- **Always Stable or Increasing:** Confidence should not drop abruptly:

$$\begin{aligned} \Delta\tilde{c}(t) &= \tilde{c}(t) - \tilde{c}(t-1) \\ \text{STL2} &= \square_{[t_1, t_2]}(\Delta\tilde{c}(t) \geq -\epsilon) \end{aligned}$$

- **Local Smoothness:** Confidence should not change too much between steps:

$$\text{STL3} = \square_{[t_1, t_2]}(|\Delta\tilde{c}(t)| \leq \delta)$$

Each resulting score $\hat{c} = \text{ReLU}(\rho(\tilde{c}, \text{STL})) \in [0, 1]$ represents an interpretable, temporally-informed confidence score derived from logic-based robustness. These scores can be used independently for analysis or combined in multi-dimensional calibration evaluation.

While our STL-based scoring framework does not require labeled data during reasoning, it does rely on threshold hyperparameters (e.g., τ, ϵ, δ) that influence robustness computation. To set them, we perform a grid search on a held-out validation set. This makes our approach partially *post-hoc* in nature: only the STL evaluation stage requires data-driven tuning, whereas the preceding confidence reshaping is fully unsupervised and model-agnostic.

4 Experiments

In Section 4, we present an ablation study and a comparison against established post-hoc calibration techniques that investigates the impact of STL parameterization and compares our method against established post-hoc calibration techniques such as *Temperature Scaling* (Guo et al., 2017) and *Histogram Binning* (Zadrozny and Elkan, 2001). Our results show that STL-based evaluation provides not only competitive calibration performance but also interpretable, temporally grounded diagnostics of reasoning quality.

Experimental Setup: We conduct our experiments using the Qwen-7B language model,¹ a high-performing open-source LLM optimized for Chinese and mathematical reasoning tasks. Our evaluation is conducted on all multiple-choice questions from Chinese national college entrance exams (Gaokao) spanning 2010 to 2022, totaling 12 years of official high school mathematics problems. These questions are drawn from GAOKAO-Bench (Zhang et al., 2023), to assess LLMs’ language understanding and symbolic reasoning capabilities using real-world exam data. Table 1 illustrates representative examples.

To prevent the model from relying on surface-level pattern matching or memorized templates, we augment the dataset following strategies inspired by GSM-Symbolic (Mirzadeh et al., 2024), which shows that LLMs often fail when symbols, numbers, or phrasing are changed. We use a more advanced reasoning model – OpenAI’s o1 API to perform all paraphrasing operations in a controlled and semantically faithful manner. Specifically, each original problem is rewritten to preserve logical structure and correct answer while varying lexical expressions (e.g., transforming “find

¹<https://huggingface.co/Qwen/Qwen-7B>

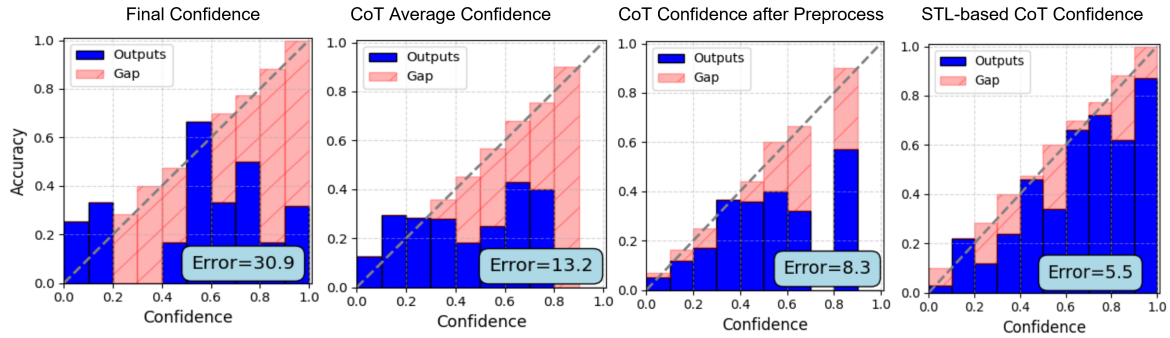


Figure 3: ECE comparison of four confidence estimation methods. (1) Final-step confidence; (2) average confidence over CoT steps; (3) CoT confidence after applying Uncertainty Reshaping Strategies; and (4) STL-based CoT confidence, which combines Uncertainty Reshaping Strategies with the **STL1** formula (*Eventually Confident*: confidence should eventually exceed a threshold τ). The STL1-enhanced method achieves the best calibration (ECE = 5.5).

the intersection of sets A and B” into “determine the elements shared by both A and B”). We further introduce linguistic variation through backtranslation, translating problems into a pivot language (such as French) and then back to English, thereby injecting natural noise without changing semantics. Additionally, symbolic formulations are diversified using template-based transformations—for example, the set expression “ $A \cap B$ ” may be rephrased as “ $x \in Z, \sqrt{x} \leq 4$ ” or “ $B = \{x \in Z \mid x^2 \leq 16\}$ ”. These augmentations collectively evaluate the model’s robustness to paraphrasing, symbol rewriting, and structural variation, ensuring assessment focuses on genuine reasoning rather than memorized syntax. The correct answer for each augmented sample is inherited from the corresponding original Gaokao-Bench problem, since paraphrasing and symbolic transformations preserve semantic and logical equivalence. In total, the original 432 questions are each rewritten twice, resulting in a final dataset of 1,296 problem instances for evaluation.

Quantitative Results and Analysis: To illustrate how different estimation methods affect calibration behavior, Figure 3 shows an example confidence histogram under four representative strategies. While it only reflects a single problem instance and one STL constraint (STL1), the figure demonstrates how reshaping and temporal logic evaluation yield more aligned and interpretable confidence estimates. We now turn to aggregate results across the full test set. All scores are averaged over three runs, and \pm denotes standard deviation caused by the randomness introduced through

temperature-controlled decoding, which affects the variability and creativity of LLM outputs. Table 2 presents the Expected Calibration Error (ECE) for various estimation strategies across three types of uncertainty sources: logits-based, self-evaluation-based, and internal-based. Table 3 complements this with Brier Scores, which jointly capture calibration and sharpness of probabilistic estimates.

From the ECE results, we observe that traditional post-hoc calibration methods such as Temperature Scaling (Guo et al., 2017) and Histogram Binning (Zadrozny and Elkan, 2001) reduce miscalibration compared to raw one-step uncertainty. For example, Histogram Binning achieves an ECE of 0.139 on logits-based predictions, improving substantially over the one-step baseline (0.324). However, these methods operate globally and do not account for the multi-step nature of reasoning in LLMs.

In contrast, CoT-based methods yield stronger performance, especially when combined with our proposed Uncertainty Reshaping Strategies. Simply averaging confidence across CoT steps reduces ECE across all sources, and applying smoothing techniques such as Causal Minimum Smoothing (CMS) or Exponential Decay Smoothing (EDS) brings further gains. For instance, CMS reduces logits-based ECE to 0.107, the lowest among all non-STL methods.

STL-based temporal evaluation further improves calibration. By enforcing high-level temporal constraints like *Eventually Confident* (STL1), *Always Stable* (STL2), and *Locally Smooth* (STL3), the model’s confidence trajectory becomes more inter-

Method	Reshaping Strategy	Logits-based ↓	Self-evaluation-based ↓	Internal-based ↓
1-step Uncertainty	-	0.324 ± 0.045	0.692 ± 0.035	0.694 ± 0.033
Temperature Scaling	-	0.246 ± 0.061	0.158 ± 0.033	0.173 ± 0.046
Histogram Binning	-	0.139 ± 0.004	0.095 ± 0.069	0.185 ± 0.129
CoT Average	-	0.141 ± 0.062	0.542 ± 0.039	0.603 ± 0.037
	CMS	0.107 ± 0.048	0.486 ± 0.040	0.579 ± 0.041
	EDS	0.126 ± 0.022	0.502 ± 0.036	0.573 ± 0.039
	MPS	0.129 ± 0.063	0.530 ± 0.037	0.602 ± 0.038
	GS	0.140 ± 0.060	0.538 ± 0.038	0.579 ± 0.021
STL1 (Eventually Confident)	-	0.174 ± 0.019	0.082 ± 0.021	0.102 ± 0.055
	CMS	0.250 ± 0.198	0.136 ± 0.006	0.119 ± 0.023
	EDS	0.236 ± 0.064	0.098 ± 0.012	0.500 ± 0.497
	MPS	0.211 ± 0.019	0.080 ± 0.017	0.100 ± 0.046
	GS	0.212 ± 0.026	0.077 ± 0.018	0.096 ± 0.035
STL2 (Always Stable)	-	0.170 ± 0.071	0.113 ± 0.011	0.075 ± 0.040
	CMS	0.153 ± 0.021	0.126 ± 0.013	0.074 ± 0.030
	EDS	0.114 ± 0.063	0.114 ± 0.015	0.056 ± 0.028
	MPS	0.188 ± 0.009	0.118 ± 0.015	0.063 ± 0.013
	GS	0.164 ± 0.074	0.111 ± 0.013	0.070 ± 0.039
STL3 (Locally Smooth)	-	0.184 ± 0.021	0.154 ± 0.015	0.099 ± 0.031
	CMS	0.122 ± 0.018	0.081 ± 0.037	0.084 ± 0.030
	EDS	0.149 ± 0.008	0.076 ± 0.040	0.083 ± 0.034
	MPS	0.171 ± 0.025	0.151 ± 0.030	0.083 ± 0.046
	GS	0.180 ± 0.017	0.118 ± 0.006	0.091 ± 0.043

Table 2: Expected Calibration Error (ECE) comparison across confidence sources and estimation strategies. STL-based methods, particularly STL1–STL3 combined with Uncertainty Reshaping (CMS, EDS), consistently yield better calibration than traditional techniques.

pretable and aligned with reasoning quality. STL1 combined with GS achieves an ECE of 0.077 on self-evaluation-based confidence and 0.096 on internal-based, outperforming all other approaches. Notably, STL2 with EDS reaches an ECE of 0.056 on internal-based confidence—the best result across all settings.

The Brier Score analysis mirrors this trend. STL methods consistently produce lower scores than CoT average or standard post-hoc techniques. STL1 with CMS achieves the best self-evaluation-based Brier Score (0.234), while STL2 with EDS offers the best internal-based score (0.056). These results confirm that applying STL logic not only improves calibration error but also leads to sharper, more reliable probability estimates.

Overall, STL-based confidence estimation outperforms traditional calibration and CoT-only baselines, particularly when paired with CMS or EDS. These findings highlight the value of structured temporal logic as a calibration framework for LLM-based reasoning, offering both theoretical guarantees and empirical gains.

5 Conclusion

This paper presents a structured approach to confidence estimation for LLM-based mathematical reasoning. By modeling stepwise confidence as a tem-

poral signal and evaluating its quality using STL, our method addresses limitations in traditional calibration and step-level aggregation techniques. We introduce a suite of uncertainty reshaping strategies and STL-based robustness constraints that enforce desirable properties such as eventual certainty, monotonic progression, and local smoothness.

Experimental results on GAOKAO-Bench demonstrate that our STL-based evaluation, particularly when paired with smoothing strategies like CMS and EDS, consistently achieves lower ECE and Brier Scores compared to standard baselines. Beyond quantitative improvements, the framework provides a principled, interpretable method for diagnosing and enhancing reasoning quality in educational LLM applications.

6 Limitations

While our method improves calibration and interpretability, it is currently limited to high school-level multiple-choice math problems. Extending the framework to open-ended questions, formal proofs, or multi-modal reasoning is a promising direction. Since all experiments are conducted on Qwen-7B, generalization to other models remains uncertain. Testing on models like Gemma 3, Llama 3.2, or DeepSeek would help assess robustness across architectures.

Method	Reshaping Strategy	Logits-based ↓	Self-evaluation-based ↓	Internal-based ↓
1-step Uncertainty	-	0.339 ± 0.019	0.677 ± 0.032	0.678 ± 0.032
Temperature Scaling	-	0.263 ± 0.040	0.255 ± 0.018	0.276 ± 0.018
Histogram Binning	-	0.284 ± 0.002	0.263 ± 0.020	0.259 ± 0.021
CoT Average	-	0.225 ± 0.033	0.498 ± 0.023	0.564 ± 0.028
	CMS	0.218 ± 0.032	0.442 ± 0.019	0.537 ± 0.029
	EDS	0.219 ± 0.032	0.455 ± 0.017	0.530 ± 0.027
	MPS	0.219 ± 0.033	0.483 ± 0.022	0.563 ± 0.028
	GS	0.222 ± 0.033	0.493 ± 0.022	0.536 ± 0.037
STL1 (Eventually Confident)	-	0.223 ± 0.003	0.244 ± 0.001	0.246 ± 0.001
	CMS	0.218 ± 0.005	0.234 ± 0.000	0.240 ± 0.000
	EDS	0.219 ± 0.003	0.238 ± 0.001	0.241 ± 0.001
	MPS	0.221 ± 0.000	0.243 ± 0.002	0.246 ± 0.001
	GS	0.223 ± 0.004	0.243 ± 0.002	0.239 ± 0.001
STL2 (Always Stable)	-	0.238 ± 0.008	0.252 ± 0.005	0.254 ± 0.002
	CMS	0.239 ± 0.009	0.246 ± 0.002	0.249 ± 0.001
	EDS	0.243 ± 0.004	0.253 ± 0.003	0.253 ± 0.001
	MPS	0.239 ± 0.006	0.252 ± 0.004	0.254 ± 0.002
	GS	0.238 ± 0.008	0.252 ± 0.004	0.257 ± 0.005
STL3 (Locally Smooth)	-	0.232 ± 0.011	0.236 ± 0.002	0.240 ± 0.001
	CMS	0.237 ± 0.008	0.241 ± 0.002	0.242 ± 0.001
	EDS	0.239 ± 0.006	0.242 ± 0.001	0.244 ± 0.000
	MPS	0.236 ± 0.010	0.236 ± 0.003	0.238 ± 0.002
	GS	0.233 ± 0.011	0.235 ± 0.001	0.249 ± 0.002

Table 3: Brier Score comparison across methods. Lower scores indicate better-calibrated and sharper probabilistic predictions. STL-based temporal constraints further improve performance beyond CoT averaging and post-hoc calibration.

Both the reshaping strategies and STL specifications are manually defined. Future work could explore learning them dynamically via reinforcement learning or differentiable logic, enabling more adaptive and data-driven calibration beyond manual tuning.

This paper focuses on linear chain-of-thought reasoning. More complex prompting paradigms, such as tree-of-thought (Yao et al., 2023), introduce branching and cyclic structures that pose new challenges for temporal modeling. Extending our evaluation to computation tree logic (CTL) to accommodate such structures would broaden its applicability to richer and more realistic cognitive processes.

Finally, our method is post-hoc and does not influence model inference. Integrating uncertainty feedback into real-time tutoring systems could enable dynamic intervention, hinting, and early error detection.

References

- Trista M Benítez, Yueyuan Xu, J Donald Boudreau, Alfred Wei Chieh Kow, Fernando Bello, Le Van Phuoc, Xiaofei Wang, Xiaodong Sun, Gilberto Ka-Kit Leung, Yanyan Lan, et al. 2024. Harnessing the potential of large language models in medical education: promise and pitfalls. *Journal of the American Medical Informatics Association*, 31(3):776–783.
- Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.
- Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.
- Alexandre Donzé and Oded Maler. 2010. Robust satisfaction of temporal logic over real-valued signals. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 92–106. Springer.
- Georgios E Fainekos and George J Pappas. 2006. Robustness of temporal logic specifications. In *International Workshop on Formal Approaches to Software Testing*, pages 178–192. Springer.
- 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.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.
- Mohamed Diab Idris, Xiaohua Feng, and Vladimir Dyo. 2024. Revolutionising higher education: Unleashing the potential of large language models for strategic transformation. *IEEE Access*.
- Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language

- models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.
- Xiao Li, Cristian-Ioan Vasile, and Calin Belta. 2017. Reinforcement learning with temporal logic rewards. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3834–3839. IEEE.
- Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. 2025. Calibrating large language models with sample consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19260–19268.
- Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.
- Mahdi Pakdaman Naeni, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.
- Nikoleta Polyxeni Paulina Kastania. 2024. Building trust in ai education: Addressing transparency and ensuring. *Trust and Inclusion in AI-mediated Education: Where Human Learning Meets Learning Machines*, page 73.
- Nicholas Rescher and Alasdair Urquhart. 2012. *Temporal logic*, volume 3. Springer Science & Business Media.
- Vedant Shah, Dingli Yu, Kaifeng Lyu, Simon Park, Jiatong Yu, Yinghui He, Nan Rosemary Ke, Michael Mozer, Yoshua Bengio, Sanjeev Arora, et al. 2024. Ai-assisted generation of difficult math questions. *arXiv preprint arXiv:2407.21009*.
- Sree Harsha Tanneru, Chirag Agarwal, and Himabindu Lakkaraju. 2024. Quantifying uncertainty in natural language explanations of large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 1072–1080. PMLR.
- Birgit C van Huijgevoort, Ruohan Wang, Sadegh Soudjani, and Sofie Haesaert. 2024. Specification-guided temporal logic control for stochastic systems: a multi-layered approach. *arXiv preprint arXiv:2407.03896*.
- 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*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.
- Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1.
- Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*.
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.
- Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.
- Yizhou Zhou, Mengqiao Zhang, Yuan-Hao Jiang, Xinyu Gao, Naijie Liu, and Bo Jiang. 2025. A study on educational data analysis and personalized feedback report generation based on tags and chatgpt. *arXiv preprint arXiv:2501.06819*.
- Yuqi Zhu, Ge Li, Xue Jiang, Jia Li, Hong Mei, Zhi Jin, and Yihong Dong. 2025. Uncertainty-guided chain-of-thought for code generation with llms. *arXiv preprint arXiv:2503.15341*.

Automated Scoring of Communication Skills in Physician-Patient Interaction: Balancing Performance and Scalability

Saed Rezayi¹, Le An Ha², Yiyun Zhou¹, Andrew Houriet¹, Angelo D'Addario¹, Peter Baldwin¹, Polina Harik¹, Ann King¹, and Victoria Yaneva¹

¹National Board of Medical Examiners, Philadelphia, USA
{srezayidemne, yyzhou, ahouriet, adaddario, pbaldwin, pharik, aking, vyaneva}@nbme.org

²Ho Chi Minh City University of Foreign Languages, Vietnam
anh1@hufliit.edu.vn

Abstract

This paper presents an automated scoring approach for a formative assessment tool aimed at helping learner physicians enhance their communication skills through simulated patient interactions. The system evaluates transcribed learner responses by detecting key communicative behaviors, such as acknowledgment, empathy, and clarity. Built on an adapted version of the ACTA scoring framework, the model achieves a mean binary F1 score of 0.94 across 8 clinical scenarios. A central contribution of this work is the investigation of how to balance scoring accuracy with scalability. We demonstrate that synthetic training data offers a promising path toward reducing reliance on large, annotated datasets—making automated scoring more accurate and scalable.

1 Introduction

The ability to automatically evaluate free-text responses has become one of the most impactful applications of natural language processing (NLP) in education. Traditionally, research in this area has focused on automated short-answer grading (ASAG) (Haller et al., 2022; Suen et al., 2023; Clauser et al., 2024) and essay scoring (Klebanov and Madnani, 2022). Recently, the scope has expanded to scoring clinical patient notes written by medical students, which involves determining whether critical medical concepts outlined in a scoring rubric are accurately addressed (Sarker et al., 2019; Harik et al., 2023; Yaneva et al., 2024). While traditional ASAG approaches focus on evaluating factual correctness or content coverage in student responses, our work extends this paradigm to assess the quality of interpersonal communication skills—a domain where responses are more nuanced and context-dependent than typical short-answer assessments.

In this paper, we further extend NLP-based scoring in medical education by introducing a new task:

automated scoring of communication skills in a learning tool for physician-patient interactions. Our work is part of the Communication Learning Assessment (CLA) framework (White et al., 2024), a structured educational program that helps medical learners practice communication skills through realistic patient interactions. In a typical CLA scenario, learners watch a brief video of a patient expressing concerns, seeking clarification about their diagnosis, or struggling with treatment adherence, among other examples. They then respond verbally to that scenario (their response can be up to one minute long), aiming to demonstrate key communication behaviors pertinent to the situation. In CLA, these expected behaviors are called learning points (LPs). For example, the LP "Praise patient's weight loss efforts" might be demonstrated by a learner saying, "I'm really proud of you for sticking with it." Evaluating these responses involves identifying specific spans of speech from the learner response that align with LPs from the scoring rubric of the scenario.

The primary contribution of this work is three-fold: (1) we investigate the application of automated approaches for scoring communication skills; (2) we evaluate various techniques aimed at improving model performance; and (3) we consider the scalability of these techniques in practical deployment scenarios. While strategies such as increasing the volume of human-annotated data can enhance performance, they are inherently limited by resource constraints and thus do not support scalable solutions. To address this, we focus on approaches such as data augmentation, few-shot learning, and automated generation of training data—methods that hold promise for improving model performance while maintaining scalability.

2 Dataset and Annotation

The dataset used in this study consists of transcribed learner responses collected from simu-

Case ID	Total	#Positive	#Negative	#LPs
174	162	91	71	3
175	120	71	49	2
176	162	80	82	3
177	236	164	72	4
178	138	55	82	3
180	165	99	66	3
182	232	171	61	4
192	236	134	102	4

Table 1: Summary statistics per case. *#Positive* refers to the number of responses that reflect a learning point (LP). *#Negative* refers to the number of responses constructed without any such reflection (i.e., the learner did not address the LP). *#LPs* denotes the number of distinct LPs associated with each case.

lated physician-patient interactions across 8 clinical cases (see Table 1). Each case contains between 120 and 236 learner responses. The learners were 3rd and 4th year US medical students who passed the USMLE[®] Step 1 exam¹. Recruitment was carried out by NBME.

Annotations were guided by a detailed rubric capturing key communication behaviors 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 26 unique Learning Points (LPs), each associated exclusively with one of the 8 clinical cases, with each case containing between 2 and 4 distinct LPs. Annotators were instructed to precisely identify reflective text spans corresponding to each LP by providing exact character-level indices within learner responses. Negative samples for each LP were systematically derived by listing all learner responses from the same clinical case that were not annotated as reflecting that specific LP, ensuring comprehensive negative examples.

Annotations were performed by NBME staff members who were trained domain experts in clinical communication. For each case, 60 responses were randomly selected for annotation development. Initially, five responses per case were independently annotated by three senior and two junior annotators, producing a total of 25 annotations. Annotators then discussed these annotations to resolve disagreements and establish consensus. Following this consensus-building step, annotators independently annotated seven additional responses each,

¹A high-stakes US medical licensure exam, <https://www.usmle.org/>

resulting in 35 annotated responses per case. Finally, a senior annotator reviewed and finalized annotations for all 60 responses per case to ensure consistency and annotation quality. On average, each LP received approximately 45 annotations, with the exact count ranging from 20 to 80 per LP. Overall, approximately 60% of annotations explicitly reflected the targeted LPs. Table 1 summarizes detailed annotation statistics by clinical case.

3 Model Adaptation and Training

We base our automated scoring on ACTA (Yaneva et al., 2024), which uses a DeBERTa-large architecture as a sequence-level classifier for identifying exact spans that reflect targeted Learning Points (LPs) in learner responses. Instead of predicting per-token labels, ACTA is trained to output the character-level start and end of the span corresponding to the LP, given the response and LP description as inputs. Training uses cross-entropy loss over all possible spans.

The original LP descriptions in our rubric were intentionally concise for human annotators (e.g., "Risks of MRI"), but this brevity posed challenges for ACTA's sequence classification architecture, which relies on semantic relationships between LP descriptions and response text. Terse descriptions lack the contextual cues necessary for distinguishing between superficially similar content and actual demonstrations of the targeted behavior. To address this limitation, we expanded LP descriptions to include specific behavioral indicators. For example, "Risks of MRI" became "Risks of MRI: Avoid unnecessary, costly, and risky tests," providing explicit guidance about the communication goal and enabling DeBERTa's attention mechanism to better identify relevant response segments.

We explored two approaches for generating these expanded LP descriptions:

- **ACTA-M (Manual Summaries):** Domain experts manually created enhanced descriptions for LPs with fewer than 20 positive annotations, incorporating clinical expertise to capture nuanced communication behaviors.
- **ACTA-A (Automated Summaries):** We used Qwen2.5-32B-instruct (4-bit) (Bai et al., 2024) to automatically generate augmented LP descriptions by synthesizing patterns from aggregated positive annotations, providing a scalable alternative to manual expansion.

Case ID	One model per case			One model for all cases			LLM scoring	
	Original LPs	ACTA-A	ACTA-M	Original LPs	ACTA-A	ACTA-M	Qwen	GPT
174	0.905	0.896	0.899	0.894	0.915	0.917	0.835	0.858
175	0.949	0.966	0.949	0.917	0.966	0.966	0.912	0.931
176	0.861	0.865	0.883	0.897	0.886	0.893	0.890	0.849
177	0.927	0.944	0.943	0.930	0.936	0.953	0.936	0.915
178	0.883	0.930	0.930	0.930	0.930	0.930	0.848	0.852
180	0.928	0.939	0.955	0.933	0.956	0.934	0.974	0.942
182	0.976	0.928	0.969	0.983	0.972	0.976	0.931	0.880
192	0.931	0.948	0.934	0.945	0.948	0.943	0.922	0.820
Average	0.920	0.927	0.933	0.929	0.938	0.939	0.906	0.881

Table 2: Comparison of binary F1 scores for ACTA with original and augmented learning point descriptions (ACTA-A and ACTA-M), and LLM-based scoring (i.e., a few-shot approach).

For evaluation, we employed 5-fold cross-validation at the case level, distributing the 8 clinical cases such that each fold used 6-7 cases for training and 1-2 cases for testing. This ensures the model is evaluated on entirely unseen clinical scenarios. We fine-tuned² DeBERTa-large on each fold’s training data. For automated summaries (ACTA-A), descriptions were generated separately for each fold using only that fold’s training annotations to prevent information leakage.

In addition to these two augmented versions of ACTA, we evaluated two other methods:

- **LLM scoring:** a few-shot scoring approach using large language models (LLMs) to detect learning points directly from learner responses. To evaluate whether few-shot classification could serve as an effective alternative or complement to ACTA without fine-tuning, we experimented with two large language models: Qwen2.5-32B-instruct (4-bit) and GPT-4o (OpenAI, 2024). Qwen was selected due to its instruction-tuning and demonstrated effectiveness in similar instructional tasks. GPT-4o was chosen based on its advanced reasoning capabilities and broad applicability to instructional scenarios (Brown et al., 2020; Wei et al., 2022). These choices align with established best practices in leveraging instruction-tuned language models for few-shot classification tasks. For each Learning Point (LP), the models were prompted with detailed task instructions alongside five positive and five negative examples, following a structured few-shot format designed to encourage consistent performance.

²epochs=10, batch_size=8, learning_rate=2e-5, max_length=256

- **Synthetic responses:** To investigate whether synthetic data can effectively address lower scoring accuracy for Learning Points (LPs) with limited annotations, we supplemented our dataset using synthetic learner responses generated by the Qwen2.5-32B-instruct model. For each LP with sparse annotations, we prompted the LLM with task-specific instructions, definitions of the target LP, and selected positive examples from real learner data. The model then generated realistic synthetic responses explicitly demonstrating the targeted LP. This synthetic dataset augmentation enabled us to expand training data without the cost and time constraints of additional student data collection or manual annotation.

Model evaluation was performed using binary F1 score, measuring accuracy in detecting the presence or absence of communication behaviors.

4 Results

Table 2 summarizes the performance of ACTA using the original learning points compared with augmented descriptions (ACTA-A and ACTA-M) using five-fold cross-validation, as well as results from the few-shot approach using LLM scoring and the use of synthetic responses. Manual summaries (ACTA-M) achieved the highest average binary F1 scores (0.933 for the one-model-per-case setting and 0.939 for the one-model-for-all-cases setting), highlighting the moderate effectiveness of human-crafted augmentation. The automated augmentation approach (ACTA-A) also yielded moderate improvements, indicating its potential as a scalable alternative. LLM-based scoring alone did not improve performance, but it produced results

that were nearly comparable to ACTA (e.g., 0.90 vs. 0.933). We note that this was achieved without the need for extensive data collection or human annotation, aside from the need for annotated data for evaluation purposes.

Finally, the use of synthetic responses led to substantial improvements in scalability. Table 3 shows that training ACTA solely on synthetic data (50 generated examples per case-LP pair) provided only moderate performance gains compared to a simple baseline (0.757 vs. 0.723 average binary F1). However, combining a small amount of real annotated data (15 encounters per case) with synthetic responses significantly improved results (0.878 average binary F1), clearly outperforming models trained only on limited real data (0.793 average binary F1). These results indicate that synthetic responses can effectively reduce the need for human annotation without sacrificing performance.

Performance varied across clinical cases, suggesting the benefit of tailoring augmentation strategies to specific learning point characteristics. We also note the comparability of results from using one model for all cases compared to using one model per case.

5 Error Analysis

We conducted an error analysis to understand the limitations and inform future scoring improvements. The analysis focuses on three perspectives: (1) the relationship between annotation quantity and model performance, (2) specific learning points with low model performance, and (3) cases that showed consistently low performance.

First, we observe a clear relationship between the number of positive annotations per LP and binary F1 scores (see Figure 1). LPs with more than 30 annotations generally achieve binary F1 scores above 0.87, indicating that sufficient annotation quantity is critical for model performance. This threshold is empirically derived by examining the distribution in Figure 1, where performance plateaus become apparent.

Second, to understand LPs with low binary F1 scores (< 0.85), we perform both quantitative and qualitative analyses. A systematic human review is conducted where three annotators independently examine 37 false negatives (instances where ACTA failed to identify an originally annotated LP) and 81 false positives. Among the 35 false negatives with complete reviews, only 51.4% were confirmed

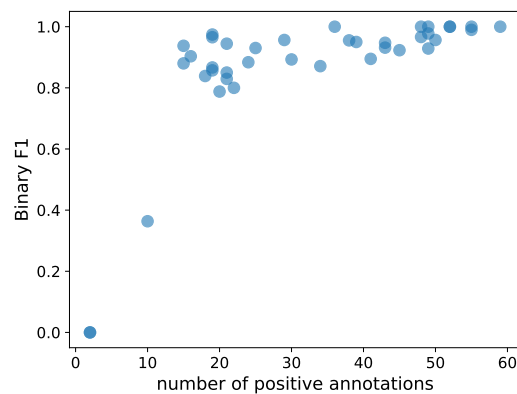


Figure 1: Relationship between the number of positive annotations per learning point and binary F1 score. LPs with more annotations generally achieve higher binary F1 scores. This indicates that sufficient annotations are important for accurate automated scoring.

as true model errors by majority vote, while 48.6% were retrospectively deemed correct predictions by ACTA. This finding reveals that nearly half of apparent model “errors” may actually reflect annotation inconsistencies rather than model failures. The analysis identified three primary factors contributing to lower performance: (1) insufficient positive training examples limiting the model’s exposure to representative spans, (2) inherently ambiguous LP definitions leading to inconsistent interpretations, and (3) the intrinsic subjectivity in identifying nuanced communication behaviors despite our rigorous annotation process.

Third, cases 174 and 176 demonstrated consistently lower performance across multiple LPs. This pattern suggests these cases contain inherently more challenging communication scenarios or LPs that are particularly difficult to identify consistently. This finding emphasizes the need for targeted annotation efforts and potentially refined LP definitions for such challenging cases.

Overall, the error analysis reveals that identifying physician communication behaviors (PCBs) is highly subjective and complex. While our annotation process included consensus-building and final adjudication by senior annotators, the nuanced nature of communication skills—such as distinguishing between implicit and explicit empathy—introduces unavoidable interpretive variation. These findings underscore the importance of sufficient annotation volume and suggest that enhanced annotation guidelines with stricter quality control would be valuable for future iterations.

Case ID	Baseline	50 Synthetic	5 Real + 50 Synthetic	15 Real	15 Real + 50 Synthetic
174	0.712	0.754	0.770	0.763	0.828
175	0.734	0.817	0.855	0.852	0.905
176	0.649	0.684	0.810	0.638	0.861
177	0.813	0.826	0.830	0.900	0.894
178	0.577	0.580	0.732	0.843	0.842
180	0.739	0.812	0.848	0.900	0.909
182	0.841	0.812	0.893	0.743	0.926
192	0.718	0.773	0.823	0.704	0.858
Average	0.723	0.757	0.820	0.793	0.878

Table 3: Binary F1 scores across different training scenarios. **Baseline** assumes every learning point is present (i.e., full recall). **5/15 Real** uses 5 or 15 real annotated encounters per case. **Synthetic** refers to LLM-generated responses (50 per case-LP pair) created using Qwen32B-4bit. Performance is evaluated on real learner responses.

6 Discussion

This study contributes to the ongoing conversation on improving NLP-driven assessment by examining whether data augmentation, few-shot learning, and synthetic data can mitigate the scalability challenges of manual annotation.

Our experiments yielded mixed results. Neither manual (ACTA-M) nor automated (ACTA-A) data augmentation methods showed substantial improvements over the baseline model. Similarly, the few-shot learning models did not outperform the ACTA model. However, the few-shot approach performed almost comparably to the baseline model without the need for extensive data collection or human annotation, which is a significant advantage in scenarios where resources for annotation are limited. A potential explanation for these findings is the relatively small sample size of cases and annotations used, which may have limited the diversity and complexity of the learning points. Moreover, our baseline model—a DeBERTa-based classifier trained with the available annotated data—had already achieved strong F1 scores, reducing the room for significant improvement via augmentation or alternative training strategies.

A key contribution for improving scalability were the synthetic data experiments. Training ACTA exclusively on synthetic responses (generated using Qwen32B-4bit) provided moderate improvements over a naive baseline, indicating synthetic responses alone may be a viable initial training strategy in low-resource settings. However, combining a relatively small set of human-annotated responses with synthetic data significantly increased model performance (average binary F1 from 0.793 to 0.878), clearly demonstrating that synthetic responses can meaningfully reduce the need for extensive manual annotation.

These results suggest that synthetic data is a practical and scalable approach to addressing annotation bottlenecks without sacrificing model accuracy.

Analysis of annotation density (Figure 1) further reaffirmed that performance improves with an increasing number of positive annotations per LP, highlighting the importance of targeted annotation efforts. Additionally, the comparable results from case-specific models and general models suggest that unified modeling strategies may be viable.

7 Limitations

Some limitations of this research stem from the small sample size of annotated responses, and the small number of vignettes. Additionally, resource constraints prevented all responses from being double-rated. While the scoring method remains interpretable by linking LPs to specific phrases in the responses, the neural models used to define phrase boundaries operate as black boxes and require careful evaluation for potential bias.

Although the few-shot LLM-based scoring approach demonstrates promising generalization without explicit fine-tuning, it shows limitations compared to ACTA models. Specifically, few-shot methods heavily depend on prompt quality and the selection of examples provided, making their performance less consistent and potentially sensitive to minor changes in prompt design. Furthermore, few-shot predictions inherently offer lower interpretability than token-level classifiers like ACTA, as LLM decisions emerge directly from prompt conditioning without explicit textual evidence or intermediate classification steps. This reduced transparency can limit their practical usefulness, especially in educational contexts where detailed feedback and justification of model predictions are often necessary. Further research is needed to investigate the

varying levels of risk that the lower explainability of few-shot learning models presents across different assessment domains. These risks can be better understood and mitigated through additional evaluation studies that provide more evidence on how to address potential concerns.

Likewise, the addition of synthetic data for training purposes needs to be carefully evaluated using a high-quality dataset of carefully annotated real learner responses. We note that while synthetic data can meaningfully reduce the need for data collection and human annotation, it cannot fully replace that need as such data will always be needed for a robust evaluation of any scoring system.

8 Ethical Considerations

Like many other products, automated scoring tools function as socio-technical systems, where their impact depends not only on their technical capabilities but also on how they are used and how their outputs are interpreted. Below, we outline specific aspects of the use of this system in different contexts that merit discussion of ethical implications.

In a summative assessment context, the models outlined here are designed as hybrid systems, ensuring that responses from examinees who are borderline or below the passing threshold are always reviewed by human raters. In a formative setting, it is essential to closely analyze how the system's implementation affects learning outcomes, offering critical validity evidence. This includes determining whether automated feedback aids or obstructs skill development, how examinees engage with the feedback, and whether the reliance on automated scoring impacts learning strategies over time. In the case of formative assessment, which is the primary purpose of the CLA tool, a possible negative consequence could also be "washback"—a focus on developing only the skills directly addressed by the tool. It is also crucial to evaluate whether specific learner groups benefit more than others and to identify any unintended effects, such as overdependence on the system or reinforcement of existing biases. A comprehensive exploration of these factors will help ensure that automated scoring systems function as valuable educational tools, rather than mechanical evaluation devices.

The scores provided by the automated scoring engine are currently in their raw form and have not yet been converted into feedback for students or faculty. This transformation into actionable feed-

back is a crucial step because raw scores alone do not provide the necessary context or guidance for improving performance. For students, feedback is essential to understand their strengths and weaknesses, guiding them on how to improve and which areas to focus on. Without clear, specific feedback, students may struggle to make meaningful improvements, as they may not fully understand the implications of their scores or how to address their performance gaps. For faculty, the feedback generated from the automated scores can provide valuable insights into student progress, helping instructors identify areas where students may be struggling, and informing instructional adjustments. This step also allows faculty to engage with the results in a more meaningful way, facilitating a deeper understanding of the learning process and ensuring that the assessment tools align with educational goals. Therefore, transforming raw scores into detailed, constructive feedback is vital to ensure that the automated scoring system contributes effectively to the learning process and supports both student development and instructional improvements.

References

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2024. Qwen2.5: The next generation of qwen large language models. *arXiv preprint arXiv:2407.10671*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Brian E Clauser, Victoria Yaneva, Peter Baldwin, Le An Ha, and Janet Mee. 2024. Automated scoring of short-answer questions: A progress report. *Applied Measurement in Education*, 37(3):209–224.
- 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*.
- Polina Harik, Janet Mee, Christopher Runyon, and Brian E Clauser. 2023. Assessment of clinical skills: a case study in constructing an nlp-based scoring

- system for patient notes. In *Advancing Natural Language Processing in Educational Assessment*, pages 58–73. Routledge.
- Beata Beigman Klebanov and Nitin Madnani. 2022. *Automated essay scoring*. Springer Nature.
- OpenAI. 2024. Gpt-4o system card. <https://arxiv.org/abs/2410.21276>. Accessed: [your access date].
- Abeed Sarker, Ari Z Klein, Janet Mee, Polina Harik, and Graciela Gonzalez-Hernandez. 2019. An interpretable natural language processing system for written medical examination assessment. *Journal of biomedical informatics*, 98:103268.
- King Yiu Suen, Victoria Yaneva, Janet Mee, Yiyun Zhou, Polina Harik, et al. 2023. Acta: Short-answer grading in high-stakes medical exams. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 443–447.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Andrew A White, Ann M King, Angelo E D’Addario, Karen Berg Brigham, Joel M Bradley, Thomas H Gallagher, and Kathleen M Mazor. 2024. Crowd-sourced feedback to improve resident physician error disclosure skills: A randomized clinical trial. *JAMA Network Open*, 7(8):e2425923–e2425923.
- Victoria Yaneva, King Yiu Suen, Janet Mee, Milton Quranda, Polina Harik, et al. 2024. 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.

Decoding Actionability: A Computational Analysis of Teacher Observation Feedback

Mayank Sharma
Stanford University
masharma@stanford.edu

Jason Zhang
Stanford University
jasonzyj@stanford.edu

Abstract

This study presents a computational analysis to classify actionability in teacher feedback. We fine-tuned a RoBERTa model on 662 manually annotated feedback examples from West African classrooms, achieving strong classification performance (accuracy = 0.94, precision = 0.90, recall = 0.96, f1 = 0.93). This enabled classification of over 12,000 feedback instances. A comparison of linguistic features indicated that actionable feedback was associated with lower word count but higher readability, greater lexical diversity, and more modifier usage. These findings suggest that concise, accessible language with precise descriptive terms may be more actionable for teachers. Our results support focusing on clarity in teacher observation protocols while demonstrating the potential of computational approaches in analyzing educational feedback at scale.

1 Introduction

Classroom observation plays a crucial role in evaluating and enhancing instructional quality (Adelman and Walker, 1975; Wragg, 2011). By offering a direct perspective on teaching in authentic settings, it provides insights into how educators engage with students and structure their instruction (Millman and Darling-Hammond, 1990; Putnam and Borko, 2000). It also serves as a vital link between teaching practices and student learning outcomes, thus creating the foundation for teacher professional development (Kane and Staiger, 2012).

Given this significance, the quality of feedback derived from classroom observations is essential (Lazarev and Newman, 2015). While various characteristics contribute to effective feedback, including constructive tone and clarity, research emphasizes that specificity and actionability are particularly crucial for enhancing teacher performance (Archer et al., 2016). Truly actionable feedback provides specific recommendations and clear di-

rection, establishing concrete performance expectations and supporting professional growth (Cannon and Witherspoon, 2005). By focusing on observable teaching behaviors rather than personal attributes, such feedback enables meaningful instructional improvements (Archer et al., 2016).

Although research on actionable feedback originated largely outside education, its principles have proven directly applicable to classroom contexts. In organizational psychology, Cannon and Witherspoon (2005) identified key elements of actionable feedback: specificity, balanced positive and constructive components, and clear connections between observed behaviors and suggested improvements. This aligned with Kluger and DeNisi (1996)'s comprehensive meta-analysis of over 3,000 feedback interventions, which found that feedback effectiveness varies dramatically based on specificity and delivery characteristics. Within education-specific research, multiple studies have confirmed and extended these general principles. For example, Allen et al. (2011) demonstrated that structured feedback systems yield measurable improvements in teaching quality. Similarly, Thurlings et al. (2013) found that effective teacher feedback typically contains explicit behavioral descriptions, rationales for suggested changes, and concrete examples of alternative approaches. Quantitative evidence from Steinberg and Sartain (2015)'s analysis of over 12,000 teacher observation records showed that feedback incorporating concrete examples and precise language led to measurable gains in subsequent evaluations. In a similar way, Hill et al. (2012) established that feedback quality directly correlates with improvements in instructional practice, particularly when including specific action steps. In fact, Darling-Hammond et al. (2017)'s work on professional development systems reinforces the critical role of actionable feedback as a bridge between observation and implementation.

Despite the importance of actionable feedback, classroom observers often struggle to provide guidance that teachers can readily implement (Kraft et al., 2018). This implementation gap stems from inconsistent understanding of what constitutes actionable feedback and the absence of systematic approaches to analyze feedback quality at scale. Computational approaches offer promising avenues for analyzing observation feedback and identifying patterns in actionable feedback. However, applying these methods to classroom observation requires addressing how actionability can be computationally defined and recognized. Our research bridges educational theory and computational methods to develop methods that can meaningfully evaluate the actionability of teacher feedback.

2 Prior Work

While we were not able to identify any existing studies specifically focused on using NLP approaches to identify actionable teacher feedback, adjacent educational research provides relevant context for our work. In the domain of classroom observation, Demszky et al. (2021) analyzed linguistic features in teacher speech to evaluate instructional effectiveness. Similarly, Suresh et al. (2019) examined different dimensions of teacher feedback, though their work did not address actionability specifically. Beyond teacher-focused research, computational analyses of student-centered feedback have shown promising results. Leeman-Munk et al. (2014) developed methods to evaluate student writing and identify improvement areas, while Madnani et al. (2017) created models for standardized writing assessments that demonstrated reliability comparable to human raters.

The emergence of large language models (LLMs) has also sparked interest in their potential for educational annotation and classification tasks. However, Wang et al. (2023) found that models like GPT struggled to accurately classify nuanced educational distinctions. This aligns with Hardy (2025)’s assertion that classroom settings represent “out-of-distribution” data for LLMs, which are primarily trained on broad internet crawls. Additionally, concerns about data privacy, environmental impact, and the ethics of automated educational assessments complicate their use in education. In contrast, specialized transformer-based models offer more promising results for educational applications. Research indicates that models such as BERT (De-

vlin et al., 2019) and RoBERTa (Liu et al., 2019), when properly trained on educational data, can outperform larger LLMs in classifying teacher-student interactions (Wang et al., 2023). Zhang and Litman (2021) demonstrated that these models can be trained on modest amounts of annotated educational data while maintaining strong performance, making them more practical for applications where annotated data may be limited.

Our Study

Despite substantial research on the importance of actionable feedback, computational approaches for identifying actionability in teacher observation feedback remain largely unexplored. This gap appears to exist primarily because of: (1) the lack of clear, consensus definitions of “actionability” in educational contexts; and (2) the scarcity of annotated datasets, as creating these typically requires time-consuming and resource-intensive manual annotation by educational experts (Shah and Pabel, 2019; Shaik et al., 2022).

Our study addresses these gaps through a novel approach where we first established a training dataset by annotation of approximately 660 instances of classroom observation feedback as either actionable or vague. Using this annotated corpus, we fine-tuned RoBERTa to extend this classification to a much larger dataset of over 12,000 feedback instances. With this comprehensive dataset, we conducted an examination of the linguistic features associated with actionability. These findings hold potential to inform the training of classroom observers, guide the development of automated feedback assessment tools, and help improve teacher professional development.

3 Data

This study utilized a large-scale dataset collected from classrooms in Sierra Leone, Liberia, and Ghana by Rising Academies during 2023-2025. The dataset includes $N = 13,118$ classroom observation records, each documenting teacher feedback provided by trained observers. Descriptive statistics on the schools, grades and subjects from which these observations were sourced are presented in Table 1. As shown, the observations come from 273 schools (approx. 48 observations/school) and were recorded by 76 observers (approx. 173 observations/observer).

Each observation was recorded using a struc-

Category	Value (Percentage)
Observers	
Number of Observers	76
Avg. Observations/Observer	172.6
Schools	
Number of Schools Observed	273
Avg. Observations/School	48.1
School Categories	
Top performing	481 (3.7%)
High Impact	2327 (17.7%)
Middle performing	580 (4.4%)
Moderate	4443 (33.9%)
Developing	3198 (24.4%)
Challenging	501 (3.8%)
Critical	507 (3.9%)
N/A	1081 (8.2%)
Grades	
Grade 1	2393 (18.2%)
Grade 2	2621 (20.0%)
Grade 3	2562 (19.5%)
Grade 4	2949 (22.5%)
Grade 5	1470 (11.2%)
Grade 6	1123 (8.6%)
Subjects	
Math	5201 (39.6%)
Faster Math	1978 (15.1%)
Reading	2806 (21.4%)
Faster Reading	3133 (23.9%)

Table 1: Descriptive statistics on observations. *Note.* For the purposes of this study, data from Grades 1-6 and 4 subjects: Reading, Math, Faster Reading, and Faster Math, were selected. Faster Reading and Faster Math are accelerated learning programs designed to supplement regular school curricula ($N = 13118$)

tured two-column format that included: (1) *What Went Well* (WWW) statements, which highlighted teacher strengths or effective strategies, and (2) *Even Better If* (EBI) suggestions, aimed at guiding improvements in teaching practices. As shown in the distribution in Figure 1, the average feedback length was 16.95 words ($SD = 10.83$). While the feedback length was relatively short, there was significant variation in its detail and clarity, ranging from broad praise/criticism to more specific recommendations. Due to the short length, no textual preprocessing was applied.

4 Methods

We organized our study into five sequential phases (visualized in Figure 2):

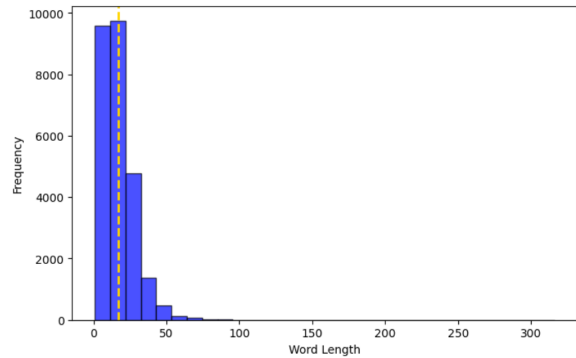


Figure 1: Probability density distribution of feedback length (in words; $N = 13118$)

Phase 1: Annotation and rubric development

In this phase, we developed a training dataset for fine-tuning RoBERTa through a rigorous annotation process. A stratified subsample of 750 comments was selected across multiple dimensions (school categories, grade levels, and subject areas) to ensure representativeness. Two independent researchers annotated each observation feedback according to a standardized rubric derived from established literature (see Appendix A for details), classifying comments as either “actionable” or “vague.” This dual-annotation approach facilitated the calculation of inter-rater reliability using Cohen’s Kappa (κ), yielding a coefficient of 0.60, which indicated moderate agreement.

Discrepancies were methodically resolved through iterative analytical discussions, which simultaneously informed the refinement of our annotation protocol, culminating in the revised rubric presented in Appendix A. The operationalization of “actionable” feedback centered on the presence of concrete, specific suggestions with explicit guidance on both implementation targets and mechanisms. For instance, the comment “*The teacher did great grouping learners and made them pick one word on a flash card where the group later leads in learning the new word. It would be better if the teacher completed the lesson in one hour to allow time for other lessons*” exemplified actionable feedback due to its specific temporal recommendation and clear rationale. Conversely, feedback was classified as “vague” when it lacked implementation specificity, regardless of the presence of ostensibly directive phrases such as “*even better if*” or “*could have.*” The comment “*Giving more energy to make the class exciting was absolutely missing*” illustrates this classification, as it presents an evaluative

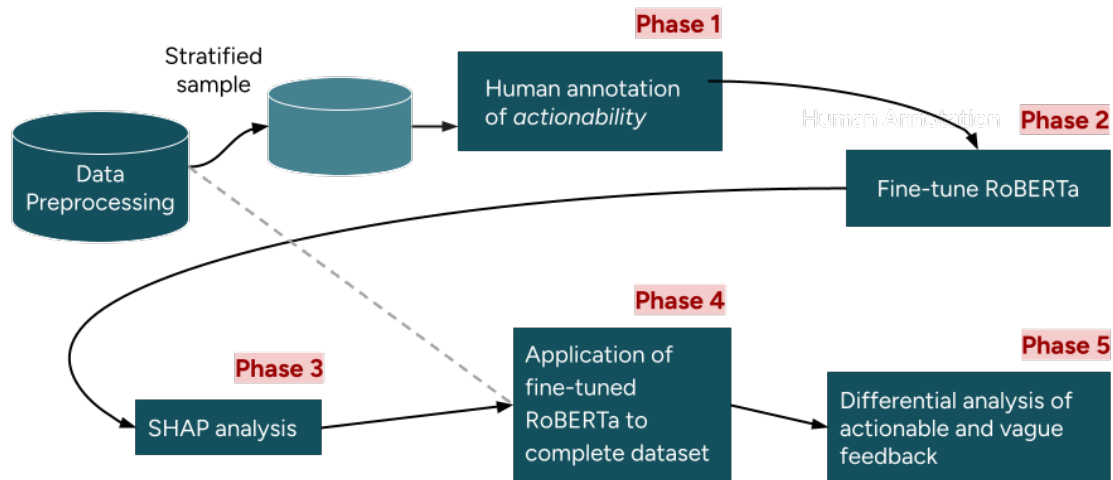


Figure 2: Flow chart for the study

statement without concrete behavioral specifications for improvement. We also excluded instances lacking sufficient evaluative clarity, resulting in a final annotated corpus of 662 comments (reduced from the initial 750). This dataset served as the training corpus for the next phase.

Phase 2: Fine-tuning RoBERTa on the annotated dataset

Model Specification

We fine-tuned RoBERTa (Liu et al., 2019) (using Hugging Face’s `roberta-base`¹) on our annotated dataset. An input token length of 512 was selected for the textual embeddings, with truncation and padding applied as needed. This vector was thresholded using `threshold=0.6` to produce the output vector. We chose this value to prioritize precision over recall, as our context requires high-confidence predictions of actionability rather than maximizing the identification of all potentially actionable feedback.

Training

The model was trained on a T4 GPU via Google Colab² using adam optimizer with `learning_rate=2e-5` with `linear_decay` of 0.01. For training, a `batch_size=16` was cho-

¹Available at <https://huggingface.co/FacebookAI/roberta-base>

²Available at <https://colab.research.google.com/>

sen, while `batch_size=32` was chosen for evaluation, keeping in mind compute bandwidths. We trained the model for 5 epochs and reported results from epoch 3 as the final epoch because model performance degraded afterward. The standard cross-entropy loss function was chosen (default for `roberta-base`).

Evaluation

To evaluate the model’s performance, we used a held-out test set comprising 20% of the total dataset. The assessment was based on standard classification metrics: accuracy, precision, recall, and f1-score.

Phase 3: SHAP analysis

To gain deeper insights into the textual features driving our model’s decisions, we employed SHapley Additive exPlanations (SHAP) analysis (Lundberg and Lee, 2017). This model-agnostic technique provides interpretability by attributing prediction outcomes to specific input features; in this study, words and phrases within the observation comments.

Recent work by Benslimane et al. (2024) validated SHAP’s effectiveness for analyzing short, informal texts, demonstrating its reliability in identifying semantic patterns including emotional tone, gender references, and political language. Building on this empirical evidence, we applied SHAP to analyze 500 teacher feedback instances (250 ac-

tionable, 250 vague) and quantified each token's influence on model classification decisions. To identify consistent patterns, we aggregated SHAP values by unique tokens, calculated mean importance scores across all samples, and ranked terms according to their average contribution to classification outcomes.

Phase 4: Application of fine-tuned RoBERTa to the complete dataset

We utilized our fine-tuned RoBERTa model to categorize the rest of the observations as “vague” or “actionable.” The model from the best-performing cross-validation fold was selected and used to make these predictions. A custom PyTorch Dataset class was implemented, which tokenized input text using the RoBERTa tokenizer with a maximum sequence length of 512. Tokenized inputs were converted into tensors with appropriate attention masks. To ensure computational efficiency, batch predictions (with `batch_size=32`) were performed using PyTorch's DataLoader. For each batch, input tensors were used to extract logits. From these logits, class predictions were obtained using the `argmax` function, and class probabilities using the `softmax` function. Instances with `softmax` probabilities less than 0.90 were classified as “low probability” instances and removed from the dataset.

Phase 5: Differential Analysis of Actionable and Vague Feedback

In this step, we extracted several linguistic features known to be associated with text clarity, specificity, and directiveness. These features were selected to potentially distinguish actionable observations from vague observations classified in the last step:

1. **Word Count:** We calculated the total number of words in each observation using NLTK's word tokenization. Previous research suggests that actionable feedback tends to be more detailed, which could potentially result in higher word counts that provide implementable information (Winstone et al., 2016).
2. **Reading Ease:** We calculated Flesch reading ease using `textstat`. In this metric, the readability of the observation was scored on a 100-point scale, with higher scores indicating easier reading (Flesch, 1948). More accessible language may correlate with feedback that can be readily understood and implemented. Previous work has demonstrated that Flesch

Reading Ease can be effectively used with short-form textual data such as tweets, and can enable robust analysis of readability even in brief, informal text (Davenport and DeLine, 2014).

3. **Lexical Diversity:** This was calculated using NLTK's word tokenization as the ratio of unique words to total words in the observation text. Higher lexical diversity may indicate more specific feedback, potentially offering clearer guidance for action. Conversely, excessive diversity might introduce complexity that reduces actionability.
4. **Modifier Count:** This was calculated by counting modifiers (adjectives and adverbs) in the observations using spaCy's POS tagger. Higher modifier counts might indicate more descriptive or qualifying language, which could potentially correlate with either actionability.

We ran a logistic regression model that included the linguistic features as predictors and feedback category as the outcome to examine the odds ratio for the categories.

The code used in the study is available on a publicly accessible [GitHub repository](#).

5 Results and Discussion

In this section, we present results from Phases 2-5, as Phase 1 has already been described completely in the Methods section.

Phase 2: Fine-tuning RoBERTa on the annotated dataset

Fine-tuned RoBERTa demonstrated strong and stable performance in distinguishing actionable from vague teacher feedback. Using stratified 5-fold cross-validation on 662 annotated examples, the model achieved a mean accuracy of 0.94, precision of 0.90, recall of 0.96, and f1 of 0.93 across folds, with an F1 standard deviation of 0.03. These metrics reflect performance on held-out validation sets and suggest the model generalizes well despite the modest dataset size. This aligns with findings by Zhang and Litman (2021) that well-curated educational data, even in small quantities, can yield high-performing models when paired with appropriate architectures.

Overfitting was monitored via 5-fold cross-validation and early stopping. The model showed

consistently high validation performance (mean F1 = 0.93, SD = 0.03), with no signs of overfitting.

Phase 3: SHAP analysis

Table 2 presents the top 20 most influential words for actionable and vague feedback based on SHAP analysis (positive values indicate features associated with “actionable” class, while negative values indicate association with “vague” class). The results provide mixed evidence without clear-cut patterns. While some action verbs and specific instructional behaviors (e.g., “struggled,” “checks,” “encourages,” “provide”) appear in the actionable feedback category, and certain comparative and conditional terms (“whether,” “enough,” “instead”) appear in the vague feedback category, the overall linguistic distinctions lack sufficient consistency to draw definitive conclusions. The absence of strong patterns suggests that actionability may be determined by relationship between words rather than individual word choices alone.

Actionable Feedback		Vague Feedback	
Word	SHAP Value	Word	SHAP Value
struggled	0.055	sa	-0.051
checks	0.052	equal	-0.025
genders	0.044	whether	-0.025
sentences	0.033	needed	-0.025
tried	0.030	called	-0.023
encourages	0.029	easier	-0.019
improv	0.029	25	-0.018
introduced	0.027	avoid	-0.017
achers	0.026	kick	-0.017
provide	0.026	8	-0.016
helpful	0.026	enough	-0.016
stage	0.026	q	-0.015
minutes	0.026	arus	-0.015
excellent	0.024	had	-0.015
teaches	0.023	name	-0.015
helped	0.023	creative	-0.014
pared	0.023	instead	-0.014
rew	0.023	enable	-0.013
creat	0.022	supposed	-0.013
days	0.022	note	-0.012

Table 2: Top 20 most important words for feedback classification with their SHAP values.

Phase 4: Application of fine-tuned RoBERTa to the complete dataset

“Low probability” predictions constituted about 329 observations (2.5%) of the total data. After their removal, distribution in the complete dataset was as follows: 52.7% (or $n = 6741$) classified as “vague”, and 47.3% (or $n = 6048$) as “actionable.”

Phase 5: Differential Analysis of Actionable and Vague Feedback

Logistic regression analysis (Table 3 and Figure 3) revealed several significant associations between linguistic features and feedback actionability. Word count showed a strong negative relationship with actionability ($-16.637, p < .001$), indicating shorter feedback was more likely classified as actionable, contrary to our proposed hypothesis.

Flesch Reading Ease demonstrated a strong positive association with actionability ($11.751, p < .001$), suggesting more readable feedback was more likely to be actionable, aligning with our hypothesis about language complexity.

Lexical diversity showed a moderate positive association ($0.418, p < .001$, odds ratio = 1.52), with more varied vocabulary correlating with actionability. Similarly, modifier count had a significant positive relationship ($0.187, p < .001$, odds ratio = 1.21), suggesting adjectives and adverbs may help describe teaching behaviors with needed precision.

Overall, the model showed a pseudo R^2 of 0.159, accuracy of 0.68, precision of 0.70 (actionable), recall of 0.58, F1-score of 0.63, and an AUC-ROC of 0.76.

6 Conclusion

Our study provides empirical support for computational approaches to analyzing actionable teacher feedback. The high performance of our fine-tuned RoBERTa model (accuracy = 0.94, precision = 0.90, recall = 0.96, $f1 = 0.93$) demonstrates that RoBERTa can effectively distinguish between actionable and vague feedback, even with a relatively modest training dataset of 662 annotated examples.

The SHAP analysis revealed several interesting patterns in the linguistic features associated with actionable feedback. Action verbs (e.g., “struggled,” “checks,” “encourages”) and specific instructional behaviors appeared more frequently in actionable feedback, while comparative and conditional language (e.g., “whether,” “enough,” “instead”) was more characteristic of vague feedback. However, these patterns were not uniformly consistent, suggesting that actionability may be determined more by the relationships between words and phrases rather than by individual word choices alone.

An analysis of linguistic features suggested that contrary to our initial expectations, word count showed a significant negative relationship with actionability, indicating that shorter feedback was

Feature	Coefficient	Std Error	Odds Ratio
Word Count	-16.637***	1.510	5.95×10^{-8}
Flesch Reading Ease	11.751***	1.013	1.3×10^5
Lexical Diversity	0.418***	0.029	1.52
Modifier Count	0.187***	0.033	1.21

Table 3: Results of logistic regression predicting feedback actionability (*** $p < .001$)

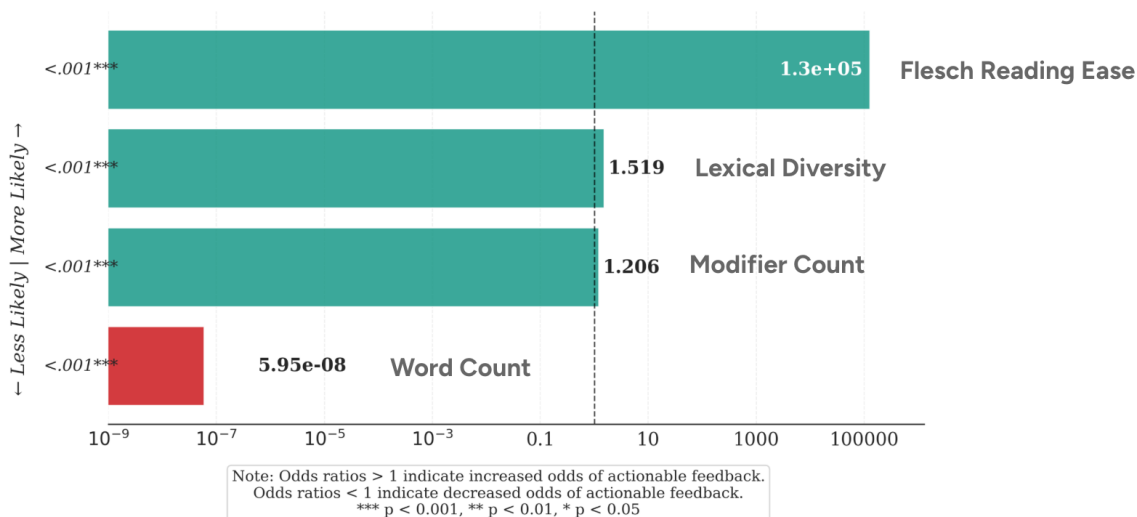


Figure 3: Odds ratios for predictors of actionable feedback

more likely to be classified as actionable. This finding challenges the common assumption that more detailed feedback is necessarily more actionable. It suggests that concision may actually enhance clarity and implementability—verbose feedback might obscure key action points with extraneous information.

The positive association between Flesch Reading Ease and actionability aligns with our hypothesis that more readable feedback is more actionable. This finding indicates that accessible language is crucial for feedback that can be readily understood and implemented.

Our findings suggest three practical implications for teacher professional development. First, our study suggests that concise, readable feedback with precise descriptive language could enhance actionability of feedback given by observers. Second, training programs for classroom observers could benefit from incorporating linguistic guidelines that emphasize readability, appropriate lexical diversity, and effective use of modifiers to enhance feedback actionability. Third, computational approaches like our RoBERTa model could serve as supportive tools for observers to assess and potentially improve the actionability of their feedback before

sharing it with teachers, though such applications should complement rather than replace human judgment.

7 Limitations and Future Work

This study has several limitations that point to directions for future research. While our RoBERTa model performed strongly even with 662 annotated examples, the relatively small training set still poses challenges for generalizability. Its success reflects the effectiveness of fine-tuning on well-curated educational data, but broader representation across feedback styles, school contexts, and observer types would strengthen model robustness and reduce the risk of overfitting.

Second, the scope of this study was limited to early primary classrooms (Grades 1–6) and core subjects (English and Math) in a specific cultural setting. Findings may not fully generalize to other grade levels, subjects, or educational systems. Additionally, because the model was trained on English-language feedback, linguistic and cultural differences in how actionability is expressed remain underexplored.

Third, while SHAP analysis revealed some useful patterns, many influential words, especially

in vague feedback, were ambiguous or context-dependent, highlighting the challenge of capturing actionability through isolated word-level features.

Finally, our binary classification approach, while practical, likely oversimplifies the feedback quality spectrum. Actionability may be better understood as a continuum (from highly vague to highly specific). A multi-point ordinal scale (e.g., 5–7 categories) could offer more granular insights, especially for training observers or improving vague feedback. Moving to such a framework would require more complex annotation protocols, higher inter-rater alignment, and substantially larger datasets—but the added nuance may justify this investment by producing models that offer not just detection, but actionable guidance.

Future work should: (1) expand annotations across more diverse educational contexts, (2) test cross-cultural variation in feedback actionability, and (3) explore methods for refining or rewriting vague comments into actionable ones to support professional development more directly.

Acknowledgments

We are deeply grateful to Francisco Carballo Santiago from Rising Academies for providing us with this dataset and assisting in our initial exploration. We also extend our thanks to Sanne Smith and Michael Hardy for their invaluable insights into our research questions, figures, tables, and early drafts of the paper. Finally, we thank our peers for their participation in peer review sessions, which significantly contributed to strengthening our study.

Ethics Statement

Potential Misuse: Our model could be misused in high-stakes evaluations, leading to the automated assessment of observer performance without appropriate human oversight. We explicitly discourage such use and advocate for responsible deployment. **Privacy Considerations:** Implementing this technology would require strict privacy protocols to protect teacher identities. Observation data should be de-identified before being fed into the model to ensure confidentiality.

References

C. Adelman and R. Walker. 1975. *A Guide to Classroom Observation (1st ed.)*. Routledge.

Joseph P Allen, Robert C Pianta, Anne Gregory, Amori Yee Mikami, and Janetta Lun. 2011. An interaction-based approach to enhancing secondary school instruction and student achievement. *Science*, 333(6045):1034–1037.

J. Archer, S. Cantrell, S. L. Holtzman, J. N. Joe, C. M. Tocci, and J. Wood. 2016. *Better feedback for better teaching: A practical guide to improving classroom observations*. Jossey-Bass, a Wiley Brand.

S. Benslimane, T. Papastergiou, J. Azé, S. Bringay, M. Servajean, and C. Mollevi. 2024. [A shap-based controversy analysis through communities on twitter](#). *World Wide Web*, 27(5).

M. D. Cannon and R. Witherspoon. 2005. [Actionable feedback: Unlocking the power of learning and performance improvement](#). *Academy of Management Perspectives*, 19(2):120–134.

Linda Darling-Hammond, Maria E Hyler, and Madelyn Gardner. 2017. Effective teacher professional development. *Learning policy institute*.

James R. A. Davenport and Robert DeLine. 2014. [The readability of tweets and their geographic correlation with education](#). ArXiv preprint arXiv:1401.6058.

D. Demszky, J. Liu, Z. Mancenido, J. Cohen, H. Hill, D. Jurafsky, and T. 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)*.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). *arXiv preprint arXiv:1810.04805*.

R. Flesch. 1948. [A new readability yardstick](#). *Journal of Applied Psychology*, 32(3):221–233.

M. Hardy. 2025. “all that glitters”: Approaches to evaluations with unreliable model and human annotations.

Heather C Hill, Charalambos Y Charalambous, and Matthew A Kraft. 2012. When rater reliability is not enough. *Educ. Res.*, 41(2):56–64.

T. J. Kane and D. O. Staiger. 2012. [Gathering feedback for teaching: Combining high-quality observations with student surveys and achievement gains](#).

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. *Psychol. Bull.*, 119(2):254–284.

Matthew A Kraft, David Blazar, and Dylan Hogan. 2018. The effect of teacher coaching on instruction and achievement: A meta-analysis of the causal evidence. *Rev. Educ. Res.*, 88(4):547–588.

- V. Lazarev and D. Newman. 2015. [How teacher evaluation is affected by class characteristics: Are observations biased?](#) In *Paper presented at the Annual Meeting of AEFPP, Washington, DC*.
- S. P. Leeman-Munk, E. N. Wiebe, and J. C. Lester. 2014. [Assessing elementary students' science competency with text analytics](#). In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, pages 143–147.
- 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.
- Scott M. Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions](#). *CoRR*, abs/1705.07874.
- N. Madnani, A. Loukina, A. von Davier, J. Burstein, and A. 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.
- J. Millman and L. Darling-Hammond. 1990. *The new handbook of teacher evaluation*. SAGE Publications, Ltd.
- R. T. Putnam and H. Borko. 2000. [What do new views of knowledge and thinking have to say about research on teacher learning?](#) *Educational Researcher*, 29(1):4–15.
- M. Shah and A. Pabel. 2019. [Making the student voice count: using qualitative student feedback to enhance the student experience](#). *Journal of Applied Research in Higher Education*, 12(2):194–209.
- T. Shaik, X. Tao, Y. Li, C. Dann, J. McDonald, P. Redmond, and L. Galligan. 2022. [A review of the trends and challenges in adopting natural language processing methods for education feedback analysis](#). *IEEE Access: Practical Innovations, Open Solutions*, 10:56720–56739.
- Matthew P Steinberg and Lauren Sartain. 2015. [Does teacher evaluation improve school performance? experimental evidence from chicago's excellence in teaching project](#). *Educ. Finance Policy*, 10(4):535–572.
- Abhijit Suresh, Tamara Sumner, Jennifer Jacobs, Bill Foland, and Wayne Ward. 2019. [Automating analysis and feedback to improve mathematics teachers' classroom discourse](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9721–9728.
- Marieke Thurlings, Marjan Vermeulen, Theo Bastiaens, and Sjef Stijnen. 2013. [Understanding feedback: A learning theory perspective](#). *Educ. Res. Rev.*, 9:1–15.
- Deliang Wang, Dapeng Shan, Yaqian Zheng, Kai Guo, Gaowei Chen, and Yu Lu. 2023. [Can chatgpt detect student talk moves in classroom discourse? a preliminary comparison with bert](#). In *Proceedings of the 16th International Conference on Educational Data Mining*, pages 515–519, Bengaluru, India. International Educational Data Mining Society.
- N. E. Winstone, R. A. Nash, M. Parker, and J. Rowntree. 2016. [Supporting learners' agentic engagement with feedback: A systematic review and a taxonomy of recipience processes](#). *Educational Psychologist*, 52(1):17–37.
- T. Wragg. 2011. *An Introduction to Classroom Observation (Classic Edition)*. Routledge.
- H. Zhang and D. Litman. 2021. [Essay quality signals as weak supervision for source-based essay scoring](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–96. Association for Computational Linguistics.

A Appendix

Feedback Type	Rubric
Actionable	<ul style="list-style-type: none">• Provides clear and specific suggestions for improvement (Archer et al., 2016; Cannon and Witherspoon, 2005)• Offers explicit guidance on:<ul style="list-style-type: none">– <i>What</i> the teacher should do next– <i>How</i> the suggested change can be implemented• Focuses on observable behaviors rather than personality traits (Archer et al., 2016)• Establishes clear connections between observed behaviors and suggested improvements (Cannon and Witherspoon, 2005)• Provides balanced positive and constructive components (Cannon and Witherspoon, 2005)• May or may not contain indicative phrases (e.g., “<i>even better if</i>,” “<i>could have</i>”); presence of such phrases is not required• Includes concrete examples of alternative approaches
Vague	<ul style="list-style-type: none">• Lacks concrete or specific suggestions for improvement (Archer et al., 2016)• Fails to provide clear guidance on implementation steps (Kraft et al., 2018)• May focus on general impressions rather than specific teaching behaviors (Archer et al., 2016)• Lacks explicit connection between observation and suggested change (Cannon and Witherspoon, 2005)• Provides limited or no concrete examples of alternative approaches• May use evaluative language without actionable direction (Allen et al., 2011)• May include general phrases (e.g., “<i>even better if</i>,” “<i>could have</i>”), but their presence does not ensure clarity; feedback is considered vague if the intended action or direction remains ambiguous or insufficiently specified

EduCSW: Building a Mandarin-English Code-Switched Generation Pipeline for Computer Science Learning

Ruishi Chen*
Stanford University
ruishich@stanford.edu

Yiling Zhao*
Stanford University
ylzhao@stanford.edu

Abstract

This paper presents *EduCSW*, a novel pipeline for generating Mandarin-English code-switched text to support AI-powered educational tools that adapt computer science instruction to learners’ language proficiency through mixed-language delivery. To address the scarcity of code-mixed datasets, we propose an encoder-decoder architecture that generates natural code-switched text using only minimal existing code-mixed examples and parallel corpora. Evaluated on a corpus curated for computer science education, human annotators rated 60–64% of our model’s outputs as natural, significantly outperforming both a baseline fine-tuned neural machine translation (NMT) model (22–24%) and the DeepSeek-R1 model (34–44%). The generated text achieves a Code-Mixing Index (CMI) of 25.28%, aligning with patterns observed in spontaneous Mandarin-English code-switching. Designed to be generalizable across language pairs and domains, this pipeline lays the groundwork for generating training data to support the development of educational tools with dynamic code-switching capabilities.

1 Introduction

Code-switching (CSW), the practice of alternating between two or more languages within an utterance or conversation, is prevalent across diverse settings and multilingual communities (Gardner-Chloros, 2009; Poplack, 2001). Prior research has shown that CSW enables language learners to express their perspectives, convey culturally specific ideas, and build social relationships (Bhatia and Ritchie, 2006). In educational contexts, CSW has been found to enhance student engagement and help teachers clarify complex concepts, making it a valuable pedagogical strategy in multilingual classrooms (Sakaria and Priyana, 2018).

Despite its demonstrated benefits, support for code-mixing in educational tools remains limited (Yong et al., 2023). This gap is particularly pronounced in computer science education, where much of the terminology originates in English (Foote, 2023). For English-as-a-second-language learners, especially Chinese students pursuing studies abroad, this creates a dual challenge: mastering both general English and domain-specific vocabulary needed to comprehend technical content and participate in academic discourse.

Recent advances in large language models (LLMs) and speech recognition have shown potential in addressing challenges in CSW research (Giattino et al., 2023). While efforts have been made in speech translation for code-switched recognition (Alastruey et al., 2023; Wang and Li, 2023) and decoding code-mixed text (Sterner and Teufel, 2023), progress remains hindered by several issues. Studies reveal that even advanced multilingual LLMs struggle to produce natural code-switched text, often defaulting to full translation instead of authentically mixing languages (Kaji and Shah, 2023). This limitation stems from training predominantly on monolingual datasets, rather than natural code-switched corpora (Zhang et al., 2023). Moreover, challenges such as limited availability of code-mixed textual data, grammatical complexity, and domain mismatch further restrict development (Hussein et al., 2023). In particular, the lack of publicly available Mandarin-English code-mixed datasets impedes the creation of LLM-powered educational tools that support CSW.

To address these challenges, our work makes two primary contributions to CSW research. First, we introduce a generalizable pipeline for code-mixed data generation that can be adapted to various language pairs and subject domains. Second, we demonstrate its effectiveness by curating a domain-specific dataset for computer science education, focused on Chinese students studying at English-

*Equal contribution.

medium universities. This implementation lays the foundation for developing AI-powered tutoring systems that dynamically incorporate code-switching to support learners' acquisition of English technical language.

2 Related Work

2.1 Code-switching Background

CSW research has a relatively long history, dating back to the early 1900s (Winata et al., 2023). As the field evolved, advancements in machine learning, particularly deep learning (Gupta et al., 2020), have enabled more effective methods for both curating CSW datasets and managing various CSW tasks (Yong et al., 2023). However, the field faces considerable challenges, notably the scarcity of publicly available CSW datasets (Pratapa et al., 2018; Winata et al., 2023). Additionally, formal records of CSW texts are limited, and a significant portion of existing data is private or restricted, making it difficult to evaluate models and expand CSW research into new languages and contexts. These limitations hinder the diversification of CSW tasks and slow progress in generating comprehensive multilingual datasets.

Studies have attempted to identify key linguistic features in CSW with the goal of generating synthetic CSW data to address various challenges. Prior research has highlighted the Equivalence Constraint theory, which suggests that CSW occurs when the grammatical rules of all involved languages are maintained in a given sentence (Winata et al., 2023; Deuchar, 2020). Other works have identified the Matrix Language Frame (MLF) model (Myers-Scotton, 2001), which posits the existence of a dominant "matrix" language providing the grammatical structure, while the "embedded" language contributes additional content. This model has been proven successful in preserving syntactic features and grammatical structures from the matrix language (Callahan, 2002; Deuchar, 2006; Rahimi and Dabaghi, 2013).

2.2 Code-switching in Education

Most of the research has focused on the use of CSW in bilingual-classroom settings, suggesting its potential in enhancing instruction across subjects and improving classroom engagement. Sakaria and Priyana have identified that the use of code-switched instructional language can increase the efficiency in delivering lesson objectives and pro-

vide a theoretical framework (Sakaria and Priyana, 2018). Meanwhile, Milroy et al. also proposed that the use of code-switching can help teachers shape classroom culture, fostering different teacher-student relationships in the classroom environment (Milroy and Muysken, 1995). For instance, when teachers use the students' first language in instruction, it creates a playful and less formal environment. When the teachers switch back to the language the students are learning in that session, they reassert their authority and thus redefine the situation to be more formal.

These studies reveal the multifaceted benefits of code-switching, providing greater motivation for us to empower education by addressing the data scarcity issues in this field.

2.3 Algorithmic Solutions to Generating Code-mixed Data

Prior studies have adopted various linguistic theories and advanced language models to address the challenges in generating code-mixed texts, each reflecting distinct emphases.

For instance, Pratapa et al. (Pratapa et al., 2018) employed equivalence constraint theory, focusing on syntactic compatibility at switch points where language structures coincide. They used projections of parallel monolingual sentences to generate grammatically valid code-mixed sentences. Gupta et al. (Gupta et al., 2020) applied the Matrix Language Frame (MLF) theory, emphasizing the role of a dominant language in structuring code-mixed sentences. Tarunesh et al. (Tarunesh et al., 2021) utilized the Embedded Matrix Theory (EMT), a variation of MLF, applying clause substitution methods to create code-mixed text that satisfies Hindi-English grammatical structures.

For code-mixed data evaluation, prior scholars have proved the efficiency in various methods when assessing the naturalness of code-mixed data. Pratapa et al. (Pratapa et al., 2018) primarily assessed perplexity reductions on real code-mixed test sets using their RNN language model, which was trained on various combinations of monolingual, synthetic, and real code-mixed data. In contrast, Gupta et al. (Gupta et al., 2020) employed more direct metrics such as BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2001), ROUGE (Lin and Hovy, 2002), and METEOR (Lavie and Agarwal, 2007), along with human evaluation to assess the syntactic and semantic correctness, and natural-

ness of the generated code-mixed sentences. These diverse approaches guided our team in developing appropriate validation methods for our generated synthetic texts.

3 Method

This section outlines the data source utilized for this project and then presents the generalizable code-switched generation pipeline (see Appendix A.1 for more details). The repository is publicly available¹.

3.1 Data

A representative Mandarin-English code-mixed dataset for computer science education must possess two essential characteristics.

First, the dataset should accurately represent instructional language and encompass educational materials in computer science. This provides a domain-specific context that can shape the generated code-mixed corpus to offer effective and specific support for computer science instruction.

Second, the corpus should align with how bilingual users naturally develop and use code-mixed content in educational and daily contexts. This naturalness is critical as it ensures the code-mixed text authentically reflects the language patterns observed in real-world bilingual educational settings.

Accordingly, we utilize two datasets that satisfy the above criteria in our project: a Mandarin dataset capturing domain-specific computer science instructional content, and a second dataset reflecting spontaneous code-mixing patterns in Mandarin-English speakers’ daily communication.

3.1.1 Computer Science Instruction Dataset

This study incorporates the Hugging Face dataset `2imi9/llama2_7B_data_10G`, which contains ten gigabytes of bilingual text data sourced from Hugging Face and the Chinese Software Developer Network (CSDN), covering technical instructions in computer science. The dataset was carefully curated to support the development of AI-powered educational tools for personalized learning in Shenzhen University’s *University Computer* course. It includes a column of conceptual questions (“Instruction”) and serves as the primary input for generating code-mixed representations in this study. Due to computational constraints, we used a subset of this dataset containing 744 technical instructions for computer science (file name:

¹<https://github.com/RuishiCh-git/EduCSW/tree/main>

`data_alpaca_standardized_data`), which captures common questions and explanations of key computer science terminology.

Instruction

什么是计算机?
(what is computer?)

如何解释人工智能在不同领域（如医疗、金融、教育）中的应用及其带来的影响?
(How to explain the application of artificial intelligence in various fields (such as healthcare, finance, education) and the impacts it brings?)

Table 1: Sample Data Entries (The parentheses contain translations, not part of the data.)

3.1.2 Spontaneous Mandarin-English Code-Mixed Dataset

To train our model on real CSW data, we incorporated the speech transcription dataset `CAiRE/ASCEND` (Lovenia et al., 2022)² into our pipeline. We filtered the original training dataset to retain only code-mixed text, resulting in 2,739 code-mixed utterances used in this study. This subset provides a Mandarin-English code-switching corpus that reflects authentic code-switched language patterns in bilingual speakers’ habits. Sample code-mixed transcriptions from this dataset are shown in Table 2.

Code-mixed Data in ASCEND

快快要期末考试了他可能觉得非常stress 非常nervous
(It’s getting close to the final exam. He might feel very stressed and nervous.)

放在剧情上的focus on the script but not the action but not the 特效
(Focus on the script rather than the action or the special effects.)

Table 2: Sample Mandarin-English code-mixed data (The parentheses contain translations, not part of the data.)

3.2 Pipeline

Overall, the code-mixed text generation included three key stages: the *Preparation* stage, the *Code-mixed generation* stage, and the *Evaluation* stage.

²ASCEND (A Spontaneous Chinese-English Dataset) is a spontaneous multi-turn conversational dialogue recorded in Hong Kong.

3.2.1 Preparation Stage

The preparation stage includes three major steps: parallel corpus preparation, language alignment, and hard-coded code-mixed data generation.

Firstly, the machine translation model Helsinki-NLP/opus-mt-zh-en (Tiedemann and Thottingal, 2020)³ was used to obtain the corresponding English corpus for the Mandarin computer science instruction dataset.

Secondly, the word aligner awesome-align (Dou and Neubig, 2021) was employed to create an alignment matrix for the parallel corpus. The input consists of parallel sentences separated by “|||”, and the output is in the i-j Pharaoh format. A pair i-j indicates that the i-th word (zero-indexed) of the source sentence (Mandarin) is aligned to the j-th word of the target sentence (English). An example is shown in Appendix A.2.

Thirdly, a BERT-based Named Entity model and the *jieba* module were used to tokenize and extract linguistic features and tags from English and Mandarin corpus. The Matrix Language Frame (MLF)⁴ was followed to generate code-switched text (Myers-Scotton, 2002). In this study, Mandarin serves as the matrix language dominating the sentence, while English is the embedded language inserted into the sentence. Named entities (NE), noun phrases (NP), and adjectives (ADJ) in the English sentence were identified as candidate words/phrases for insertion into the Chinese sentence.

For each candidate word/phrase, the language switch-point was determined based on the POS tag and position in the sentence. Insertion probabilities were set to 20% - 30% to achieve an observed code-mixing index (CMI) consistent with natural code-mixed utterances, based on prior literature (Li et al., 2012). If a switch was decided, the English word/phrase was inserted into the corresponding position in the Mandarin sentence. The resulting dataset was used as the first round of “hard-coded” CSW data.

³This model was developed by the Language Technology Research Group at the University of Helsinki and is designed to translate from Chinese (source language) to English (target language).

⁴MLF, proposed by Myers-Scotton, introduced the “asymmetry principle,” where the language providing the morphosyntactic structure is the “matrix language,” while the “embedded language” contributes elements that switch into the matrix language (Myers-Scotton, 2002)

3.2.2 Code-Mixed Generation Model

We experimented with three approaches for code-mixed data generation. The first approach extends a neural machine translation (NMT) model, serving as a baseline for comparison. The second uses the DeepSeek-R1 model to establish a benchmark performance. The third, and our primary contribution, is a custom encoder-decoder architecture designed specifically for generating natural code-switched text.

Approach 1: Fine-tuning NMT For the Neural Machine Translation (NMT) fine-tuning approach, we used the Helsinki-NLP/opus-mt-zh-en model⁵, originally designed for Chinese-to-English translation. This model served as our baseline for generating code-switched text. It consists of approximately 77 million parameters and features an architecture with 6 encoder layers and 6 decoder layers, offering a robust foundation for capturing the complexities of both Chinese and English, as well as the nuances of code-switching patterns.

To adapt the model to our specific task, we used two primary sources of training data: the first round of “hard-coded” CSW data and code-mixed transcriptions from the ASCEND dataset (Lovenia et al., 2022). This combination was selected to balance domain-specific accuracy with the naturalness of authentic code-switching.

The model was fine-tuned over 3 epochs, using a learning rate of $2e-5$ and a batch size of 16 per device. These parameters were chosen to ensure adequate adaptation to the code-switching task while minimizing the risk of overfitting. The fine-tuning concluded with a final training loss of 0.709, indicating a solid trade-off between specialization and generalization. The resulting model is publicly available⁶.

Approach 2: DeepSeek-R1 Benchmark To benchmark our code-mixed text generation pipeline against a strong pre-trained baseline, we utilized Distilled DeepSeek-R1 7B, based on Qwen—a large language model trained on both Chinese and English corpora (DeepSeek-AI, 2025). DeepSeek has demonstrated remarkable performance across a range of Chinese natural language understanding and generation tasks, making it a valuable reference point for evaluating code-switching capabilities.

⁵<https://huggingface.co/Helsinki-NLP/opus-mt-zh-en>

⁶<https://huggingface.co/y131/code-mixed-cs-edu-model>

ties. Although DeepSeek is not explicitly trained for code-switching, it offers insight into how well general-purpose, state-of-the-art language models can handle code-mixing in the absence of domain-specific supervision. As such, this benchmark serves as a reasonable point of comparison for our customized generation pipeline.

We adopted a few-shot prompting strategy to guide DeepSeek toward producing Mandarin-English code-switched output. Each prompt included two illustrative examples demonstrating how to naturally integrate English computer science terminology into Mandarin instructional sentences. These examples showcased both noun phrase-level and verb-level switches—patterns commonly observed in bilingual academic discourse. The complete prompt is provided in Appendix A.5. This prompt was applied to all 744 Mandarin instructional sentences in our dataset. Model outputs were collected without any post-processing to preserve their authenticity for subsequent evaluation.

Approach 3: Encoder-Decoder Architecture

For the encoder-decoder architecture model, the rationale is to use the encoder to provide context while the decoder generates target sequences with a copy mechanism, improving model performance through a combination of translation and copying from input text.

We first leverage **transfer learning** to initiate our code-mixed generation model. This approach aims to reduce the required training data for code-mixed generation while ensuring high-quality bilingual representations essential for natural code-switching data generation. Specifically, we fine-tune the neural machine translation model [Helsinki-NLP/opus-mt-zh-en](#) on our curated parallel corpus of computer science educational content and dialogue. The fine-tuning process enables the model to capture language-specific features, including domain-specific terminology and language patterns unique to computer science education in both Mandarin and English, as well as cross-lingual mappings, such as semantic equivalences and contextual relationships between the language pairs.

The weights learned during this fine-tuning phase provide monolingual understanding and capture cross-lingual feature characteristics. The next step is to use an **encoder-decoder architecture** that builds on the fine-tuned weights to integrate additional components extracted from the prelimi-

nary code-mixed dataset to build the code-mixed text generation model.

The **encoder**, built on the transformer layers of the MarianMTModel, processes the sequences of tokens in Chinese texts to produce hidden states that capture sequential dependencies and generate contextual representations for the sentences. These representations are then received by the attention mechanism in the decoder, allowing the model to have more focused access to relevant source information. This enables the preservation of both language-specific features and cross-lingual relationships.

Subsequently, the **decoder** uses a processing mechanism to adopt a standard decoder path for translation logits and a dedicated gate mechanism for copy probability calculation. With the attention mechanism, the encoder’s representations are processed to produce hidden states, which inform both generation and copying decisions. When copying from the input texts is decided, the model computes copy probabilities for the input tokens. Subsequently, the model expands input tokens to align with the target sequence length and then maps the tokens into the known vocabulary space using scatter operations, locating the vocabulary tokens in the input text. Such a mechanism is important to preserve technical terminology for conversational corpus related to Computer Science, where many words tend to co-occur for domain-specific meanings. For example, with the term ‘neural network,’ the model can directly copy these tokens rather than regenerate words for “network” or “neural” to maintain precise technical accuracy.

With the encoder-decoder architecture built, we optimize our operation with a specialized loss function that combines loss with a mixing ratio penalty. In particular, we incorporate a Code-Mixing Loss function to calculate the ratio of Chinese to English tokens and penalize the outputs that deviate from a ratio of 0.5 (set for a minimal mixing ratio). This approach preserves semantic accuracy within the code-mixed dataset while encouraging the model to learn from the trained dataset and generate balanced code-mixing data.

During training, the model processes both the hard-coded CSW data and the transcriptions from the ASCEND dataset (Lovenia et al., 2022). The training setup uses parallel data: the original Mandarin text serves as input, while the corresponding code-mixed versions (both hard-coded and AS-

CEND transcripts) serve as the target outputs. The generation strategy employs beam search with a beam width of 5, meaning it maintains the top 5 most probable sequences at each decoding step. Then, the model uses a 2-gram prevention strategy to prevent two consecutive tokens from appearing more than once in the generated sequence. These parameters were chosen to maintain output diversity and technical accuracy while preventing common generation issues like repetitive text.

4 Results and Evaluation

4.1 Description of Generated CSW Data

The 744 Mandarin text entries from the [2imi9/llama2_7B_data_10G](#) dataset were used as input for all three of our code-mixed generation models: the fine-tuned NMT approach, DeepSeek-R1, and the encoder-decoder architecture. This parallel processing enabled the generation of three distinct sets of code-switched (CSW) data, facilitating a comparative analysis across methods.

The generated CSW text preserves the educational content and structure of the original Mandarin entries while incorporating English elements in a way that reflects natural code-switching patterns commonly observed in bilingual educational contexts.

4.2 Evaluation

4.2.1 Code-Mixing Index

The Code-Mixing Index (CMI) (Das and Gambäck, 2014) is a widely used metric for measuring the complexity of code-mixed text (Srivastava and Singh, 2021). It quantifies the fraction of tokens or words that differ from the matrix language⁷. In our study, we calculated the sentence-level CMI⁸ by dividing the number of English tokens by the total word count in each CSW sentence.

The overall CMI for each generated CSW dataset was computed as the average of all sentence-level CMIs within that dataset. As presented in Table 3, the CMI for the hard-coded first round of generated CSW data is 26.98%. The CMIs for the NMT fine-tuning, DeepSeek-R1, and encoder-decoder approaches are 23.05%, 9.95%, and 25.28%, respectively.

Notably, the CMIs for most of our generated CSW datasets fall within the 20% to 30% range,

⁷<https://tech.skit.ai/Code-Mixing-Metrics/>

⁸See Appendix A.4 for CMI formula.

Method	Matrix Lang.	CMI
Hard Code	/	26.89%
NMT Fine-tuning	Chinese	23.05%
Deepseek R1	Chinese	9.95%
Encoder/Decoder	Chinese	25.28%

Table 3: CMIs for Different Methods

which aligns with values observed in spontaneous Chinese-English code-switching utterances from prior studies (see Appendix A.3). This suggests that our generated CSW data—excluding the output from DeepSeek-R1—closely mirrors natural code-mixing patterns, reinforcing the credibility and authenticity of the synthetic text. The substantially lower CMI of DeepSeek-R1 (9.95%) indicates limited code-switching behavior, which may reduce its effectiveness for simulating natural bilingual communication.

4.2.2 Human Labeling

To comprehensively evaluate the quality of the generated data, we recruited two bilingual annotators to label the CSW outputs from the NMT model, DeepSeek-R1, and the encoder-decoder framework. Both annotators were proficient in Mandarin-English code-mixing and had familiarity with domain-specific computer science terminology. They were instructed to rate the naturalness of each sentence using a standardized 3-point Likert scale (Joshi et al., 2015): unnatural (1), acceptable (2), and natural (3). If a sentence contained nonsensical segments that severely disrupted its meaning, annotators could label it as “wrong,” in which case it was excluded from the naturalness evaluation.

Each annotator labeled 50 entries from each of the three models. These entries were derived from 50 randomly sampled Chinese input sentences. To assess annotation consistency, we calculated inter-rater reliability using Cohen’s kappa coefficient (Blackman and Koval, 2000). The resulting κ values were 0.6739 for the fine-tuned NMT model, 0.6793 for the encoder-decoder model, and 0.7622 for DeepSeek-R1—indicating moderate to strong agreement between annotators.

We then compared the performance of the three models in generating natural CSW outputs. Table 4 presents the percentage of outputs rated as natural. The encoder-decoder approach significantly outperformed both the fine-tuned NMT and DeepSeek-R1

models. Annotators consistently rated a higher proportion of encoder-decoder outputs as natural (64% and 60%) compared to those from the NMT model (22% and 24%) and DeepSeek-R1 (34% and 44%).

Labeler	Fine-tuned NMT	DeepSeek R1-Distill	Encoder Decoder
1	22%	44%	64%
2	24%	34%	60%

Table 4: Comparison of Natural Output Percentages by Annotators

Sentences annotated as natural typically demonstrated preservation of the grammatical rules of the matrix language (Mandarin) and exhibited switches at technical terms and language-sensitive words (words more commonly used in English). For instance, in the example shown in Table 5, the technical terms “自然语言处理” and “机器学习” in the input Mandarin sentence were switched to English expressions “language processing” and “machine learning” respectively, and the resulting sentence was labeled as natural.

Conversely, sentences labeled as unnatural often disobeyed Mandarin grammar and displayed issues such as incomplete semantic segments, mistranslations, or unbalanced proportions of Mandarin and English segments. Examples of such cases are also provided in Table 5.

4.2.3 Qualitative Evaluation

To further assess the quality of the generated code-switched text, we conducted a qualitative evaluation of outputs from the fine-tuned NMT approach, DeepSeek-R1, and the encoder-decoder framework. This analysis revealed clear differences in code-switching quality among the three methods.

The encoder-decoder framework demonstrated a superior ability to generate natural and coherent code-switched text. As shown in Appendix A.6, its outputs exhibit several favorable characteristics. The code-switched segments primarily consist of noun phrases and computer science-related terms in English, reflecting authentic bilingual speech patterns. Language switch points appear more natural and intuitive, and grammatical structures in both languages are better preserved, resulting in higher overall linguistic quality.

In contrast, the fine-tuned NMT model showed notable limitations. As illustrated in Appendix A.6, its outputs often exhibit grammatical inconsistencies when transitioning between English and Chi-

Input	Output and Label	
在自然语言处理(NLP)中, 如何利用机器学习进行情感分析? 请描述其过程和应用场景。	在language processing(nlp)中, 如何利用machine learning进行emotional analysis? 请描述其过程and application场景。 (In NLP, how can machine learning be utilized for emotional analysis? Please describe the process and application scenarios.)	<i>Natural</i>
(In NLP, how to utilize machine learning for sentiment analysis? Please describe the process and application scenarios.)	在language processing(nlp)中, 如何用machine learning进analysis? please depletion of中的processing and application processing. (In NLP, how can we use machine learning for analysis? Please clarify the meanings of “processing” and “application processing” in “depletion of”.)	<i>Unnatural</i>

Table 5: Comparison of Natural and Unnatural Labels (The parentheses contain translations, not part of the data.)

nese. Additionally, it occasionally produces nonsensical or incoherent English terms (e.g., “convergence,” “protology,” “diploration”), leading to awkward transitions and a lower degree of naturalness compared to the encoder-decoder output.

DeepSeek-R1, a large language model trained on Chinese text, also displayed weaknesses in generating natural code-switching. Many outputs defaulted to full English translations rather than producing genuine code-switched language, resulting in a low Code-Mixing Index (CMI) and limited alignment with real-world bilingual discourse. While DeepSeek-R1 occasionally produced natural-sounding examples, its performance was inconsistent, and it was outperformed overall by the encoder-decoder framework.

In summary, the qualitative evaluation shows that the encoder-decoder model consistently generates more natural, coherent, and contextually appropriate code-switched text than both the fine-tuned NMT and DeepSeek-R1 approaches. Its outputs closely mimic authentic bilingual communication, particularly in technical domains, and exhibit a balanced and grammatically sound integration of

English terminology.

5 Discussion & Conclusion

In this study, we developed a comprehensive, effective, and reusable pipeline for generating synthetic code-mixed data, with the goal of supporting the training of human-centered tutoring large language models (LLMs) and chatbots that communicate using a code-mixed approach. This work is motivated by the pedagogical value of code-mixed instruction for bilingual learners adapting to second-language environments. At the same time, existing publicly available LLMs show limited proficiency in handling code-switching, often focusing narrowly on topic-related nouns (Yong et al., 2023). In addition to proposing a general pipeline, we apply it to create a Mandarin-English code-mixed dataset specifically curated for computer science education.

We accomplished two key objectives:

First, we successfully developed a generalizable pipeline for generating code-mixed data across language pairs (with English as one of the languages). The pipeline consists of three main steps: (1) generating preliminary synthetic code-switched data using the Matrix Language Frame (MLF) theory and BERT-based Named Entity Recognition to prepare the non-English monolingual data; (2) passing the text through an encoder-decoder architecture initialized with weights from an NMT model fine-tuned on a parallel corpus, and training it using both the synthetically generated and real code-mixed data; and (3) iteratively annotating and retraining to enhance the naturalness of the generated outputs.

To adapt the pipeline for other language pairs, users only need to modify two components: (1) the Matrix Language Frame to match the grammatical structure of the target language, and (2) the code-switched speech transcription dataset, which is often more readily available than textual resources. With these changes, users can input their own monolingual data and generate suitable code-mixed datasets for downstream tasks.

Second, we successfully curated a domain-specific code-mixed dataset for computer science education that can support downstream training of LLMs or chatbots. This dataset was validated through three evaluation methods: the Code-Mixing Index (CMI), human ratings, and qualitative analysis. Across all measures, our encoder-decoder architecture outperformed both the state-

of-the-art DeepSeek LLM and a traditional fine-tuned neural machine translation model in generating natural code-switched text.

We offer two suggestions based on our findings. First, given the success of our pipeline in the computer science domain, we recommend applying this approach in other STEM fields where technical vocabulary creates challenges for bilingual learners (Bhatia and Ritchie, 2006). Second, we encourage the development of interactive tutoring systems and LLM-powered chatbots using our curated dataset and pipeline, with the capacity to dynamically adjust the degree of code-mixing based on learners' language proficiency. As supported by prior work (Milroy and Muysken, 1995), flexible language use in educational settings can greatly enhance learner engagement and comprehension.

5.1 Limitation and Future Work

We identify two limitations in this study.

First, although we use transcriptions from a code-mixed audio dataset to fine-tune the naturalness of our model's outputs, the ASCEND training dataset occasionally contains spelling errors, incomplete sentences, and casual conversational utterances. These issues may affect the quality of the generated code-mixed text. Future researchers may improve results by further cleaning and curating a high-quality subset of the transcription data or by sourcing data from more professional or domain-relevant contexts.

Second, due to the nature of the fine-tuned NMT model being primarily designed for translation tasks, it occasionally produces fully translated English output. This indicates that the model's control over the language mixing ratio is not yet optimal. Future work could explore increasing the number of training iterations and implementing a feedback loop to monitor and dynamically adjust the language balance during generation, thereby enhancing the consistency and naturalness of code-switching.

6 Ethical Consideration

Our primary data source, the Mandarin instructional dataset for computer science learning, is open-sourced on Hugging Face and explicitly designed to improve AI model performance in educational settings. Our use aligns with this stated purpose, and we have properly cited the source. Similarly, the ASCEND dataset, used for code-

mixing patterns, is open-sourced and appropriately cited. For annotation, we engaged voluntary participants, ensuring ethical practices in data labeling.

The primary application of our work is developing AI-powered tutoring chat bots for personalized computer science learning, bridging the gap for bilingual learners transitioning from Mandarin to English-language education. We acknowledge the need to preserve language integrity, respect cultural nuances, and avoid exacerbating educational disparities.

References

- Belen Alastruey, Matthias Sperber, Christian Gollan, Dominic Telaar, Tim Ng, and Aashish Agarwal. 2023. [Towards real-world streaming speech translation for code-switched speech](#). *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, page 14–22.
- Tej K. Bhatia and William C. Ritchie. 2006. *The Handbook of Bilingualism*. Blackwell Pub.
- Nicole J-M Blackman and John J Koval. 2000. Interval estimation for cohen’s kappa as a measure of agreement. *Statistics in medicine*, 19(5):723–741.
- Laura Callahan. 2002. [The matrix language frame model and spanish/english codeswitching in fiction](#). *Language Communication*, 22(1):1–16.
- Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#).
- Margaret Deuchar. 2006. [Welsh-english code-switching and the matrix language frame model](#). *Lingua*, 116(11):1986–2011. Celtic Linguistics.
- Margaret Deuchar. 2020. [Code-switching in linguistics: A position paper](#). *Languages*, 5(2):22.
- Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.
- Keith D. Foote. 2023. [A brief history of large language models](#).
- Penelope Gardner-Chloros. 2009. *Code-switching*. Cambridge University Press.
- Charlie Giattino, Edouard Mathieu, Veronika Samborska, and Max Roser. 2023. Artificial intelligence. *Our World in Data*. <https://ourworldindata.org/artificial-intelligence>.
- Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. [A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning](#). *Findings of the Association for Computational Linguistics: EMNLP 2020*, page 2267–2280.
- Amir Hussein, Shammur Absar Chowdhury, Ahmed Abdelali, Najim Dehak, Ahmed Ali, and Sanjeev Khudanpur. 2023. [Textual data augmentation for arabic-english code-switching speech recognition](#). *2022 IEEE Spoken Language Technology Workshop (SLT)*.
- Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. 2015. Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396–403.
- Arshad Kaji and Manan Shah. 2023. Contextual code switching for machine translation using language models. *arXiv preprint arXiv:2312.13179*.
- Alon Lavie and Abhaya Agarwal. 2007. [Meteor](#). *Proceedings of the Second Workshop on Statistical Machine Translation - StatMT '07*, page 228–231.
- Ying Li, Yue Yu, and Pascale Fung. 2012. A mandarin-english code-switching corpus. In *LREC*, pages 2515–2519.
- Chin-Yew Lin and Eduard Hovy. 2002. [Manual and automatic evaluation of summaries](#). *Proceedings of the ACL-02 Workshop on Automatic Summarization -*, 4:45–51.
- Holy Lovenia, Samuel Cahyawijaya, Genta Indra Winata, Peng Xu, Xu Yan, Zihan Liu, Rita Frieske, Tiezheng Yu, Wenliang Dai, Elham J Barezi, et al. 2022. Ascend: A spontaneous chinese-english dataset for code-switching in multi-turn conversation. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*.
- Dau-Cheng Lyu, Tien Ping Tan, Engsiong Chng, and Haizhou Li. 2010. Seame: a mandarin-english code-switching speech corpus in south-east asia. In *Inter-speech*, volume 10, pages 1986–1989.
- L. Milroy and P.C. Muysken. 1995. *One speaker, two languages: Cross-disciplinary perspectives on code-switching*. Cambridge University Press.
- Carol Myers-Scotton. 2001. The matrix language frame model: Development and responses. *Trends in Linguistics Studies and Monographs*, 126:23–58.
- Carol Myers-Scotton. 2002. *Contact linguistics: Bilingual encounters and grammatical outcomes*. Oxford University Press, USA.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. [Bleu](#). *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311.
- S. Poplack. 2001. [Code switching: Linguistic](#). *International Encyclopedia of the Social and Behavioral Sciences*, page 2062–2065.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. [Language modeling for code-mixing: The role of linguistic theory based synthetic data](#). *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 1543–1553.

Meisam Rahimi and Azizollah Dabaghi. 2013. [Persian–english codeswitching: A test of the matrix language frame \(mlf\) model](#). *System*, 41(2):322–351.

Severinus Sakaria and Joko Priyana. 2018. [Code-switching: A pedagogical strategy in bilingual classrooms](#). *American Journal of Educational Research*, 6(3):175–180.

Vivek Srivastava and Mayank Singh. 2021. [Challenges and limitations with the metrics measuring the complexity of code-mixed text](#). *arXiv preprint arXiv:2106.10123*.

Igor Sterner and Simone Teufel. 2023. [Tongueswitcher: Fine-grained identification of german-english code-switching](#). *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, page 1–13.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. [From machine translation to code-switching: Generating high-quality code-switched text](#). *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)*, page 3154–3169.

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 (EAMT)*, Lisbon, Portugal.

Qinyi Wang and Haizhou Li. 2023. [Text-derived language identity incorporation for end-to-end code-switching speech recognition](#). *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, page 33–42.

Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2023. [The decades progress on code-switching research in nlp: A systematic survey on trends and challenges](#). *Findings of the Association for Computational Linguistics: ACL 2023*, page 2936–2978.

Zheng-Xin Yong, Ruochen Zhang, Jessica Zosa Forde, Skyler Wang, Arjun Subramonian, Holy Lovenia, Samuel Cahyawijaya, Genta Indra Winata, Lintang Sutawika, Jan Christian Blaise Cruz, Yin Lin Tan, Long Phan, Rowena Garcia, Thamar Solorio, and Alham Fikri Aji. 2023. [Prompting multilingual large language models to generate code-mixed texts: The case of south east asian languages](#).

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Fikri Aji. 2023. [Multilingual large language models are not \(yet\) code-switchers](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582, Singapore. Association for Computational Linguistics.

A Appendix

A.1 Pipeline Flowchart

Note: The pipeline flowchart shown on the next page ([Appendix: 1](#)) illustrates our overall approach.

A.2 Example of input and output for word alignment using awesome-align

Type	Content
Input	我 喜欢 吃苹果 (zh) I like to eat apples (en)
Output	0-0 1-1 2-3 2-4

A.3 Reference CMI values from literature

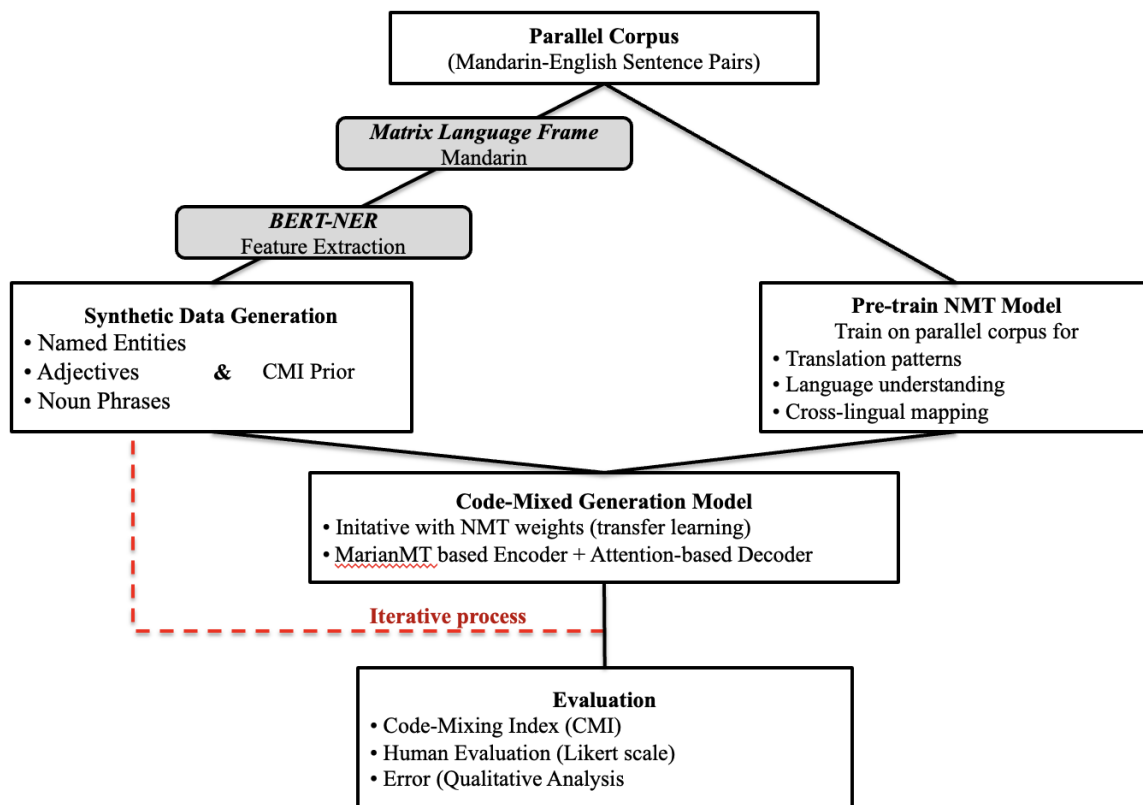
Reference	Matrix Language	CMI
(Li et al., 2012)	Chinese	21.15%
(Lyu et al., 2010)	Chinese	25%

A.4 CMI Formula

The CMI is calculated using the following formula:

$$CMI = 100 * \left(1 - \frac{\max(w_i)}{n - u} \right) \text{ if } n > u \quad (1)$$

where w_i is the number of words in language i (English), n is the total number of words, and u is the number of language-independent words.



Appendix A.1: Overall Pipeline. This flowchart shows the steps involved in the code-mixed generation model.

A.5 DeepSeek Prompt Template

```

Prompt
<system>
You are a helpful assistant. Your job is to
convert Mandarin computer science ques-
tions into Mandarin-English code-switched
sentences that sound natural to bilingual
learners.
Only output the sentence. Do not explain or
comment.
<user>
Input: 在深度学习中，如何训练卷积神
经网络？
Output: 在deep learning中，如何train con-
volutional neural network?
Input: 什么是计算机网络的拓扑结构？
Output: 什么是computer network 的 topol-
ogy 结构？
Now process the following:
Input: {text}
Output:
  
```

A.6 Comparison of Encoder-Decoder and NMT Generated Outputs (The parentheses contain translations, not part of the data.)

Output	Label
<i>Encoder-Decoder Generated</i>	
什么是data consistence? (What is data consistence?)	Natural
深度学习 中curly network (cnn) 如何实现image 分类and 对象检测? 请详细解释其working principles and technologies. (How does the curly network (CNN) in deep learning achieve image classification and object detection? Please elaborate on its working principles and technologies.)	Natural
<i>Fine-tuned NMT Generated</i>	
什么是data converence? (What is data converence?)	Wrong
深度学习中的blough network (CNN) 如何实现image diagation and operation processing? please process processing work chrinkings and key processings. (How does the blough network (CNN) in deep learning achieve image diffusion and operation processing? please process processing work chrinkings and key processings.)	Acceptable
<i>DeepSeek-RI-Distill</i>	
什么是data consistency? (What is data consistency?)	Natural
How to train a deep learning model to recognize cats and dogs in images?	Wrong

STAIR-AIG: Optimizing the Automated Item Generation Process through Human-AI Collaboration for Critical Thinking Assessment

Euigyum Kim¹, Seewoo Li², Salah Khalil³, and Hyo Jeong Shin^{1*}

¹Sogang University, Seoul, South Korea

²University of California, Los Angeles, USA

³MACAT International Ltd., United Kingdom

Abstract

The advent of artificial intelligence (AI) has marked a transformative era in educational measurement and evaluation, particularly in the development of assessment items. Large language models (LLMs) have emerged as promising tools for scalable automatic item generation (AIG), yet concerns remain about the validity of AI-generated items in various domains. To address this issue, we propose STAIR-AIG (*Systematic Tool for Assessment Item Review in Automatic Item Generation*), a human-in-the-loop framework that integrates expert judgment to optimize the quality of AIG items. To explore the functionality of the tool, AIG items were generated in the domain of critical thinking. Subsequently, the human expert and four OpenAI LLMs conducted a review of the AIG items. The results show that while the LLMs demonstrated high consistency in their rating of the AIG items, they exhibited a tendency towards leniency. In contrast, the human expert provided more variable and strict evaluations, identifying issues such as the irrelevance of the construct and cultural insensitivity. These findings highlight the viability of STAIR-AIG as a structured human-AI collaboration approach that facilitates rigorous item review, thus optimizing the quality of AIG items. Furthermore, STAIR-AIG enables iterative review processes and accumulates human feedback, facilitating the refinement of models and prompts. This, in turn, would establish a more reliable and comprehensive pipeline to improve AIG practices.

1 Introduction

Recent advances in natural language processing (NLP) and generative artificial intelligence (AI), particularly large language models (LLMs), have transformed educational measurement from relatively labor-intensive processes to more automated, scalable, and efficient approaches (Srinivasan, 2022; Wang et al., 2024).

Prominent examples include automated scoring (Latif and Zhai, 2024; Lee et al., 2024; Luchini et al., 2025) and automated feedback generation (Hahn et al., 2021; Chan et al., 2025), which substantially improve efficiency by reducing human labor while ensuring relatively valid and consistent outcomes.

Among these innovations, automatic item generation (AIG) has emerged as a particularly pertinent application of LLM for the rapid and effective development of assessment items (Gierl and Lai, 2013; Kurdi et al., 2020). Traditional AIG approaches generated new items by replacing different numbers or words in predefined models or templates, aiming to assess the same underlying construct. With the advent of LLMs, AIG has now entered a new phase, enabling educational researchers and practitioners to generate numerous items with minimal programming expertise. However, regardless of the AIG model used, the quality, appropriateness, and validity of AI-generated items still remain questionable. Consequently, the incorporation of quality assurance processes and human participation is deemed inevitable to ensure that AIG systems are generating content as intended (von Davier and Burstein, 2024).

In particular, it is important to ensure that the assessment items are aligned with target measurement constructs, as poorly defined constructs and superficially designed items can undermine the validity and reliability of the assessment (Liu et al., 2016). Consequently, a robust human-AI collaboration (HAIC) (Fragiadakis et al., 2025) is essential not only to leverage the scalability and efficiency of the AIG process, but also to ensure overall quality and safeguard the validity of AI-generated assessment items (Hao et al., 2024). Nevertheless, prior literature reveals a lack of empirical studies validating the appropriateness of AI-generated items for assessing cognitive skills within human-AI collaborative contexts.

*Corresponding author: hshinedu@sogang.ac.kr

To address this gap, the present study introduces **STAIR-AIG** (*Systematic Tool for Assessment Item Review in Automatic Item Generation*), an item review tool that supports systematic and efficient human review of AI-generated assessment items. We illustrate its potential as both a practical tool and a conceptual AIG framework by applying it to the domain of critical thinking (CT), a higher-order cognitive skill widely recognized as an essential 21st-century core competency (World Economic Forum, 2015). In complex cognitive domains, such as CT, the expert review by the human is particularly important in that defining the measurement structures and developing the assessment items are quite challenging (Shin et al., 2025).

By leveraging NLP techniques, our tool provides a comprehensive linguistic feature analysis of items. This empowers human reviewers to integrate their domain knowledge in a more effective way. Furthermore, the evaluations of AIG items by human experts are stored as data, so they continuously contribute to the improvement and refinement of the internal LLMs within the AIG pipeline. In contrast to conventional methods, which generally rely exclusively on human review as a final gatekeeping measure in a linear fashion, STAIR-AIG incorporates multiple structured touch-points for expert judgment at each stage. This facilitates continuous evaluation, targeted refinement of AI-generated elements, and ongoing enhancement of LLMs for AIG through structured human feedback and prompt optimization in a dynamic manner.

In the following, we illustrate the use of the STAIR-AIG tool as a human-in-the-loop AIG process. We review the relevant literature on AIG and the traditional item review process. Then, we present a case study that demonstrates the use of the STAIR-AIG tool in the CT domain. Subsequently, we compare the evaluations performed by a human expert with those generated by the LLM to identify discrepancies and examine the implications of their collaboration for enhancing the AIG process.

2 Related Works

2.1 Automatic Item Generation

With the growing interest in AIG to build reliable computer-based assessments by stably and efficiently feeding items into the item bank, the number of publications on AIG has recently increased (Kurdi et al., 2020). Before the advent of LLMs, the techniques of AIG studies were based on syntax

or templates that harness computational power to reduce human labor, such as employing grammar correction programs and developing templates to build software programs (Bejar, 1996, 2002; Singley and Bennett, 2002). In contrast, the recent rise of LLMs in the AI research field has enabled AIG researchers to generate items without extensive software engineering, while empowering item developers to effectively realize their nuanced intentions within the generation process (Attali et al., 2022; Bezirhan and von Davier, 2023).

In line with current research trends in AIG based on LLMs, this study utilizes CT items developed through a structured AIG procedure (Shin et al., 2025). This approach leverages prompt engineering techniques using LLM and is structured into three distinct modules—passage, question, and choices statements—to support systematic generation and monitoring. Within each module, detailed prompts are provided to the LLM to generate components of items intended to assess CT skills. The modules are executed sequentially to form a complete item, which is then finalized through expert review and revision. Psychometric analyses of the pilot-study data confirmed that the generated items were functioning as intended (Shin et al., 2025).

2.2 Assessment Item Review Procedure

Traditionally, the development and validation of assessment items have relied heavily on expert-driven review procedures to ensure validity, cognitive alignment, and fairness (Haladyna and Rodriguez, 2013). Guidelines from organizations such as the National Council on Measurement in Education (NCME) and the International Test Commission (ITC) emphasize the need for refinements guided by expert judgment to avoid common errors in the writing of items and to secure the validity of the construct (Haladyna and Rodriguez, 2013; Commission and of Test Publishers, 2022). However, this systematic review process, while essential, is highly time-consuming, especially in large-scale assessment contexts.

To overcome these challenges and efficiently support assessments at scale, hybrid frameworks integrating automation with human supervision are increasingly adopted. An innovative example is the *Item Factory* developed for the Duolingo English Test (DET), an item review system that incorporates human-in-the-loop processes, particularly for the development of high-stakes international DET items (von Davier et al., 2024). The *Item Factory*

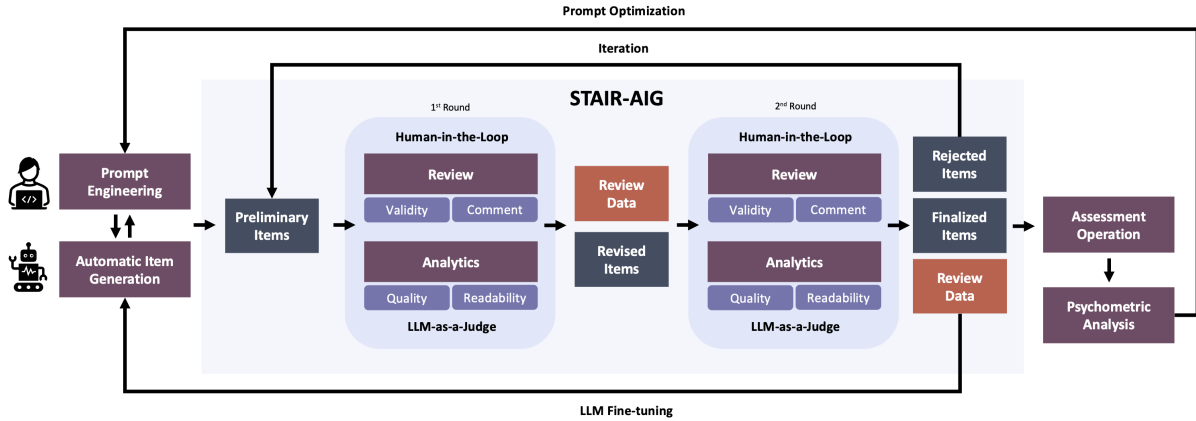


Figure 1: Pipeline of STAIR-AIG workflow

facilitates asynchronous collaboration between subject matter experts, supports reviewer calibration, and provides a structured audit trail of editorial decisions (von Davier et al., 2024). This approach not only maintains rigorous educational standards and test fairness, but also exemplifies how scalable and automated processes complemented by human oversight can enhance the quality and efficiency of assessment item review.

Item review tools, including *Item Factory*, are likely to be designed according to the types of items that are closely related to measurement constructs. To our knowledge, no open-source tool yet facilitates AIG item review for higher-order thinking skills. In the following, we present the STAIR-AIG tool and workflow as a human-in-the-loop procedure to review and optimize AIG items for CT.

3 Development of STAIR-AIG

3.1 STAIR-AIG Workflow

STAIR-AIG is developed as an iterative HAIC framework that goes beyond the static and unidirectional AIG process by continuously incorporating human reviewers' feedback to refine LLM behavior. By providing supplementary NLP features to human reviewers, human experts are expected to integrate their domain knowledge more effectively. In addition, it envisions the advancement of an AIG pipeline by automatically converting human reviews into training data for LLMs. These evaluations and human expert insights are then used to iteratively improve both AIG models through reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2020) and optimize their associated prompts (Lin et al., 2024),

ultimately reducing the human effort required to develop and review items that target complex cognitive constructs such as CT.

Figure 1 represents a comprehensive pipeline of the STAIR-AIG workflow. As seen in the figure, the STAIR-AIG workflow is organized as a multistage iterative loop. Preliminary items generated through prompt engineering by LLMs undergo initial evaluation and review via automated analytics, where LLMs function as auxiliary reviewers. Human reviewers then assess each item based on qualitative criteria, including content validity, appropriateness, and cognitive alignment using the STAIR-AIG tool. Importantly, reviewers provide both three-point scale ratings and open-ended feedback, and in many cases, they can directly edit the content of items. These structured data, comments, scores, and editorial changes are saved as review metadata and would be utilized to refine and enhance the performance of the AIG models.

What distinguishes STAIR-AIG is its integration of these human-generated review signals into both upstream and downstream optimization processes. On the one hand, reviewer feedback is used for prompt optimization (Lin et al., 2024), improving future item generation by refining how prompts are constructed. On the other hand, the accumulated data from reviews and edits serves as training data for RLHF (Christiano et al., 2017), fine-tuning the LLM to produce items that better align with expert judgment and the intended assessment objectives. As shown in Figure 2, this feedback loop system, inspired by the HAIC framework presented in Huang (2019), exemplifies a HAIC-based workflow designed to optimize the quality of AIG items.

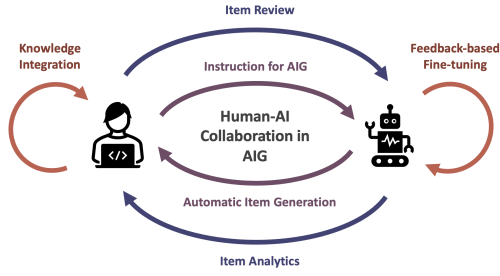


Figure 2: HAIC workflow in AIG

3.2 STAIR-AIG Modules

The STAIR-AIG system comprises two central modules designed to systematically evaluate and continuously improve the AIG process.

3.2.1 Item Analysis Module

The item analysis module operates as the preliminary review stage. Items undergo automated analysis based on quantitative linguistic metrics. The metrics include traditional NLP features, including type-token ratio, sentence length, and readability indices such as Flesch-Kincaid grade level, ensuring that the items are written clearly for the target age groups (Collins-Thompson, 2014). These metrics are selected to capture linguistic features that influence item clarity, cognitive load, and appropriateness, and to support early-stage quality screening for human review.

- **Type-Token Ratio (TTR):** A common measure of lexical diversity, defined as

$$\text{TTR} = \frac{|V|}{|W|} \quad (1)$$

where $|V|$ is the number of unique types and $|W|$ is the total number of tokens.

- **Average Sentence Length (ASL):** A measure of syntactic complexity, defined as

$$\text{ASL} = \frac{N_w}{N_s} \quad (2)$$

where N_w is the total words count and N_s is the total number of sentences.

- **Average Syllables per Word (ASW):** A measure of word complexity, defined as

$$\text{ASW} = \frac{N_{syll}}{N_w} \quad (3)$$

where N_{syll} is the total number of syllables and N_w is the total number of words.

- **Flesch-Kincaid Grade Level:** A readability index that estimates the school grade level required to understand a given text (Kincaid et al., 1975), calculated as

$$\text{FKGL} = 0.39 \cdot \text{ASL} + 11.8 \cdot \text{ASW} - 15.59 \quad (4)$$

We compute linguistic features by applying an XLM-RoBERTa tokenizer as a text preprocessing step (Conneau et al., 2020). Leveraging these linguistic features, the module automatically evaluates text difficulty, grade-level appropriateness, and lexical diversity metrics, which significantly reduce the workload placed upon human reviewers, thereby enhancing review efficiency and providing human reviewer with detailed item specification information to facilitate effective and timely review.

3.2.2 Item Review Module

Central to the STAIR-AIG system is the item review module, a structured interface that enables human experts to systematically evaluate AI-generated items. Items approved by the initial automated analysis are presented through this module interface. This module segments each item into specific components, such as passages, questions, and answer choices, allowing reviewers to provide detailed evaluations of each component.

Expert reviewers evaluate each component using a three-point quality scale that serves as the basis for determining whether an item would be accepted, revised, or discarded. Reviewer feedback serves a dual purpose. Qualitative comments contribute to improving the item generation prompts, while direct revision suggestions help finalize the item for operational use and also support future model refinement. Through this human-in-the-loop iterative process, STAIR-AIG continuously improves the quality and validity of the items. Once finalized, high-quality items generated by AI and modified by human experts are stored in an item bank for operational deployment. Item review module as an interface of STAIR-AIG is shown in Figure 3.

4 Empirical Research

In this empirical study, only the first round review was performed within the STAIR-AIG workflow. This initial implementation served to examine the utility of the tool and to investigate the discrepancies of review results between the human reviewer and LLM judges at the early stage of the proposed STAIR-AIG workflow.

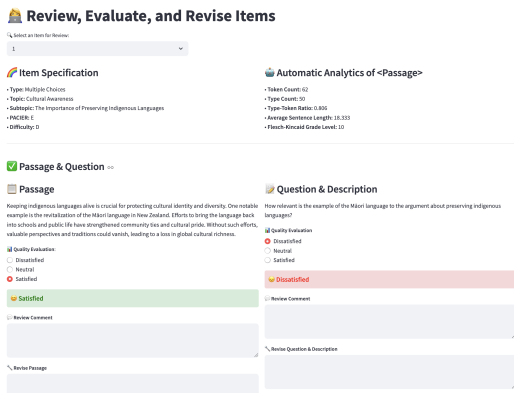


Figure 3: STAIR-AIG interface

4.1 Data

The items that were reviewed through STAIR-AIG in this study were developed by a MACAT, specializing in CT frameworks and evaluation solutions. They are based on a framework that measures and assesses CT competencies across six subdomains—Problem solving, Analysis, Creative thinking, Interpretation, Evaluation, and Reasoning (PACIER) (MACAT, 2025; Shin et al., 2025).

In this round, a total of 24 AI-generated items were reviewed, comprising multiple choice (MC) and fill-in-the-blank (FIB) types. Specifically, the assessment included 18 MC items (3 per PACIER domain) and 6 FIB items (1 per PACIER domain). Although actual CT assessment typically employs 4 choices for MC items and 3 choices for FIB items, the initial AIG items were deliberately prompted to generate 10 and 6 choices respectively, to promote a rigorous quality review without being forced to choose from all the bad choices. As for an example, an operating sample item for MACAT’s CT assessment is illustrated in Figure 4.

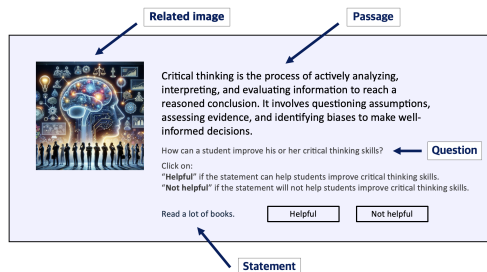


Figure 4: Sample item of CT assessment.

4.2 Item Review by Human Expert

The key review process for the 24 AIG items was conducted by a human expert who specialized in

CT domain. The human expert reviewed each item systematically following the instructions and steps using the STAIR-AIG tool, indicating the quality of the items and their components on three-point rating scales.

- **Dissatisfied:** Fundamentally flawed or inappropriate item for CT assessment, and thus should be discarded. (Score: 1)
- **Neutral:** Requires revisions to improve clarity and relevance or modification of difficulty level. (Score: 2)
- **Satisfied:** Suitable for immediate use or requires minimal edits. (Score: 3)

Specifically, the expert provided ratings and comments on each of the item components, including passages, questions, choices, and overall quality of the items, referencing the analytic information provided by the item analysis module. Revision suggestions were also written directly by the expert in the open text field when necessary. Items that were rated as *neutral* or *satisfied* received detailed revision suggestions to support iterative refinement. After the review, all data including evaluations, revisions, and edits were provisionally stored as a CSV file for future model fine-tuning.

4.3 Item Quality Review by LLMs

In parallel to the human review, four OpenAI LLMs (GPT-4o, GPT-4.5-preview, o1-mini, and o3-mini) performed independent quality assessments using the LLM-as-a-judge methodology (Zheng et al., 2023). Although prior work has shown that LLM-as-a-judge is closely aligned with human preferences on a variety of tasks (Zheng et al., 2023; Gu et al., 2025), there is a lack of prior research exploring its applicability in the context of complex cognitive skills, specifically in the evaluation of the quality of the AIG items. Therefore, we explored the possibility of using LLM-as-a-judge as an additional reviewer.

Each model evaluated the AIG items based on the same criteria and the same interface used by human reviewers. The prompts were carefully aligned and mirrored with the human evaluation guidelines to ensure methodological consistency. To maintain independence between human and LLM evaluations, we adopted zero-shot learning as an in-context learning approach in which models relied solely on their pre-trained knowledge without being

provided with any task-specific examples (Brown et al., 2020). This prevents potential contamination between evaluation sources while utilizing LLM’s generalized reasoning capabilities, distinct from human influence. The evaluations by LLMs were then compared with human review. Detailed prompts are provided in the Appendix A.

5 Results

5.1 Quantitative Results

5.1.1 Comparison of Human Reviews with LLM-generated Reviews

Analysis of 18 MC and 6 FIB items reveals differences in rating patterns between the human expert and LLM judges. The descriptive statistics for both item types are reported in Table 1, indicating that a human expert tends to assign lower scores overall and exhibits greater variability across all items.

In contrast, LLM judges consistently delivered higher scores across all evaluated dimensions with lower standard deviations. The o3-mini model, in particular, demonstrated extreme uniformity, assigning perfect or near-perfect scores with minimal variance. Specifically, even among LLMs, there is a subtle stratification that GPT-4.5-preview and GPT-4o exhibited slightly more variation and lower means than o3-mini. Also, in MC evaluations, the scores of the o1-mini model were closer to those of the human expert, especially in question quality.

Concretely, as illustrated in Figure 5 and Figure 6, LLMs tend to be consistently generous in their evaluations, while the human expert demonstrated a more critical and sensitive attitude marked by greater variability. A particularly notable pattern emerges in the ‘Question Rating’ category for FIB items, that the human expert consistently assigned the highest score to the 6 items. This uniformity is not coincidental. Since all FIB items had an identical question format, a consistent rating is justifiable and is an expected result, whereas some LLMs failed to reflect this.

5.1.2 Distribution of Ratings across Evaluators

Table 2 further illuminates the contrasting behaviors of human expert and LLM judges in evaluating the quality of AIG items. A notable pattern is the relatively frequent use of the lowest rating *Dissatisfied* (score of 1) by the human expert. Rather than indicating inconsistency, this tendency may reflect the human expert’s awareness of the qualitative

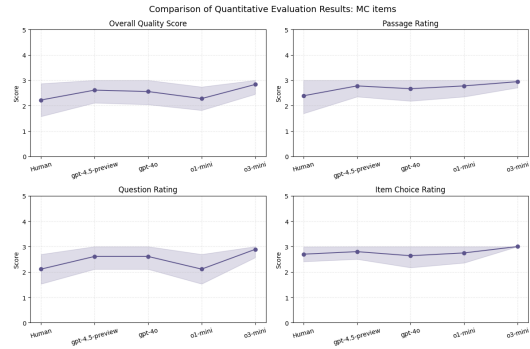


Figure 5: Rating patterns by evaluators for MC items

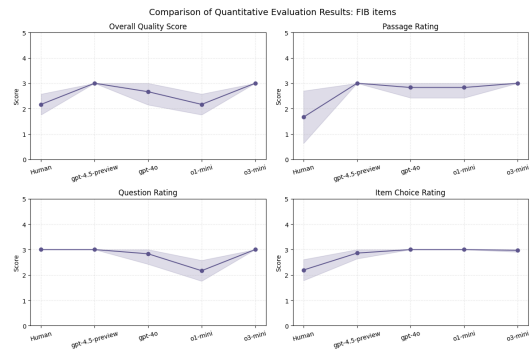


Figure 6: Rating patterns by evaluators for FIB item

aspects of the content of the item. This indicates that contextual appropriateness, coherence, and educational validity are often more readily detected through human expert, whereas automated systems may overlook such nuanced deficiencies.

In comparison, LLMs rarely gave the lowest rating of *Dissatisfied*. For example, o3-mini gave 100% *Satisfied* (score of 3) ratings in nearly every category. In the human rater effect study, this can be interpreted as a leniency or generosity (Wolfe, 2004). Even more conservative models such as o1-mini and GPT-4o showed minimal to zero use of the lowest category across MC and FIB items.

Furthermore, the human evaluator showed a more frequent use of the *Neutral* category (score of 2), which accounts for most of the responses. This middle-ground positioning can be interpreted as a nuanced case-by-case approach by the human evaluator, in contrast to the strong tendency of LLMs to assign the highest rating across most items.

5.2 Qualitative Feedback from Human Expert

To closely examine the reviews provided by the human expert, we performed a qualitative analysis of the reviewer’s written comments. Table 3 lists four themes that categorize and summarize the feedback. The human expert specialized in the as-

Table 1: Descriptive statistics for MC and FIB item reviews by evaluators

Item Type	Evaluator	Overall Quality Score				Passage Rating				Question Rating				Item Choices Rating			
		Mean	Std	Min	Max	Mean	Std	Min	Max	Mean	Std	Min	Max	Mean	Std	Min	Max
MC	Human	2.22	0.65	1	3	2.39	0.70	1	3	2.11	0.58	1	3	2.70	0.29	2	3
	GPT-4.5-preview	2.61	0.50	2	3	2.78	0.43	2	3	2.61	0.50	2	3	2.80	0.30	2	3
	GPT-4o	2.56	0.51	2	3	2.67	0.49	2	3	2.61	0.50	2	3	2.60	0.54	1	3
	o1-mini	2.28	0.46	2	3	2.78	0.43	2	3	2.11	0.58	1	3	2.81	0.35	2	3
	o3-mini	2.82	0.39	2	3	2.94	0.24	2	3	2.89	0.32	2	3	3.00	0.00	3	3
FIB	Human	2.17	0.41	2	3	1.67	1.03	1	3	3.00	0.00	3	3	2.19	0.41	1	3
	GPT-4.5-preview	3.00	0.00	3	3	3.00	0.00	3	3	3.00	0.00	3	3	2.86	0.22	2	3
	GPT-4o	2.67	0.52	2	3	2.83	0.41	2	3	2.83	0.41	2	3	3.00	0.00	3	3
	o1-mini	2.17	0.41	2	3	2.83	0.41	2	3	2.17	0.41	2	3	3.00	0.00	3	3
	o3-mini	3.00	0.00	3	3	3.00	0.00	3	3	3.00	0.00	3	3	2.97	0.07	2	3

Table 2: Rating frequency and proportion for MC and FIB item reviews by evaluators

Item Type	Evaluator	Overall Quality			Passage			Question			Item Choice		
		Dissatisfied	Neutral	Satisfied	Dissatisfied	Neutral	Satisfied	Dissatisfied	Neutral	Satisfied	Dissatisfied	Neutral	Satisfied
MC	Human	2 (11%)	10 (56%)	6 (33%)	2 (11%)	7 (39%)	9 (50%)	2 (11%)	12 (67%)	4 (22%)	4 (2%)	46 (26%)	130 (72%)
	GPT-4.5	0 (0%)	7 (39%)	11 (61%)	0 (0%)	4 (22%)	14 (78%)	0 (0%)	7 (39%)	11 (61%)	3 (2%)	30 (17%)	147 (82%)
	GPT-4o	0 (0%)	8 (44%)	10 (56%)	0 (0%)	6 (33%)	12 (67%)	0 (0%)	7 (39%)	11 (61%)	28 (16%)	9 (5%)	143 (79%)
	o1-mini	0 (0%)	13 (72%)	5 (28%)	0 (0%)	4 (22%)	14 (78%)	2 (11%)	12 (67%)	4 (22%)	15 (8%)	15 (8%)	150 (83%)
	o3-mini	0 (0%)	3 (17%)	15 (83%)	0 (0%)	1 (6%)	17 (94%)	0 (0%)	2 (11%)	16 (89%)	0 (0%)	0 (0%)	180 (100%)
FIB	Human	0 (0%)	5 (83%)	1 (17%)	4 (67%)	0 (0%)	2 (33%)	0 (0%)	0 (0%)	6 (100%)	10 (28%)	9 (25%)	17 (47%)
	GPT-4.5	0 (0%)	0 (0%)	6 (100%)	0 (0%)	0 (0%)	6 (100%)	0 (0%)	0 (0%)	6 (100%)	0 (0%)	5 (14%)	31 (86%)
	GPT-4o	0 (0%)	2 (33%)	4 (67%)	0 (0%)	1 (17%)	5 (83%)	0 (0%)	1 (17%)	5 (83%)	0 (0%)	0 (0%)	36 (100%)
	o1-mini	0 (0%)	5 (83%)	1 (17%)	0 (0%)	1 (17%)	5 (83%)	0 (0%)	5 (83%)	1 (17%)	0 (0%)	0 (0%)	36 (100%)
	o3-mini	0 (0%)	0 (0%)	6 (100%)	0 (0%)	0 (0%)	6 (100%)	0 (0%)	0 (0%)	6 (100%)	0 (0%)	1 (3%)	35 (97%)

assessment of CT skills provided detailed comments, such as concerns about vague terminology, overly obvious item structure, conceptual inconsistencies, and cultural bias, which were often overlooked by LLM judges. These qualitative insights are stored as data and will play an instrumental role in shaping the future STAIR-AIG protocol, particularly in optimizing the prompts used for AIG and in systematizing the rubrics for the LLM-based review.

It is also worth noting that the human expert raised the issue of the content validity of some AIG items. Specifically, some items were on the borderline of assessing CT or reading comprehension. In such cases, the human expert not only provided a detailed explanation of their reasoning but also directly revised the wording of the items to better align with the intended purpose of the assessment. Such feedback can also be saved as data and used to fine-tune the LLMs, ultimately supporting the development of more valid and reliable AIG-powered assessment content.

6 Conclusions & Implications

6.1 Conclusions

This study introduces STAIR-AIG, a structured, human-in-the-loop framework designed to improve the quality and validity of AI-generated assessment items. Using the STAIR-AIG tool, we collected and compared item reviews from a human expert

and four OpenAI LLMs. Our quantitative and qualitative analyses revealed that, while LLM’s evaluations demonstrated high consistency, their feedback was generally superficial and overly lenient. Often, LLMs neglected critical issues such as ambiguous terminology, cultural insensitivity, and insufficient cognitive depth. In contrast, the human expert provided more critical and nuanced feedback, effectively identifying subtle yet significant flaws.

The two core modules of STAIR-AIG significantly support human reviewers in conducting rigorous, systematic evaluations aligned with the test-taker’s background and the assessment goals, enhancing review efficiency. Notably, the discrepancies observed between human reviewers and LLM judges underscore the importance of a human-in-the-loop framework and an iterative review process. Ultimately, the data collected through these structured reviews is expected to improve the quality of AIG items and facilitate the development of more robust and refined assessment items.

6.2 Implications

As an example of a human-in-the-loop approach to AIG, this study sets the groundwork for extending STAIR-AIG into a comprehensive, full-cycle framework encompassing AIG, collaborative human-AI review, iterative refinement, pilot testing, psychometric validation, and model retrain-

Feedback Category	Review Comments
Terminology & Language Use Vague, overly technical, and structurally complex, which makes it misaligned with the assessment’s purpose.	<ul style="list-style-type: none"> - "Do not use so many different words for the same meaning." - "(...) is a difficult formulation for not-so-strong readers." - "(...) is unnecessarily vague scientific jargon." - "The term (...) might be too technical for many students and may lead to incorrect interpretations."
Item Construction & Clue Issues Wording or structure that makes answers too obvious or misleads test-takers.	<ul style="list-style-type: none"> - "When mentioning acronym, use full name, and in all further mentions, use acronym." - "Change order to avoid misinterpretation." - "Answer appears verbatim in the passage." - "Too simple and easy to see the answer." - "Why use the term (...) whereas in all statements you use the term (...)? Be consistent."
Conceptual Accuracy & Fit Inaccurate or inconsistent statements, which make it unsuitable for valid assessment.	<ul style="list-style-type: none"> - "I have read some publications about (...), but the definition that is used here does not really fit very well." - "Biased or misleading conclusion." - "(...) and (...) depends on interpretation."
Cultural Sensitivity Culturally biased, which offers a limited perspective and potentially disadvantaging test-takers from diverse backgrounds.	<ul style="list-style-type: none"> - "The concept of the (...) varies by culture and perspective." - "(...) might be ideal in some contexts, while (...) may carry a clearer negative connotation." - "(...) portrayed in a one-sided positive light." - "(...) is culturally or ethically biased."

Table 3: Categorization of reviewer feedback and representative comments

ing. The human-generated reviews collected in this study would serve as a valuable resource for the first round of LLM refinement. Drawing on this empirical data, future work would focus on optimizing LLM prompting strategies and applying RLHF to improve both the quality and validity of AI-generated items. This process will help establish a more data-driven and feedback-informed basis for optimizing AIG systems.

In addition, this research contributes to the emerging field of HAIC-based test design and administration, where prior work remains limited. By demonstrating the utility of structured human reviews in guiding both AIG prompting and model fine-tuning, the study highlights a scalable pathway for the application of AI to educational measurement. Similar to how the *Item Factory* is used for DET, the proposed STAIR-AIG tool is being implemented for MACAT’s CT assessment. The number of CT assessment items has rapidly doubled with the STAIR-AIG process, and the tool is being fully implemented to create an item bank of human-authored items alongside AIG for the CT assessment (Shin et al., 2024). This HAIC-driven approach showcases the increasing potential for the scholarly and sustainable use of AI in education.

6.3 Limitations

Despite its promise, this study has several limitations. First, the study was confined to an initial review by a human expert and four OpenAI LLMs,

followed by a comparative analysis of their ratings. The end-to-end STAIR-AIG workflow process, particularly the integration and refinement of the AIG model through iterative review, has yet to be realized. Future work will involve more comprehensive testing of the entire STAIR-AIG pipeline.

Second, although the STAIR-AIG framework is designed to support multiple rounds of review, the current study only included one round of review by one expert reviewer. Consequently, the results may not reflect the full potential of iterative refinement, thereby limiting the framework’s generalizability. Future research should explore the point at which discrepancies between LLMs and expert ratings converge. This will help us understand how LLMs behave when judging higher-order thinking skills, as well as inform the optimal stage for finalizing items for operational use and determining the maximum number of review cycles.

Third, while the item-review module was helpful to human reviewers, it could only analyze superficial metrics, such as TTR, ASL, and conventional readability indices. In the present study, grade-level suitability was judged solely based on these readability measures. Moving forward, the review module will integrate additional linguistic indicators that capture semantic dimensions in order to provide reviewers with more comprehensive support. Similarly, we did not directly measure whether the module substantially reduced the time reviewers

needed to complete their tasks. Therefore, future research would evaluate the practical effectiveness of STAIR-AIG by determining the degree to which it aids item review and the amount of time it saves compared to standard, tool-free review procedures.

Lastly, LLMs were given instructions that closely mirrored those provided to the human reviewer, yet their evaluations consistently exhibited leniency. To achieve a more harmonious integration of human and LLM ratings, future work should consider various prompt engineering techniques to calibrate LLM judgments more closely with the human evaluation standard in the CT domain. Furthermore, optimizing prompts accompanied by the psychometric results of the test data is expected to improve AIG models' ability to accurately generate and evaluate item difficulty and distractor plausibility. This would, in turn, strengthen the efficiency and validity of human-AI collaboration in test development.

Acknowledgments

This research was conducted in collaboration with the MACAT International Ltd., who provided support. We also sincerely appreciate the insightful comments and thoughtful suggestions on potential future directions for this research from the anonymous reviewers.

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.
- Isaac I. Bejar. 1996. Generative response modeling: Leveraging the computer as a test delivery medium. ETS Research Report RR-96-13, Educational Testing Service, Princeton, NJ.
- Isaac I. Bejar. 2002. Generative testing: From conception to implementation. In Sidney H. Irvine and Patrick C. Kyllonen, editors, *Item Generation for Test Development*, pages 199–218. Lawrence Erlbaum Associates, Mahwah, NJ.
- Ummugul Bezirhan and Matthias von Davier. 2023. [Automated reading passage generation with openai's large language model](#). *Computers and Education: Artificial Intelligence*, 5:100161.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). *Preprint*, arXiv:2005.14165.
- Sumie Chan, Noble Lo, and Alan Wong. 2025. [Leveraging generative ai for enhancing university-level english writing: comparative insights on automated feedback and student engagement](#). *Cogent Education*, 12(1):2440182.
- Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307.
- 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.
- International Test Commission and Association of Test Publishers. 2022. Guidelines for technology-based assessment. <https://www.intestcom.org/page/28> and <https://www.testpublishers.org/white-papers>. ISBN 979-8-88862-517-0.
- 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.
- George Fragiadakis, Christos Diou, George Kousiouris, and Mara Nikolaidou. 2025. [Evaluating human-ai collaboration: A review and methodological framework](#). *Preprint*, arXiv:2407.19098.
- Mark J Gierl and Hollis Lai. 2013. Instructional topics in educational measurement (ITEMS) module: Using automated processes to generate test items. *Educational Measurement: Issues and Practice*, 32(3):36–50.
- 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.
- Marcelo Guerra Hahn, Silvia Margarita Baldiris Navarro, Luis De La Fuente Valentín, and Daniel Burgos. 2021. [A systematic review of the effects of automatic scoring and automatic feedback in educational settings](#). *IEEE Access*, 9:108190–108198.

- Thomas M. Haladyna and Michael C. Rodriguez. 2013. *Developing and Validating Test Items*. Routledge, London, UK.
- Jiangang Hao, Alina A. von Davier, Victoria Yaneva, Susan Lottridge, Matthias von Davier, and Deborah J. Harris. 2024. [Transforming assessment: The impacts and implications of large language models and generative ai](#). *Educational Measurement: Issues and Practice*. All authors contributed equally.
- Janet Huang. 2019. Human-ai co-learning for data-driven ai. <https://speakerdeck.com/janetyc/human-ai-co-learning-for-data-driven-ai>. Accessed: 2025-05-03.
- J. Peter Kincaid, Richard P. Fishburne, Robert L. Rogers, and Brad S. Chissom. 1975. [Derivation of new readability formulas \(automated readability index, fog count and flesch reading ease formula\) for navy enlisted personnel](#). Research Branch Report 8-75, Naval Technical Training, U.S. Naval Air Station, Millington, TN. Archived from the original on December 10, 2020.
- 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.
- Ehsan Latif and Xiaoming Zhai. 2024. [Fine-tuning chatgpt for automatic scoring](#). *Computers and Education: Artificial Intelligence*, 6:100210.
- 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.
- Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024. [Prompt optimization with human feedback](#). *Preprint*, arXiv:2405.17346.
- Ou Lydia Liu, Liyang Mao, Lois Frankel, and Jun Xu. 2016. [Assessing critical thinking in higher education: The heighten™ approach and preliminary validity evidence](#). *Assessment and Evaluation in Higher Education*, 41(5):677–694.
- S. A. Luchini, N. T. Maliakkal, P. V. DiStefano, A. Laverghetta Jr., J. D. Patterson, R. E. Beaty, and R. Reiter-Palmon. 2025. [Automated scoring of creative problem solving with large language models: A comparison of originality and quality ratings](#). *Psychology of Aesthetics, Creativity, and the Arts*. Advance online publication.
- MACAT. 2025. [Critical thinking assessments](https://www.macat.com/critical-thinking). <https://www.macat.com/critical-thinking>. Retrieved April 16, 2025.
- Hyo Jeong. Shin, Seewoo. Li, Salah. Khalil, and Alina A. von Davier. 2024. [Designing for adaptive testing using automatically generated items](#). In *Proceedings of the Annual Meeting of the International Association for Computerized Adaptive Testing (IACAT)*, Seoul, Korea.
- Hyo Jeong. Shin, Seewoo. Li, Jihoon. Ryoo, Alina A. von Davier, T. Lubart, and Salah. Khalil. 2025. [The nature and measure of critical thinking: The pacier framework and assessment](#). Manuscript submitted for publication.
- Mark K. Singley and Randy E. Bennett. 2002. [Item generation and beyond: Applications of schema theory to mathematics assessment](#). In Sidney H. Irvine and Patrick C. Kyllonen, editors, *Item Generation for Test Development*, pages 361–384. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.
- Venkat Srinivasan. 2022. [AI & learning: A preferred future](#). *Computers and Education: Artificial Intelligence*, 3:100062.
- Alina A. von Davier and Jill Burstein. 2024. [Ai in the assessment ecosystem: A human-centered ai perspective](#). In Peter Ilic, Ian Casebourne, and Rupert Wegerif, editors, *Artificial Intelligence in Education: The Intersection of Technology and Pedagogy*, volume 261 of *Intelligent Systems Reference Library*. Springer, Cham.
- Alina A. von Davier, Andrew Runge, Yena Park, Yigal Attali, Jacqueline Church, and Geoff LaFlair. 2024. [The item factory: Intelligent automation in support of test development at scale](#). In *Machine Learning, Natural Language Processing, and Psychometrics*, pages 1–25. Information Age Publishing, Charlotte, NC.
- Shan Wang, Fang Wang, Zhen Zhu, Jingxuan Wang, Tam Tran, and Zhao Du. 2024. [Artificial intelligence in education: A systematic literature review](#). *Expert Systems with Applications*, 252(Part A):124167.
- Edward W Wolfe. 2004. [Identifying rater effects using latent trait models](#). *Psychology Science*, 46:35–51.
- World Economic Forum. 2015. [New vision for education: Unlocking the potential of technology](#). <https://widgets.weforum.org/nve-2015/chapter1.html>. Accessed April 14, 2025.
- 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](#). *Preprint*, arXiv:2306.05685.
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. [Fine-tuning language models from human preferences](#). *Preprint*, arXiv:1909.08593.

A Appendix

A.1 Prompt for Item Review by LLMs

The following is an excerpt of the prompt used to instruct the LLMs in reviewing the quality of CT items. The prompt defines the evaluation criteria, output structure, and PACIER framework to assess item quality.

Listing 1: System prompt

```
You are a Critical Thinking Assessment's Item Review Expert with extensive experience in educational evaluation and test design, specializing in critical thinking.

Your role is to systematically evaluate the quality of test items based on established frameworks, ensuring fairness, reliability, and alignment with learning objectives.
```

```
Item_1_Choice_1 Review, Item_1_Choice_1 Rating,
Item_1_Choice_1 Revision Suggestion, ... (repeat for
Choices 2 through 10)
```

```
## Additional Guidelines
```

- Ensure alignment with cognitive and linguistic proficiency standards.
- **Maintain consistency** across evaluations to avoid bias.
- Do not include markdown, bullet points, or additional explanations.
- Return only key-value pairs as output.

Listing 2: User prompt: Review Context

```
## Review Context
- The exam items are designed for Grade 7~8 learners.
- Each item consists of a Passage, a Question, and 6 Answer Choices (each with an Explanation).
- Your task is to rigorously evaluate the quality of each component and provide structured feedback.

## PACIER Framework (Cognitive Process Dimensions)
The PACIER framework categorizes cognitive processes into six distinct levels:
- Problem-Solving (P): (...)
- Creative Thinking (C): (...)
- Interpretation (I): (...)
- Evaluation (E): (...)
- Reasoning (R): (...)
Each test item should align with at least one PACIER category, ensuring it assesses critical thinking skills effectively.
```

Listing 3: User prompt: Review Methods

```
## Evaluation Methodology
1. Assessment Criteria
- Passage: Relevance, clarity, and cognitive demand.
- Question: Alignment with passage, clarity, and ability to assess critical thinking.
- Answer Choices: Plausibility of distractors, clarity, and correctness of explanations.

2. Comparative Judgment
- Evaluate each component relative to high-quality reference items to ensure consistency.

3. Rating Scale
- Dissatisfied: Fundamentally flawed or inappropriate for assessment and thus discarded without revision suggestions.
- Neutral: Requires revisions to improve clarity, relevance, or difficulty. You should provide detailed feedback and specific revision recommendations.
- Satisfied: Suitable for immediate use or required minimal edits. You could directly accept these items or suggest minor enhancements.

4. Actionable Feedback
- Provide concise but specific feedback justifying each rating.

5. Final Output Format (Plain Key-Value Pairs, CSV-Ready)
Output only concise final results in plain key-value pairs (one per line) using the following CSV column structure:

Item Number, Type, Topic, Subtopic, PACIER, Difficulty, Overall Quality Score, Overall Comment, Passage Comment, Passage Rating, Passage Revision, Question Comment, Question Rating, Question Revision,
```

UPSC2M: Benchmarking Adaptive Learning from Two Million MCQ Attempts

Kevin Shi, Karttikeya Mangalam

SigIQ.ai

Correspondence: kevin@sigiq.ai

Abstract

We present UPSC2M, a large-scale dataset comprising two million multiple-choice question attempts from over 46,000 students, spanning nearly 9,000 questions across seven subject areas. The questions are drawn from the Union Public Service Commission (UPSC) examination, one of India’s most competitive and high-stakes assessments. Each attempt includes both response correctness and time taken, enabling fine-grained analysis of learner behavior and question characteristics. Over this dataset, we define two core benchmark tasks: question difficulty estimation and student performance prediction. The first task involves predicting empirical correctness rates using only question text. The second task focuses on predicting the likelihood of a correct response based on prior interactions. We evaluate simple baseline models on both tasks to demonstrate feasibility and establish reference points. Together, the dataset and benchmarks offer a strong foundation for building scalable, personalized educational systems. We release the dataset and code to support further research at the intersection of content understanding, learner modeling, and adaptive assessment: github.com/kevins-hi/upsc2m.

1 Introduction

As digital learning platforms become increasingly central to education, there is growing demand for intelligent systems that can adapt to individual learners, curate relevant content, and deliver targeted assessments. At the heart of such systems lie two fundamental modeling tasks: estimating the difficulty of educational content and predicting student performance. These capabilities underpin a wide range of applications—from personalized question selection to real-time learner diagnostics. When combined, they serve as the foundation for fully automated adaptive learning systems that dynamically tailor instruction based on both content complexity and learner proficiency.

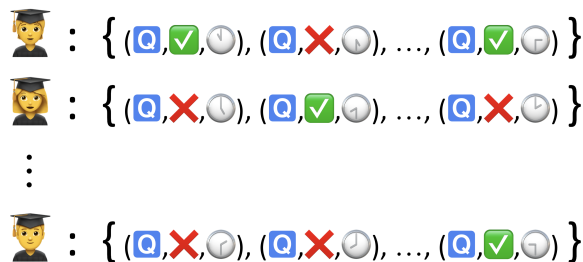


Figure 1: UPSC2M visualized as a list of students, each associated with a set of question attempts. Each attempt records the student ID, question ID, selected answer, whether it was correct, and the time taken to answer.

Statistic	Count
Unique Students	46,235
Unique Questions	8,973
Total Interactions	1,962,573

Table 1: Summary statistics for the UPSC2M dataset.

Much of the existing work in educational modeling has relied on small-scale classroom data or narrow subject domains, limiting the development of models for real-world settings. To bridge this gap, we introduce UPSC2M, a large-scale dataset of 1,962,573 question attempts from aspirants preparing for the Union Public Service Commission (UPSC) examination—one of India’s most competitive standardized tests. Spanning 8,973 questions across seven subjects, UPSC2M includes correctness and timing data from 46,235 students.

We propose two core tasks supported by this dataset. The first is *Question Difficulty Estimation*, where models predict empirical difficulty from question text alone. The second is *Student Performance Prediction*, where models forecast whether a student will answer a question correctly, given their prior interactions. These tasks reflect key challenges in real-world adaptivity and serve as modular building blocks for intelligent tutoring and assessment systems.

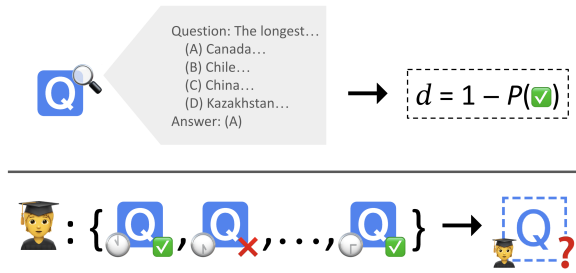


Figure 2: Illustration of the two benchmark tasks: question difficulty estimation (top) and student performance prediction (bottom). In the first task, the goal is to estimate the difficulty of a question—defined as one minus the empirical probability of a correct response—based solely on its text. In the second task, given a student’s prior question attempts, predict whether the student will correctly answer a new, unseen question.

Our contributions are threefold: (1) We release UPSC2M, a large-scale dataset capturing both question content and behavioral interaction data in a high-stakes, multi-subject testing context. (2) We define two core prediction tasks that capture key challenges in adaptive education. (3) We establish baselines and outline directions for future work. Together, UPSC2M and its benchmark tasks provide a robust foundation for research in scalable personalized education. By supporting more accurate models of question difficulty and student performance, this work lays the groundwork for educational platforms that adapt to individual needs at scale, expanding access to high-quality, personalized learning for students regardless of background.

2 Related Work

Large-scale Interaction Datasets A number of publicly available datasets have driven progress in student modeling and adaptive learning. The PSLC DataShop repository provides tens of thousands of student–problem interactions across diverse domains (Stamper et al., 2011), and the ASSISTments dataset offers fine-grained logs of middle-school mathematics practice. More recently, EdNet—a hierarchical dataset of over 130 million interactions from an online tutoring platform—has enabled deep sequence models at unprecedented scale (Choi et al., 2020). Our dataset, UPSC2M, complements these by focusing on a highly competitive, multi-subject exam context, capturing both correctness and response-time signals for UPSC aspirants.

Question Difficulty Estimation Classical item response theory (IRT) models difficulty as a latent

parameter estimated from response patterns (Lord, 1980), but they rely solely on interaction counts. Recent work has explored textual and semantic features to predict question difficulty directly from content (Blum and Corter, 2014). By pairing a large, annotated UPSC question bank with empirical accuracy rates, UPSC2M supports both purely content-based difficulty regression and hybrid approaches that integrate behavioral priors.

Student Performance Prediction Predicting learner outcomes has a long history in educational data mining. Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1994) and Performance Factor Analysis (PFA) (Pavlik Jr et al., 2009) established early probabilistic frameworks for tracking mastery. The advent of neural methods—e.g. Deep Knowledge Tracing (DKT) (Piech et al., 2015) has further improved sequence-based prediction. The UPSC2M dataset, with its detailed question content, student attempt outcomes, and rich temporal metadata, offers a new testbed for benchmarking such models on high-stakes exam data.

Applications for Adaptive Testing Adaptive testing algorithms—such as computerized adaptive testing (CAT) (Weiss, 2011)—depend critically on calibrated item difficulties and real-time performance estimates. Datasets that combine content features with large-scale attempt logs enable more responsive and personalized CAT systems. We anticipate that UPSC2M will spur advances in adaptive exam design, question selection strategies, and real-time learner diagnostics.

3 Proposed Dataset

3.1 Motivation and Collection

The UPSC exam is among the most competitive and high-stakes assessments in India, attracting over one million aspirants annually. The exam begins with Paper 1, a 2-hour, 100-question multiple-choice test that spans a broad spectrum of subjects, including history, polity, economy, science, geography, environment, and current affairs. Questions are carefully crafted to assess not only factual recall, but also higher-order reasoning, elimination strategies, and nuanced interpretive understanding under strict time constraints.

This examination offers a rich environment for studying educational modeling tasks. In particular, Paper 1 presents a uniquely challenging setting: questions span multiple knowledge domains, often

Subject	Question Count	Students per Question			Questions per Student		
		Mean	Median	Max	Mean	Median	Max
Current Affairs	1793	127.79	112	3576	20.13	5	1502
Polity	1487	348.00	79	3284	19.31	5	1425
History	1449	259.72	77	2559	20.94	5	1227
Economy	1111	183.86	72	1728	20.17	5	1069
Science	1094	139.81	19	2869	11.48	5	1008
Environment	1022	181.63	104	2801	11.70	4	913
Geography	1017	291.82	145	3055	19.93	5	956
Overall	8973	218.72	91	3576	42.45	8	6553

Table 2: Per-subject statistics in the UPSC2M dataset, including the number of questions and summary statistics for student and question engagement—measured as students per question and questions per student.

require implicit reasoning, and are attempted by a large student body interacting with a shared question bank. These characteristics make it an ideal testbed for developing, benchmarking, and evaluating adaptive educational technologies at scale.

To support research on adaptive learning algorithms, we deployed a custom learning platform targeting UPSC aspirants. Students engaged with a curated bank of 8,973 multiple-choice questions. Over a 2-year period, we collected interaction data from 46,235 students, totaling 1,962,573 question attempts. The resulting dataset has been rigorously cleaned and anonymized to ensure student privacy while retaining the signals necessary for downstream modeling tasks.

3.2 Dataset Schema

UPSC2M is a large-scale dataset comprising two components: an *attempts dataset* and a *questions dataset*. Each row in the attempts dataset represents a single interaction between a student and a question, capturing key fields including `user_id`, `question_id`, `user_answer`, `user_correct`, and `time_taken`. The accompanying questions dataset provides metadata for each question, including its `id`, `subject`, `question stem`, `multiple-choice options`, and the `correct answer`. While no student metadata is included, the dataset enables rich behavioral analysis: the `user_answer` field supports investigations into distractor effectiveness and common misconceptions, while the `time_taken` field—measured in seconds—offers a proxy for question engagement and fluency under time pressure. Each question is constrained to a 60-second limit, mirroring the real-world pacing of the UPSC exam.

3.3 Dataset Statistics

UPSC2M exhibits substantial scale and diversity in learner behavior across content categories. As shown in Table 2, each question is attempted by an average of 219 students, with some questions receiving over 3,000 attempts. This breadth of coverage stems from both the temporal dynamics of question exposure—where older or more prominently featured questions accumulate more interactions—and varying levels of learner interest across subject areas. Such variation necessitates models capable of generalizing across both high-frequency and low-frequency questions.

The average student attempted 42 questions, with the most active student answering over 6,500. This long-tailed distribution, typical of open educational platforms, supports modeling across a wide range of engagement levels. However, the low median number of questions per student indicates that many students engage only briefly, emphasizing the need for models that are robust to cold-start scenarios and sparse interaction histories.

4 Question Difficulty Estimation

4.1 Problem Formulation

We propose a task to estimate the empirical difficulty of a multiple-choice question using only its textual content. Each question is represented as a tuple (`id`, `subject`, `stem`, `options`, `answer`), where `stem` denotes the question prompt, `options` is a list of four candidate choices, and `answer` specifies the index of the correct option.

The empirical difficulty of a question is defined as $1 - p_{\text{correct}}$, rounded to two decimal places, where p_{correct} denotes the proportion of students in UPSC2M who answered the question correctly among those who attempted it. This definition re-

Method	RMSE	MAE	R ²
Training Mean	0.2057	0.1699	-0.0001
Text Embedding	0.1910	0.1543	0.1375

Table 3: Test set performance of regression models for question difficulty estimation. The *Training Mean* baseline predicts the mean difficulty for all training samples.

flects the intuition that more difficult questions are associated with lower observed accuracy.

Setup To support reproducible evaluation, the questions dataset includes a predefined `split` field designating train, validation, and test partitions in a 70/15/15 split. Each question is also annotated with a precomputed difficulty score based on the formulation above.

4.2 Text Embedding Regression

As a baseline for question difficulty estimation, we adopt a simple regression approach. Specifically, we encode the question using a frozen pretrained text encoder and train a small MLP to predict the associated difficulty.

Each question is serialized as a single string combining the stem and options, which is then passed through OpenAI’s `text-embedding-3-large` model—a general-purpose text embedding model. The resulting fixed-dimensional embedding serves as input to an MLP trained to minimize mean squared error against ground-truth difficulty scores. This approach offers a lightweight text-to-score mapping that sets a lower bound for models leveraging richer representations.

4.3 Results and Discussion

Our baseline achieves modest gains over a dummy regressor, reducing RMSE by 7.1% and MAE by 9.2%. While this demonstrates that semantic features carry some signal, the limited improvement underscores the difficulty of estimating question difficulty from text alone. These results motivate the incorporation of richer features—such as behavioral priors and structural cues.

Beyond benchmarking, automatic estimation of question difficulty has broad value in educational applications, enabling adaptive learning systems to personalize content to learner proficiency and maintain engagement. It also aids large-scale content management by facilitating question bank auditing, difficulty calibration, and the efficient construction

of balanced assessments with minimal manual effort. In generative settings, difficulty estimation models can act as verifiers to ensure that newly created questions meet predefined pedagogical goals. As educational platforms scale across diverse curricula and learner populations, automated question difficulty estimation will become a cornerstone of personalized adaptive learning infrastructure.

5 Student Performance Prediction

5.1 Problem Formulation

We propose a task to predict whether a student will answer a given multiple-choice question correctly, based on their prior interaction history. Each row in the `attempts` dataset represents a single interaction and is formatted as a tuple (`user_id`, `question_id`, `user_answer`, `user_correct`, `time_taken`), where `user_correct` is a binary label indicating whether the response was correct.

For evaluation, the fields `user_answer`, `user_correct`, and `time_taken` are treated as target variables—models may access them during training but must not use them as input features at inference time. At test time, each example is defined solely by the pair (`user_id`, `question_id`), and the model must predict whether the student answers the question correctly.

Formally, this task involves estimating the conditional probability that a student answers a question correctly, given their historical behavior. This formulation mirrors real-world scenarios in adaptive learning systems, where predicting a learner’s performance is essential.

Setup To facilitate reproducible evaluation, the `attempts` dataset includes a predefined `split` field that assigns each interaction to the training, validation, or test set, following an 80/10/10 ratio. The split is randomized at the interaction level, with post-processing to ensure that all students and questions in the validation and test sets also appear in the training set. This constraint ensures that models are evaluated on their ability to generalize to new interactions, rather than on cold-start cases with unseen students or questions.

5.2 Baselines

To contextualize the performance of more sophisticated models, we evaluate several simple baselines for this task.

Random and Zero Predictors As naive reference points, we consider two trivial classifiers. The *Random* baseline predicts correctness by sampling from the empirical label distribution in the training set, which shows a slight class imbalance (59.81% incorrect). The *Zero Predictor* always predicts the majority class (0 for incorrect), thereby serving as a worst-case lower bound on accuracy and calibration. While uninformative, these baselines are useful for verifying that more complex models exploit meaningful structure in the data.

Difficulty-Based Heuristic As a simple yet informative baseline, we ignore the student’s interaction history and estimate the probability of a correct response based solely on the difficulty of the target question. Specifically, we compute the predicted probability as $1 - d$, where d denotes the difficulty score of the question, defined in Section 4.1. This formulation assumes that all students have an equal chance of answering a question correctly, modulated only by how empirically difficult the question is for the population.

Despite its simplicity, this baseline captures coarse priors over questions and highlights the influence of item difficulty on student performance. Comparing it to history-aware models underscores the value of incorporating personalized signals.

5.3 Collaborative Filtering

To assess the utility of standard recommender system techniques for modeling student performance, we evaluate several collaborative filtering (CF) (Su and Khoshgoftaar, 2009) methods that treat the task as a matrix completion problem. The student-question interaction matrix is constructed from observed correctness labels, and models are trained to predict whether a student will answer a given question correctly.

We include matrix factorization methods such as Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF), which learn low-dimensional embeddings for students and questions based on historical responses. We also evaluate a bias-only model that estimates correctness using additive student and item biases, as well as a K-nearest neighbors (KNN) approach that aggregates correctness labels from similar students. Together, these methods span a spectrum of personalization strategies, from global baselines to fine-grained models that exploit relational structure in the data.

Method	Accuracy	AUC	Brier
Random	0.5204	0.5000	0.2400
Zero Predictor	0.6002	0.5000	0.3998
Heuristic	0.6698	0.7118	0.2080
KNN CF	0.6429	0.6461	0.2330
SVD CF	0.6755	0.7133	0.2076
NMF CF	0.6757	0.7157	0.2100
Bias Only CF	0.6788	0.7210	0.2051

Table 4: Test set performance of baseline methods on the student performance prediction task. *CF* denotes collaborative filtering.

These models serve as a classical baseline for student performance prediction, illustrating how much signal can be captured from past interactions alone, without access to question content.

5.4 Results and Discussion

Table 4 reports the performance of all baseline models on the student performance prediction task. The *Heuristic* model substantially outperforms trivial baselines, demonstrating that question difficulty alone provides a strong prior for estimating student success. This suggests that well-estimated item-level difficulty can serve as a meaningful signal, even without any personalization.

Among collaborative filtering methods, *Bias Only* yields the highest overall performance, while more expressive models such as *SVD*, *NMF*, and *KNN* fail to produce significant gains in accuracy. The high sparsity of the student-question matrix (99.62%) likely inhibits the ability of these models to learn effective representations or student neighborhoods, constraining their ability to capture student-specific patterns beyond simple item and student-level tendencies.

Predicting student performance is vital to adaptive educational systems, enabling personalized question selection, targeted review, and adaptive pacing to support diverse learners. When paired with difficulty estimation, it lays the groundwork for fully automated instruction by combining item-level insights with behavioral modeling. As educational platforms scale, these predictive capabilities are key to delivering truly individualized learning—ensuring each student receives the right content at the right time. Together, these tasks form the backbone of scalable, data-driven education.

Limitations

Our question difficulty estimation labels are based solely on correctness rates and ignore temporal or student-specific variation; future work may redefine difficulty through joint modeling of student and item characteristics, potentially incorporating response times. Our collaborative filtering models are likely hindered by the high prevalence of low-activity learners—the median questions attempted per student is just 8—which may limit generalization and overall performance. None of our current models incorporate response time features, which could offer valuable signals related to fluency or hesitation. Finally, while UPSC2M is large and diverse, its focus on one high-stakes exam context may limit direct transferability to other educational domains. Despite these limitations, we view our dataset and task formulations as a strong foundation for building more expressive, interpretable, and personalized models of learner behavior.

References

- Au Blum and James E. Corter. 2014. Estimating question difficulty and user ability in a collaborative question answering community. In *Workshop on Personalized and Adaptive Learning in EDM*.
- Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. 2020. [Ednet: A large-scale hierarchical dataset in education](#). In *Proceedings of the 21st International Conference on Artificial Intelligence in Education (AIED 2020)*, volume 12164 of *Lecture Notes in Computer Science*, pages 69–73. Springer.
- Albert T. Corbett and John R. Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*.
- Frederic M. Lord. 1980. *Applications of Item Response Theory to Practical Testing Problems*. Erlbaum.
- Philip I. Pavlik Jr, Hui Cen, and Kenneth R. Koedinger. 2009. [Performance factors analysis: A new alternative to knowledge tracing](#). In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press.
- Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 505–513.
- John C. Stamper, Kenneth R. Koedinger, Ryan S. J. d. Baker, Alida Skogsholm, Brett Leber, Sandy Demi, Shawnwen Yu, and Duncan Spencer. 2011. [Datashop: A data repository and analysis service for the learning science community](#). In *Proceedings of the 15th International Conference on Artificial Intelligence in Education (AIED)*, page 628, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. [A survey of collaborative filtering techniques](#). *Advances in Artificial Intelligence*, 2009:1–19.
- David J. Weiss. 2011. *Adaptive Testing*. Oxford University Press.

Can GPTZero’s AI Vocabulary Distinguish Between LLM-Generated and Student-Written Essays?

Veronica Juliana Schmalz

Anais Tack

KU Leuven

¹ Faculty of Arts, Research Unit Linguistics

² imec research group itec

veronicajuliana.schmalz@kuleuven.be

anais.tack@kuleuven.be

Abstract

Despite recent advances in AI detection methods, their practical application, especially in education, remains limited. Educators need functional tools pointing to *AI indicators within texts*, rather than merely estimating *whether* AI was used. GPTZero’s new AI Vocabulary feature, which highlights parts of a text likely to be AI-generated based on frequent words and phrases from LLM-generated texts, offers a potential solution. However, its effectiveness has not yet been empirically validated.

In this study, we examine whether GPTZero’s AI Vocabulary can effectively distinguish between LLM-generated and student-written essays. We analyze the AI Vocabulary lists published from October 2024 to March 2025 and evaluate them on a subset of the Ghostbuster dataset, which includes student and LLM essays. We train multiple Bag-of-Words classifiers using GPTZero’s AI Vocabulary terms as features and examine their individual contributions to classification.

Our findings show that simply checking for the presence, not the frequency, of specific AI terms yields the best results, particularly with ChatGPT-generated essays. However, performance drops to near-random when applied to Claude-generated essays, indicating that GPTZero’s AI Vocabulary may not generalize well to texts generated by LLMs other than ChatGPT. Additionally, all classifiers based on GPTZero’s AI Vocabulary significantly underperform compared to Bag-of-Words classifiers trained directly on the full dataset vocabulary. These findings suggest that fixed vocabularies based solely on lexical features, despite their interpretability, have limited effectiveness across different LLMs and educational writing contexts.

1 Introduction

Recently, the introduction of user-friendly interfaces such as ChatGPT (OpenAI, 2023) has made

a significant impact on education. An increasing number of students are using large language models (LLMs) to write essays (among other things), and this creates new challenges for educators to assess various skills and ensure academic integrity (Cotton et al., 2024). Even experienced teachers and those familiar with LLMs often struggle to tell apart student-written essays from those created by LLMs, as studies have shown (Fleckenstein et al., 2024; Waltzer et al., 2024; Perkins et al., 2024).

To address these challenges, numerous AI detection methods and tools have been developed (see Wu et al. 2025 for a review). However, as highlighted by Weber-Wulff et al. (2023), most detection tools in the market lack robustness with student texts and interpretability for non-expert users such as teachers. GPTZero (Tian and Cui, 2025), a popular AI detection tool, aims to offer a more transparent and interpretable solution. It analyzes texts for patterns, vocabulary and styles that are more common in AI-generated writing than in human writing, aiming to assist educators in verifying the authenticity of student work.

In October 2024, GPTZero introduced a new **AI Vocabulary**¹ feature (Figure 1), which highlights text parts that are likely to be AI-generated. This feature includes a list of the 50 words and phrases most commonly used by LLMs (Constantino, 2024), which can be interpreted as AI indicators, and is updated monthly. Each term is assigned a weight, indicating its frequency in AI-generated texts relative to human-written ones, and is accompanied by a contextual example. Since December 2024, the list has been expanded to include the top 100 words and phrases commonly used by AI. A key question, however, is whether this feature can be used to effectively distinguish LLM-generated essays from student-written ones. In this paper, we address this question by conducting

¹<https://gptzero.me/ai-vocabulary>

OCTOBER

Top 50 AI Words and Phrases Updated October 2024

These words and phrases are ranked based on the frequency they appear in AI documents, compared to human documents in our research of 3.3 million texts.

Phrase	Frequency used by AI vs. Human	Example
1. objective study aimed	269x more frequent in AI	The objective of the study aimed to uncover new insights into climate change.
2. research needed to understand	235x more frequent in AI	Further research is needed to understand the long-term effects.
3. despite facing	209x more frequent in AI	Despite facing numerous obstacles, the team succeeded.
4. play significant role shaping	182x more frequent in AI	Education systems play a significant role in shaping future generations.
5. crucial role in shaping	155x more frequent in AI	Parents play a crucial role in shaping their children's values.

Figure 1: Screenshot of GPTZero’s AI Vocabulary Released on October 7, 2024.

the first systematic study that assesses GPTZero’s AI Vocabulary feature in detecting LLM-generated content from educational contexts. Specifically, we integrate the AI Vocabulary lists (from October 2024 to March 2025) within supervised Bag-of-Words (BoW) classification models, namely two Naive Bayes classifiers trained on a subset of the Ghostbuster detector dataset (Verma et al., 2024), containing student and LLM-generated essays from ChatGPT (OpenAI, 2023) and Claude (Anthropic, 2023). We selected these models for their interpretability, as they allow us to inspect the contribution of each feature, namely the AI Vocabulary terms, to classification decisions. We position our work as a step toward evaluating the real-world utility of interpretable detection tools in educational contexts, where the use of AI is becoming increasingly widespread and, therefore, both reliable and efficient solutions are needed.

2 Background

Being able to differentiate between human-written and LLM-generated texts has recently become a much-discussed research topic, especially in academic and educational contexts. However, current AI detection methods present two main issues: (i) they often rely on non-transparent features, abstract and difficult for the average person to interpret and (ii) they have limited applicability for texts written by students, who are underrepresented in the training data.

Several current AI detection methods and sys-

tems prioritize model-based statistical metrics over basic linguistic features, such as perplexity (Vasilatos et al., 2023) and burstiness (Tian and Cui, 2025), log-probability (Solaiman et al., 2021) and high-dimensional neural representations (Guo et al., 2024). While highly performative, these methods do not offer interpretable justifications for their predictions, making it difficult for educators to reliably use and understand their outcomes (Ji et al., 2024). The underrepresentation of student texts in the detectors’ training sets represents another significant challenge. Student writing can exhibit lower fluency, formulaic phrasing or genre-specific traits that differ from both typical human and LLM-generated outputs. This mismatch can lead to high false positive rates, as observed in recent evaluations (Weber-Wulff et al., 2023; Liang et al., 2023; Perkins et al., 2024), as well as numerous false negatives, particularly when texts undergo simple adversarial modifications to evade the detectors (Weber-Wulff et al., 2023; Perkins et al., 2024).

In response to these AI detectors’ transparency issues, interpretable alternatives focusing on word frequency, n-gram patterns and stylometric indicators (Opara, 2024; Ciccarelli et al., 2024; Muñoz-Ortiz et al., 2024) have emerged to offer more transparent and pedagogically useful solutions. However, these methods are often less robust when applied to domain shifts or with LLM-generated texts modified to become less detectable. A hybrid detection tool, GPTZero (Tian and Cui, 2025), combines statistical features, such as perplexity and bursti-

ness, with more interpretable metrics, including readability, text complexity and average sentence length. Although it does not fully disclose the rationale behind its classifications, GPTZero claims to be “the top AI detector for teachers” (Tian and Cui, 2025). As such, it has recently introduced AI Vocabulary lists that highlight in text terms disproportionately used by LLMs compared to human authors, as a way to enhance interpretability and better support educational use among teachers.

Beyond AI detection models, some studies have recently emerged that focus on quantifying and analyzing a significant increase in the use of certain words and phrases, especially in scientific writing, after the introduction of LLMs. Kobak et al. (2024) employed large-scale corpus analysis of medical abstracts to track excess word usage and revealed a sharp rise in usually less frequent terms such as “delve” and “intricacies”. Juzek and Ward (2025) used model testing methods and human evaluators to explore why LLMs overrepresent certain terms, focusing on 21 “focal words”. However, their results turned out to be inconclusive. Mingmeng and Roberto (2024) quantitatively compared academic texts before and after the spread of LLMs, documenting a general trend towards producing more complex and abstract texts. Liang et al. (2024) analyzed textual features and metadata from papers across different domains, linking higher rates of LLMs use with texts whose first authors published more preprints, shorter papers or in more popular research fields.

While these studies provide interesting insights into possible LLM-influenced term choices, they mostly remain focused on quantitative and comparative vocabulary studies that analyze the language of scientific publications and do not directly extend to student writing or educational contexts. Moreover, to the best of our knowledge, no existing works have systematically leveraged terms more frequently used by LLMs to build and evaluate AI detection models focused on the educational domain, where they would currently be highly needed. This highlights a critical research gap, in which to explore whether vocabulary-based AI detection methods could be applied to distinguish student-written texts from LLM-generated ones, supporting educators in a more interpretable and linguistically justified manner. To address this gap, in this paper we evaluate for the first time GPTZero’s AI Vocabulary lists as a promising way to detect LLM-generated essays among student-written ones. To

the best of our knowledge, these are the only publicly available, extensive AI-vocabulary lists derived from a significant number of documents that go beyond scientific publications and likely include student-written texts, given GPTZero’s commitment to teachers and educational contexts². Moreover, they also provide data concerning the different word and phrase frequencies found in human-authored and LLM-generated texts (see Figure 1), further increasing their relevance. Starting with this study we aim to work towards developing a more transparent AI detection methodology applicable in educational contexts, reliable and better aligned with educators’ needs.

3 Method

3.1 GPTZero’s AI Vocabulary Lists

We collected the AI Vocabulary lists published on the GPTZero website between October 2024 and March 2025.³ Each month, we gathered a list of words and phrases together with their frequency estimates (see Appendix A), which had been estimated on 3.3 million texts (Tian and Cui, 2025). The October 2024 list featured the 50 most frequent AI-related terms, including single words and multi-word expressions. In November 2024, the same list (“Updated October 2024”) remained online. In December 2024, a new list (“Updated November 2024”), now including 99 items, was published.⁴ However, this updated list contained some errors, such as missing words, duplicate entries, and phrase variations. Subsequently, a corrected list with 100 items was published later on in December, 2024. This one still contained duplicates, so we removed exact double entries for the purpose of our experiments. In January 2025, a new list (“Updated January 2025”) with 100 unique phrases was published. No new list was published in February 2025; instead, the January 2025 list remained online for that month. The March 2025 list was labeled as “updated”, but it was identical to the January and February lists. As a result, there were only three distinct AI Vocabulary lists that we could use in our experiments: (a) the October 2024 list, (b) the November/December 2024 list, and (c) the January/February/March 2025 list. In addition, we constructed a *combined* list (“All”)

²<https://gptzero.me/educators>

³The reader can retrieve these lists using <https://web.archive.org/>.

⁴<https://web.archive.org/web/20241208223132/https://gptzero.me/ai-vocabulary>

that merged all unique words and phrases from these three sources.

3.2 Data

To detect LLM-generated essays using GPTZero’s AI Vocabulary, we used a subset of the data initially employed to train the Ghostbuster detector (Verma et al., 2024). This dataset originally contained 21,000 documents, including articles, creative writing pieces and student essays. For our experiments, we focus on a subset of 1,000 university student essays sourced from IvyPanda (IvyPanda, 2025), a platform where users can submit essays from high school and university levels concerning various topics and subjects, and 2,000 LLM-generated essays. To obtain the latter, Verma et al. (2024) used ChatGPT to generate the prompts corresponding to the unique 1,000 assignments based on which the student texts were written. These prompts were then used to generate 1,000 essays with ChatGPT and 1,000 essays with Claude. The desired essay length was also specified in them to match the human-written texts. The resulting median word count was 661 for student essays, 536 for ChatGPT-generated essays and 456 for Claude-generated essays. For the rest of the paper, we will refer to this subset of essays as the *Ghostbuster corpus*.

3.3 Models

We experimented with three classification models, each trained to predict whether an essay is written by a student or by (a) an LLM, (b) Claude, or (c) ChatGPT. To this end, we performed binary classification (using only the binary labels “AI” and “human”) on different dataset partitions: 1,000 student essays with (a) all 2,000 LLM-generated essays, (b) 1,000 essays generated by Claude, or (c) 1,000 essays generated by ChatGPT. We used *scikit-learn* (Pedregosa et al., 2011) to implement and train these classification models.

For each of these three classifiers, we estimated separate models for each of the four AI Vocabulary lists (monthly or combined), computing different feature vectors for the words and phrases in the list. For each list, we counted the occurrences of its items in the corpus using a Bag-of-Words (BoW) approach. Each AI word or phrase was treated as a distinct feature. Since the vocabulary included multi-word units (i.e., AI phrases), we employed an n -gram vectorization strategy to capture these phrases, setting n to range from 1 token up to the maximum number of tokens found in the longest

phrase in the list.⁵

We integrated the BoW features in a binary Naive Bayes classifier

$$P(c|w_1, \dots, w_n) \propto P(c) \prod_{i=1}^n P(w_i|c) \quad (1)$$

to predict the class c , namely whether an essay is generated by an AI (positive class) or written by a student (negative class). The models assumed independence between each word/phrase w and always used a uniform prior, assuming an equal chance (50%) that an essay belonged to one of the two⁶ classes.

We experimented with two types of feature vectors: (a) a Multinomial feature vector indicating the counts of the words and phrases in the essay, and (b) a Bernoulli feature vector indicating the presence or absence of the words and phrases in the essay.

For comparison, we also trained binary Naive Bayes classifiers – using either Multinomial or Bernoulli feature vectors – based on an alternative Bag-of-Words approach. In this configuration, the vocabulary comprised all unigram word types found in the Ghostbuster dataset, which were used to construct the feature vectors. These models served as a baseline to assess the effect of using the curated vocabulary lists in contrast to the default vocabulary derived directly from the training data.

3.4 Experiments

We trained a total of 24 classification models (3 AI x 4 lists x 2 features) using GPTZero AI Vocabulary lists, along with 6 reference models (3 AI x 2 features) based on the vocabulary derived from the Ghostbuster training data. To ensure an exhaustive evaluation, all models were trained and tested using leave-one-out cross-validation.

3.5 Metrics

We evaluated the classifiers’ performance using accuracy, (binary) precision, (binary) recall, (binary) F1-score, MCC (Matthews correlation coefficient) and AUROC (Area Under the Receiver Operating Characteristic curve) computed with *scikit-learn*

⁵This was implemented using `CountVectorizer`, with the `ngram_range` parameter set to `(1, max_phrase_length)`.

⁶It is important to reiterate that we did not perform any multiclass classification between the different LLMs in the dataset. We always compared LLM-generated to student-written, or Claude-generated to student-written, or GPT-generated to student-written (cf., *supra*).

(Pedregosa et al., 2011). Precision, recall and F1 score were computed for the positive class only (LLM-generated essays).

4 Results

4.1 GPTZero’s AI Vocabulary terms’ distribution in Ghostbuster

Table 1 lists the terms from GPTZero’s AI Vocabulary found in the Ghostbusters dataset. Of the 245 distinct words and phrases published between October 2024 and March 2025, only 98 appeared in the entire dataset, 53 in the Claude subset and 91 in the ChatGPT subset. These low and different distributions suggest that many AI-specific vocabulary terms identified by GPTZero as salient AI indicators, such as “left an indelible mark” (ranked 8th in Table 7 but only found 9 times in our corpus), “a rich tapestry” (ranked 18th in Table 7 but only found 6 times in our corpus), “offers valuable insights” (ranked 9th in Table 7 but only found 7 times in our corpus), “despite facing ” (ranked 3rd in Table 5 but only found 6 times in our corpus) and “study aims to explore” (ranked 6th in Table 5 but only found twice in our corpus) may not be frequently used in educational LLM-generated essays, in particular by models other than ChatGPT for most of the cases.

To assess the alignment between GPTZero’s AI Vocabulary rankings and their usage in LLM-generated essays, we calculated Spearman rank correlations between each term’s rank in the AI Vocabulary lists and its rank based on frequency of usage in LLM-generated texts (Claude and/or GPT). Our results (see Table 9) indicate generally weak or negative correlations between AI Vocabulary rankings and their occurrence across LLM-generated texts. There was, however, a significant positive correlation between the terms’ ranks in the October lists and their usage in Claude-generated texts ($\rho = 0.501$, $p < .001$), as well as between the terms’ ranks in the January-March lists and their usage in Claude-generated texts ($\rho = 0.476$, $p < .001$). In contrast, correlations with ChatGPT-generated texts remained low or negative, except for a modest positive correlation ($\rho = 0.211$, $p = .053$) with the November/December AI Vocabulary list⁷. Based on these findings, it is still

⁷These different correlation values suggest that while higher-ranked AI Vocabulary words tend to be relatively more frequent in Claude-generated essays compared to lower-ranked terms, their overall presence in such texts remains sparse.

unclear whether the actual ranks of the AI Vocabulary words and phrases in the list are informative and could consequently be used for AI text detection in education.

4.2 Classification performance with GPTZero’s AI Vocabulary lists

In our experiments, we evaluated two types of Naive Bayes classifiers, one using a Bernoulli feature vector and one using a Multinomial feature vector, based on GPTZero’s AI Vocabulary lists (from October 2024 to March 2025). We tested the classifiers in detecting AI-generated essays both individually, with each monthly AI Vocabulary list, and with a combined list containing 245 AI Vocabulary terms from all months. We evaluated both the full Ghostbuster essays corpus and subsets specific to Claude- and ChatGPT-generated texts.

Overall, classification results were close to random, with accuracy ranging from 0.363 to 0.755 for Bernoulli models and 0.363 to 0.729 for Multinomial models (see Table 2) using the different AI Vocabulary lists. However, we found more promising results when focusing specifically on ChatGPT-generated texts using the combined GPTZero’s AI Vocabulary lists of all months. Here, the Bernoulli model achieved the highest accuracy (0.755), high precision (0.882), moderate recall (0.588) and an F1 score of 0.705, which indicates good performance in identifying LLM-generated texts, although it might have missed some positive cases. The high precision score signals that the model does not make numerous false predictions causing it to mislabel student texts as AI-generated (a significant risk in educational contexts as highlighted by Liang et al. 2023). However, the moderate recall also indicates that the model’s sensitivity should increase in order to avoid some LLM-generated texts to go undetected (also particularly relevant in educational contexts as stressed by Fleckenstein et al. 2024; Weber-Wulff et al. 2023; Perkins et al. 2024). An MCC score of 0.541 supports our interpretation and an AUROC of 0.595 suggests that the model, despite being better than random, may struggle in more ambiguous cases. The Multinomial model, using the same feature set, yielded higher precision (0.884) and AUROC (0.705), indicating higher sensitivity to ChatGPT-generated content and a more balanced classification ability. This may be due to an overrepresentation of ChatGPT-generated texts in the datasets used by GPTZero to compile the AI Vocabulary lists. However, this model also reached

	All	C	G		All	C	G
add an extra layer	1	0	1	meticulous attention to	4	0	4
add depth to	1	0	1	meticulously crafted	2	1	1
address issues like	1	1	0	navigate challenges	2	2	0
advancements	204	42	189	navigate the complex	6	0	6
aiding	19	3	18	offer valuable insights	8	0	8
aim to explore	1	1	0	offers numerous benefits	3	0	3
aims to enhance	3	1	2	offers valuable	12	1	12
aligns	78	25	66	offers valuable insights	7	0	7
an unwavering commitment	1	0	1	potentially leading	15	0	15
commitment to excellence	2	0	2	prioritize	247	39	227
consider factors like	1	1	0	prioritizing	71	17	62
continue to inspire	4	0	4	provide an insight	1	1	1
contribute to understanding	1	0	1	provide valuable insights	20	5	17
crucial role in shaping	34	1	33	provided valuable	6	3	3
crucial role in understanding	1	0	1	provided valuable insights	4	2	2
delve deeper	4	1	4	provides valuable	36	10	28
delve deeper into	4	1	4	provides valuable insights	24	5	20
despite facing	6	4	2	providing insights	2	1	2
emphasize the need	7	2	7	relentless pursuit	4	0	4
enduring legacy	3	0	3	remarked	3	3	2
ensure long term success	2	0	2	researchers aim	1	1	0
essential to recognize	19	0	19	researchers aimed	3	0	3
explores themes	3	0	3	rich tapestry	6	0	6
findings shed	1	0	1	sense of camaraderie	8	0	8
findings shed light	1	0	1	showcasing	52	8	49
fostering	249	23	236	significant advancements	8	1	8
fostering sense	23	0	23	sparking	5	1	4
gain comprehensive understanding	10	1	9	standout	7	1	7
gain deeper	32	5	27	stark reminder	9	1	8
gain deeper insights	1	0	1	struggles faced	15	0	15
gain deeper understanding	28	5	23	study aims to explore	2	2	0
gain valuable	23	6	17	study highlights the importance	1	1	0
gain valuable insights	17	1	16	study provides valuable	2	0	2
garnered significant	1	0	1	study sheds	1	0	1
highlight the need	3	1	2	study sheds light	1	0	1
highlight the potential	1	0	1	surpassing	9	6	7
highlight the significance	7	0	7	the complex interplay	8	7	1
highlighting the need	3	2	1	the multifaceted nature	12	1	11
hindering	47	7	47	the potential to revolutionize	10	0	10
holds significant	10	1	9	the relentless pursuit	1	0	1
impacting	62	31	45	the transformative power	9	2	7
indelible mark	11	0	11	tragically	6	4	4
indicating potential	1	0	1	underscore the importance	1	0	1
intricate relationship	3	0	3	understand the behavior	1	0	1
left an indelible mark	9	0	9	understand the complexity	2	0	2
left lasting	11	4	7	unwavering commitment	2	0	2
let delve	7	0	7	unwavering support	1	1	1
making it challenging	14	2	14	valuable insights	115	21	99
marked significant	4	0	4	vital role in shaping	9	0	9

Table 1: GPTZero’s AI words/phrases with their counts in Ghostbusters (All), Claude (C), and GPT (G) subsets.

Features	LLM	Vocabulary	Accuracy	Precision	Recall	F1	MCC	AUROC
Bernoulli	All	GPTZero List: All	0.532	0.884	0.343	0.494	0.272	0.362
		GPTZero List: Oct	0.503	0.877	0.296	0.443	0.240	0.292
		GPTZero List: Nov/Dec	0.416	0.996	0.129	0.228	0.199	0.135
		GPTZero List: Jan/Feb/Mar	0.363	0.969	0.046	0.089	0.117	0.050
		Ghostbuster BoW	0.871	0.846	0.986	0.911	0.711	0.948
	Claude	GPTZero List: All	0.522	0.657	0.09	0.158	0.085	0.156
		GPTZero List: Oct	0.502	0.501	0.968	0.660	0.011	0.107
		GPTZero List: Nov/Dec	0.508	0.786	0.022	0.043	0.068	0.033
		GPTZero List: Jan/Feb/Mar	0.503	1.0	0.007	0.014	0.059	0.017
		Ghostbuster BoW	0.889	0.825	0.987	0.899	0.793	0.975
	GPT	GPTZero List: All	0.755	0.882	0.588	0.705	0.541	0.595
		GPTZero List: Oct	0.703	0.853	0.49	0.622	0.448	0.495
		GPTZero List: Nov/Dec	0.616	0.964	0.242	0.386	0.351	0.250
GPTZero List: Jan/Feb/Mar		0.544	0.968	0.092	0.167	0.209	0.102	
Ghostbuster BoW		0.929	0.892	0.977	0.933	0.862	0.990	
Multinomial	All	GPTZero List: All	0.517	0.891	0.314	0.464	0.263	0.604
		GPTZero List: Oct	0.452	0.910	0.197	0.324	0.212	0.550
		GPTZero List: Nov/Dec	0.410	0.968	0.119	0.213	0.191	0.549
		GPTZero List: Jan/Feb/Mar	0.363	0.969	0.046	0.089	0.117	0.518
		Ghostbuster BoW	0.901	0.955	0.893	0.923	0.787	0.957
	Claude	GPTZero List: All	0.518	0.673	0.072	0.130	0.082	0.517
		GPTZero List: Oct	0.504	0.538	0.064	0.114	0.019	0.506
		GPTZero List: Nov/Dec	0.508	0.786	0.022	0.043	0.068	0.509
		GPTZero List: Jan/Feb/Mar	0.503	1.0	0.007	0.014	0.059	0.503
		Ghostbuster BoW	0.964	0.976	0.951	0.964	0.928	0.991
	GPT	GPTZero List: All	0.729	0.884	0.527	0.660	0.501	0.705
		GPTZero List: Oct	0.654	0.895	0.350	0.503	0.390	0.597
		GPTZero List: Nov/Dec	0.604	0.964	0.216	0.353	0.330	0.589
GPTZero List: Jan/Feb/Mar		0.539	0.964	0.081	0.149	0.194	0.530	
Ghostbuster BoW		0.912	0.942	0.877	0.909	0.825	0.953	

Table 2: Performance of classifiers on leave-one-out cross-validation. The highest accuracy values are indicated in boldface.

lower accuracy (0.729) and F1 score (0.660), making it less reliable.

These results suggest that binary-feature BoW models like Bernoulli may be more effective at detecting ChatGPT-generated texts based solely on AI-related terms’ presence, while frequency-based models like Multinomial may be better at identifying subtler vocabulary usage patterns. Finally, both Naive Bayes classifiers were significantly outperformed by a baseline Multinomial BoW classifier trained on the full vocabulary of the Ghostbuster dataset. This model achieved a maximum accuracy of 0.964 and an AUROC score of 0.991 (see Table 2) with Claude texts - differing from the previous highest results for ChatGPT-generated essays using the AI Vocabulary lists, possibly more effective given the absence or scarcity of Claude’s generated data for the compilation of the AI Vo-

cabulary lists⁸. This highlights the limitations of relying on fixed AI Vocabulary lists for AI detection, which might not reflect the language found in educational essays written by different LLMs and students.

4.3 AI Vocabulary terms contribution to classification

To better understand which specific AI Vocabulary terms influenced classification, we analyzed their log probabilities under our best-performing Naive Bayes models, namely the Bernoulli and Multinomial variants that achieved the highest classification results. These models were trained using the subset of ChatGPT-generated texts from the Ghostbuster corpus and the full combined AI Vocabulary list (with 245 terms from October 2024 to March

⁸See GPTZero’s support article <https://support.gptzero.me/hc/en-us/articles/15129377479959> for more

2025).

Phrase	Rank	Freq.	Count	OR	LP
fostering	245	9	236	11.88	-2.15
prioritize	238	11	227	8.54	-2.22
advancements	243	9	189	3.91	-2.65
valuable in-sights	19	230	99	5.53	-2.93
prioritizing	241	9	62	2.71	-3.43
aligns	234	17	66	2.19	-3.48
showcasing	232	21	49	4.31	-3.62
hindering	242	9	47	2.64	-3.74
crucial role in shaping	37	155	33	10.53	-3.83
impacting	237	12	45	2.07	-3.89
gain deeper	86	98	27	5.39	-3.97
provides	117	86	28	2.28	-4.00
valuable gain deeper understanding	50	131	23	2.82	-4.13
provides	4	464	20	1.33	-4.27
valuable insights					
essential to recognize	213	48	19	6.56	-4.27
fostering	45	138	23	2.11	-4.27
sense					
gain valuable	100	92	17	1.99	-4.43
gain	175	59	16	1.64	-4.49
valuable insights					
potentially	231	43	15	5.81	-4.55
leading					
aiding	244	9	18	3.28	-4.55
provide	7	332	17	1.32	-4.62
valuable insights					
struggles	224	46	15	5.93	-4.70
faced					
making it	142	74	14	3.00	-4.70
challenging					
indelible	11	275	11	2.33	-4.78
mark					
the potential to revolutionize	123	83	10	4.34	-4.86
offers valuable	128	81	12	1.23	-4.96

Table 3: Top 25 phrases contributing to LLM-generated text detection, ordered by log-probability from the best Bernoulli Naive Bayes classifier using all AI Vocabulary lists on ChatGPT-generated essays. The *Rank* and *Frequency* columns relate to the combined GPTZero’s AI Vocabulary lists (245 phrases from October 2024 to March 2025), *Count* refers to the frequency in ChatGPT-generated texts, *OR* refers to the odds ratio in LLM-generated vs. human-authored texts and *LP* represents log probability of the phrase contribution to classification.

For both models, Bernoulli and Multinomial, the top 25 terms that contributed the most to classifi-

Phrase	Rank	Freq.	Count	OR	LP
fostering	245	9	236	7.49	-2.00
prioritize	238	11	227	5.49	-2.08
advancements	243	9	189	2.44	-2.33
valuable in-sights	19	230	99	4.76	-2.87
prioritizing	241	9	62	2.75	-3.42
aligns	234	17	66	2.05	-3.44
showcasing	232	21	49	4.14	-3.62
hindering	242	9	47	2.59	-3.71
crucial role in shaping	37	155	33	9.57	-3.90
impacting	237	12	45	2.13	-3.96
gain deeper	86	98	27	5.33	-4.09
provides	117	86	28	2.36	-4.13
valuable					
fostering	45	138	23	1.95	-4.25
sense					
gain deeper understanding	50	131	23	2.81	-4.25
essential to recognize	213	48	19	6.38	-4.43
provides	4	464	20	1.31	-4.43
valuable insights					
gain valuable	100	92	17	1.92	-4.54
aiding	244	9	18	3.15	-4.59
gain valuable	175	59	16	1.68	-4.59
insights					
provide valuable insights	7	332	17	1.22	-4.65
potentially	231	43	15	5.58	-4.65
leading					
struggles	224	46	15	5.41	-4.65
faced					
making it	142	74	14	3.02	-4.86
challenging					
offers valuable	128	81	12	1.28	-4.94
indelible	11	275	11	2.27	-4.94
mark					
the multifaceted nature	98	92	11	3.12	-4.94

Table 4: Top 25 phrases contributing to LLM-generated text detection, ordered by log-probability from the best Multinomial Naive Bayes classifier using all AI Vocabulary lists on ChatGPT-generated essays. The *Rank* and *Frequency* columns relate to the combined GPTZero’s AI Vocabulary lists (245 phrases from October 2024 to March 2025), *Count* refers to the frequency in ChatGPT-generated texts, *OR* refers to the odds ratio in LLM-generated vs. human-authored texts and *LP* represents log probability of the phrase contribution to classification.

cation were largely the same, although in slightly different order (see Table 3 and Table 4). Each table includes the terms’ original *Rank* and *Frequency* in the combined AI Vocabulary list, their *Count* in ChatGPT-generated texts, the *OR* (indicating their relative likelihood in LLM vs. human text based

on odds ratios) and the models' log probabilities, LP (reflecting the terms' contribution to the model decision; lower values imply weaker impact).

We noticed in the Bernoulli and Multinomial classifiers that several words and phrases found in numerous ChatGPT-generated texts, such as “fostering” (counted 236 times), “prioritize” (227), “advancements” (62), “aligns” (66) and “showcasing” (49), despite being found more frequently in Ghostbuster's texts than in GPTZero's ranking lists, were less effective in distinguishing AI-generated from student-authored essays given their low log probabilities. Similarly, when considering terms that were highly ranked and common in AI Vocabulary lists, such as “provides valuable” (4th), “provides valuable insights” (7th), “indelible mark” (11th) and “valuable insights” (19th), we observed also low log probabilities, apart from recurring phrases, meaning that they did not significantly contribute to classification.

We decided to maintain separate entries for morphological variants rather than indexing them together to investigate whether certain preferences exist in LLM-generated texts. In this way, we could check if verb tense, number, or grammatical person can also influence AI-generated text detection. By maintaining distinctions such as “provide” vs. “provides” (valuable insights) and “study shed” vs. “study sheds” we can evaluate whether specific variants display distributional biases in LLM-generated texts compared to student-authored essays. However, if these differences prove insignificant, as seems to be the case in our experiments, in future works we could consider lemmatization or stemming.

Our findings are in line with our previous observations, provided in Section 4.1, where term rankings and frequencies did not seem to notably support classification. They, nevertheless, confirm our classification results described in Section 4.2, highlighting the strengths of the BoW Bernoulli model over the Multinomial one, accounting for the terms' presence only, rather than for their frequency, to better distinguish LLM-generated texts from student-written ones.

5 Conclusion

In this study, we presented the first empirical evaluation of GPTZero's AI Vocabulary lists as a way to detect AI-generated texts in educational settings. Our findings show that these precompiled vocabu-

lary lists, despite being transparent and easily interpretable for educators, have limited effectiveness in detecting LLM-generated texts among educational essays, especially beyond ChatGPT. Even for ChatGPT-generated texts, the classification performance of our Naive Bayes models based on AI Vocabulary lists was modest and only improved when using a combined list of 245 terms. We achieved better results with BoW models that used the full Ghostbuster dataset vocabulary, suggesting that broader language patterns may be more effective for AI detection with different LLMs.

Future research should focus on a deeper, more domain-specific analysis and comparison between student and LLM-generated texts in educational domains, including more diverse student samples and LLM-generated texts. Vocabulary-based AI detectors could benefit from the inclusion of additional functional and structural features, considering each term and phrase as linguistic constructions that reflect users' language more in detail.

Overall, although our results might not come close to state-of-the-art detectors, with this work we addressed a key research gap. To the best of our knowledge, no prior study has evaluated precompiled AI Vocabulary lists, publicly available and derived from a diverse set of texts beyond scientific articles, for AI detection in education. Our findings offer practical and detailed insights into the utility and accuracy of *transparent* linguistic features, such as AI Vocabulary lists, that can support educators in distinguishing LLM-generated and student-written texts. By doing so, this work contributes to the ongoing efforts to improve AI detection systems and lays a foundation for further investigation and refinement in educational contexts.

Limitations

Although our work provides useful evidence in the analysis of GPTZero's AI Vocabulary lists for AI detection, there are several limitations that need to be accounted for. First, we only tested two Bag-of-Words classifiers (Bernoulli and Multinomial) using a Naive Bayes approach. These are relatively simple models. More advanced machine learning and neural approaches could help to expand the testing framework and potentially improve detection accuracy. Second, the dataset used in this study, the Ghostbuster essay subcorpus, represents outputs from older versions of ChatGPT

and Claude models. As new model versions are released, the vocabulary patterns of current AI systems may differ significantly. Moreover, due to the lack of metadata, we assume students to be native English speakers. Future studies should examine L2 students, who may rely more on LLMs and for whom current detectors might be less effective. Third, as LLMs continue to evolve, their outputs become closer to human language, making fixed vocabulary lists less effective over time. To remain useful, these AI Vocabulary lists would need to be updated more frequently and adapted across different writing domains, to reflect changes in language use.

Acknowledgments

The authors would like to thank Piet Desmet for for his insightful comments and feedback. The research presented in this paper was funded by the Research Foundation Flanders (FWO) through a doctoral fellowship awarded to Veronica Juliana Schmalz (1108723N).

References

- Anthropic. 2023. [Claude: An AI assistant](#).
- Vittorio Ciccarelli, Cornelia Genz, Nele Mastracchio, Wiebke Petersen, Anna Stein, and Hanxin Xia. 2024. [Team art-nat-HHU at SemEval-2024 Task 8: Stylistically Informed Fusion Model for MGT-Detection](#). In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1690–1697, Mexico City, Mexico. Association for Computational Linguistics.
- Tor Constantino. 2024. [New List Ranks AI's 50 Most Overused Words - Updates Monthly](#). Accessed on March 22, 2025.
- Deborah R. E. Cotton, Pauline A. Cotton, and James R. Shipway. 2024. [Chatting and cheating: Ensuring academic integrity in the era of ChatGPT](#). *Innovations in Education and Teaching International*, 61(2):228–239.
- Johanna Fleckenstein, Jennifer Meyer, Thorben Jansen, Stefan D. Keller, Olaf Köller, and Jens Möller. 2024. [Do teachers spot AI? Evaluating the detectability of AI-generated texts among student essays](#). *Computers and Education: Artificial Intelligence*, 6:100209.
- Xun Guo, Yongxin He, Shan Zhang, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024. [Detective: Detecting AI-Generated Text via Multi-Level Contrastive Learning](#). *Advances in Neural Information Processing Systems*, 37:88320–88347.
- IvyPanda. 2025. [Free Essay Examples and Writing Resources](#). <https://ivypanda.com/>. Accessed: March 10, 2025.
- Jiazhou Ji, Ruizhe Li, Shujun Li, Jie Guo, Weidong Qiu, Zheng Huang, Chiyu Chen, Xiaoyu Jiang, and Xinru Lu. 2024. [Detecting Machine-Generated Texts: Not Just "AI vs Humans" and Explainability is Complicated](#). *arXiv preprint*, 2406:18259.
- Tom S. Juzek and Zina B. Ward. 2025. [Why Does ChatGPT "Delve" So Much? Exploring the Sources of Lexical Overrepresentation in Large Language Models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6397–6411, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Dmitry Kobak, Rita González-Márquez, Emőke Ágnes Horvát, and Jan Lause. 2024. [Delving into ChatGPT Usage in Academic Writing through Excess Vocabulary](#). *arXiv*, 2406(07016).
- Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. [GPT detectors are biased against non-native English writers](#). *Patterns*, 4(7):100779.
- Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D Manning, and James Y. Zou. 2024. [Mapping the Increasing Use of LLMs in Scientific Papers](#). In *Proceedings of the COLM 2024 Conference*.
- Geng Mingmeng and Trotta Roberto. 2024. [Is ChatGPT Transforming Academics' Writing Style?](#) *arXiv preprint*, 2404(08627).
- Alberto Muñoz-Ortiz, Carlos Gómez-Rodríguez, and David Vilares. 2024. [Contrasting Linguistic Patterns in Human and LLM-Generated News Text](#). *Artificial Intelligence Review*, 57(10):265.
- Chidimma Opara. 2024. [StyloAI: Distinguishing AI-generated Content with Stylometric Analysis](#). In *Proceedings of the International Conference on Artificial Intelligence in Education*, pages 105–114. Springer.
- OpenAI. 2023. [Chatgpt \(gpt-4\)](#).
- 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.
- Mike Perkins, Jasper Roe, Darius Postma, James Mc-Gaughran, and Don Hickerson. 2024. [Detection of GPT-4 generated text in higher education: Combining academic judgement and software to identify generative AI tool misuse](#). *Journal of Academic Ethics*, 22(1):89–113.

- Basel Solaiman, Didier Guériot, Shaban Almouahed, Bassem Alsahwa, and Éloi Bossé. 2021. [A New Hybrid Possibilistic-Probabilistic Decision-Making Scheme for Classification](#). *Entropy*, 23(1):67.
- Edward Tian and Alexander Cui. 2025. [GPTZero: Towards detection of AI-generated text using zero-shot and supervised methods](#).
- Christoforos Vasilatos, Manaar Alam, Talal Rahman, Yasir Zaki, and Michail Maniatakos. 2023. [HowKGPT: Investigating the Detection of ChatGPT-Generated University Student Homework through Context-Aware Perplexity Analysis](#). *arXiv preprint*, 2305:18226.
- Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. ["ghostbuster: Detecting text ghostwritten by large language models"](#). 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 1702–1717. Association for Computational Linguistics.
- Talia Waltzer, Cecile Pilegard, and Gail D. Heyman. 2024. [Can you spot the bot? Identifying AI-generated writing in college essays](#). *International Journal for Educational Integrity*, 20(1):11.
- Debora Weber-Wulff, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltýnek, Jean Guerrero-Dib, Olu-mide Popoola, Petr Šigut, and Lorna Waddington. 2023. [Testing of detection tools for AI-generated text](#). *International Journal for Educational Integrity*, 19(1):1–39.
- Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia S. Chao, and Derek F. Wong. 2025. [A survey on LLM-generated text detection: Necessity, methods, and future directions](#). *Computational Linguistics*, pages 1–66.

A GPTZero’s AI Vocabulary Lists

This appendix contains lists of the top AI words and phrases from GPTZero, spanning from October 2024 to March 2025. Each monthly list includes frequently used AI-related terms, along with their frequency estimates. The November list was initially identical to the October list with 50 entries, but an update appeared in December with 99 entries. However, this updated version contained errors such as missing determiners and prepositions (e.g., *crucial role understanding* instead of *a crucial role in understanding*) and incongruencies, including duplicate entries (e.g., 4) *provide valuable insights* - 464 and 84) *provide valuable insights* - 86). This list also contains numerous variations of the same phrase (e.g., 13) *plays a crucial role in understanding* - 247 and 14) *play a crucial role in understanding*- 242) and longer phrases that are part of other shorter phrases, also appearing in the list (e.g., 24) *plays a crucial role in shaping* - 178 and 26) *crucial role in shaping* - 171). The January, February and March lists were identical, so we report them in the same table. The frequency estimates indicate how many times more frequently a term appears in AI-written texts compared to human-written texts. For example, a term with a frequency estimate of 10 means it is ten times more common in AI texts than in human texts, based on a collection of 3.3 million documents (Tian and Cui, 2025).

	Phrase	Freq.
1	objective study aimed	269
2	research needed to understand	235
3	despite facing	209
4	play significant role shaping	182
5	crucial role in shaping	155
6	study aims to explore	144
7	notable works include	121
8	consider factors like	121
9	today’s fast paced world	107
10	expressed excitement	93
11	highlights importance considering	89
12	emphasizing importance	74
13	making it challenging	74
14	aims to enhance	72
15	study sheds light	69
16	emphasizing need	68
17	today’s digital age	68
18	explores themes	66
19	address issues like	65
20	highlighting the need	63
21	study introduce	60
22	notable figures	59
23	gain valuable insights	59
24	showing promising results	59
25	media plays a significant role	57
26	shared insights	56
27	ensure long term success	55
28	make a positive impact on the world	55
29	facing criticism	52
30	providing insights	49
31	emphasized importance	48
32	indicating potential	47
33	struggles faced	46
34	secured win	46
35	secure win	44
36	potentially leading	43
37	showcasing	21
38	remarked	18
39	aligns	17
40	surpassing	12
41	tragically	12
42	impacting	12
43	prioritize	11
44	sparkling	11
45	standout	11
46	prioritizing	9
47	hindering	9
48	advancements	9
49	aiding	9
50	fostering	9

Table 5: GPTZero’s Top AI Words and Phrases for October 2024

#	Phrase	Freq.	#	Phrase	Freq.
1	provided valuable insights	902	51	provided valuable insights	113
2	gain valuable insights	739	52	mix fear	109
3	casting long shadows	561	53	crucial role maintaining	106
4	provides valuable insights	464	54	serves reminder	106
5	gain comprehensive understanding	355	55	voice dripping	106
6	study provides valuable	340	56	gain deeper insights	104
7	provide valuable insights	332	57	insights potential	101
8	left indelible mark	319	58	significant advancement	100
9	offers valuable insights	298	59	researchers aimed	100
10	indelible mark	275	60	significant advancements	98
11	unwavering commitment	256	61	gain deeper	98
12	play crucial role shaping	250	62	began voice	98
13	plays crucial role understanding	247	63	findings shed	97
14	played significant role shaping	239	64	study provide valuable	96
15	left indelible	231	65	plays crucial role regulating	96
16	valuable insights	230	66	left lasting	96
17	rich tapestry	227	67	sense camaraderie	94
18	offer valuable insights	207	68	potential revolutionize	94
19	opens new avenues	206	69	navigate challenges	94
20	help feel sense	197	70	voice surprisingly	92
21	adds layer complexity	194	71	gain valuable	92
22	significant contributions field	188	72	understanding behavior	91
23	plays crucial role shaping	178	73	delve deeper	91
24	research needed explore	171	74	plays crucial role ensuring	91
25	crucial role shaping	171	75	relentless pursuit	90
26	intricate relationship	165	76	significant role shaping	88
27	findings contribute	157	77	researchers aim	88
28	continue inspire	152	78	meticulously crafted	88
29	stark reminder	151	79	study shed light	87
30	hung heavy	147	80	dripping sarcasm	87
31	crucial role understanding	139	81	aims shed light	87
32	fostering sense	138	82	voice rising	87
33	significant attention recent years	136	83	provides valuable	86
34	needed fully understand	133	84	play significant role shaping	85
35	pivotal role shaping	131	85	renewed sense purpose	85
36	gain deeper understanding	131	86	marked significant	85
37	study sheds light	130	87	enduring legacy	84
38	continues inspire	129	88	offers numerous benefits	84
39	implications various	129	89	commitment excellence	83
40	highlights importance considering	124	90	study shed	83
41	let delve	123	91	plays crucial role determining	83
42	holds significant	121	92	significant attention recent	83
43	study sheds	120	93	offers valuable	81
44	garnered significant	120	94	plays significant role shaping	79
45	advancing understanding	119	95	play crucial role determining	78
46	voice dripping sarcasm	119	96	despite chaos	78
47	conclusion study provides	117	97	paving way future	77
48	findings shed light	116	98	highlights significance	77
49	commitment public service	116	99	locals visitors alike	77

Table 6: GPTZero's Top AI Words and Phrases for November 2024 (*first version with repetitions and errors*)

#	Phrase	Freq.	#	Phrase	Freq.
1	provided valuable insights	902	51	provided valuable	113
2	gain valuable insights	739	52	mix the fear	109
3	casting long shadows	561	53	crucial role in maintaining	106
4	provides valuable insights	464	54	serves a reminder	106
5	gain comprehensive understanding	355	55	voice is dripping	106
6	study provides valuable	340	56	gain a deeper insights	104
7	provide valuable insights	332	57	insights into the potential	101
8	left an indelible mark	319	58	a significant advancement	100
9	offers valuable insights	298	59	the researchers aimed	100
10	an indelible mark	275	60	significant advancements	98
11	an unwavering commitment	256	61	gain a deeper	98
12	play a crucial role in shaping	250	62	began to voice	98
13	plays a crucial role in understanding	247	63	findings shed light on	97
14	play a crucial role in understanding	242	64	study provides valuable	96
15	played a significant role in shaping	239	65	plays a crucial role in regulating	96
16	left an indelible	231	66	left a lasting	96
17	valuable insights	230	67	sense of camaraderie	94
18	a rich tapestry	227	68	potential to revolutionize	94
19	offer valuable insights	207	69	navigate the challenges	94
20	opens new avenues	206	70	the voice surprisingly	92
21	help to feel a sense	197	71	gain a valuable	92
22	adds a layer of complexity	194	72	understanding the behavior	91
23	significant contributions to the field	188	73	delve deeper into	91
24	plays a crucial role in shaping	178	74	plays a crucial role in ensuring	91
25	research needed to explore	171	75	relentless pursuit	90
26	crucial role in shaping	171	76	significant role in shaping	88
27	the intricate relationship	165	77	researchers aim to	88
28	findings contribute to	157	78	meticulously crafted	88
29	continue to inspire	152	79	study shed light on	87
30	a stark reminder	151	80	dripping with sarcasm	87
31	hung heavy	147	81	aims to shed light	87
32	crucial role in understanding	139	82	voice is rising	87
33	fostering sense	138	83	provides valuable insights	86
34	significant attention in recent years	136	84	play a significant role in shaping	85
35	needed to fully understand	133	85	renewed sense of purpose	85
36	pivotal role in shaping	131	86	marked a significant	85
37	gain a deeper understanding	131	87	an enduring legacy	84
38	study sheds light on	130	88	offers numerous benefits	84
39	continues to inspire	129	89	commitment to excellence	83
40	implications of various	129	90	study shed light	83
41	highlights the importance of considering	124	91	plays a crucial role in determining	83
42	let us delve	123	92	significant attention in recent	83
43	holds a significant	121	93	offers a valuable	81
44	study sheds light on	120	94	plays a significant role in shaping	79
45	garnered significant	120	95	play a crucial role in determining	78
46	advancing the understanding	119	96	despite the chaos	78
47	voice dripping with sarcasm	119	97	paving the way for the future	77
48	conclusion of the study provides	117	98	highlights the significance	77
49	findings shed light on	116	99	locals and visitors alike	77
50	commitment to public service	116			

Table 7: GPTZero’s Top AI Words and Phrases for November 2024 (*corrected version published in December 2024*)

#	Phrase	Freq.	#	Phrase	Freq.
1	provide a valuable insight	468	51	understand the behavior	61
2	left an indelible mark	317	52	broad implications	61
3	play a significant role in shaping	207	53	a prominent figure	61
4	an unwavering commitment	202	54	study highlights the importance	60
5	open a new avenue	174	55	a significant turning point	60
6	a stark reminder	166	56	curiosity piques	59
7	play a crucial role in determining	151	57	today in the digital age	59
8	finding a contribution	139	58	implication to understand	59
9	crucial role in understanding	135	59	a beacon of hope	58
10	finding a shed light	121	60	pave the way for the future	58
11	gain a comprehensive understanding	120	61	finding an important implication	57
12	conclusion of the study provides	119	62	understand the complexity	57
13	a nuanced understanding	115	63	meticulous attention to	57
14	hold a significant	114	64	add a layer	57
15	gain significant attention	107	65	the legacy of life	56
16	continue to inspire	105	66	identify the area of improvement	56
17	provide a comprehensive overview	104	67	aim to explore	56
18	finding the highlight the importance	99	68	highlight the need	55
19	endure a legacy	99	69	provide the text	55
20	mark a significant	96	70	conclusion of the study demonstrates	55
21	gain a deeper understanding	95	71	a multifaceted approach	55
22	the multifaceted nature	92	72	provide a framework to understand	55
23	the complex interplay	89	73	present a unique challenge	55
24	study shed light on	89	74	highlight the significance	54
25	need to fully understand	88	75	add depth to	54
26	navigate the complex	87	76	a significant stride	53
27	a serf reminder	85	77	gain an insight	53
28	the potential to revolutionize	83	78	underscore the need	52
29	the relentless pursuit	79	79	the importance to consider	52
30	offer a valuable	77	80	offer a unique perspective	52
31	underscore the importance	76	81	contribute to understanding	52
32	a complex multifaceted	74	82	a significant implication	52
33	the transformative power	74	83	despite the challenge faced	52
34	today the fast pace of the world	74	84	enhances the understanding	51
35	a significant milestone	73	85	make an informed decision in regard to	50
36	delve deeper into	72	86	the target intervention	50
37	provide an insight	71	87	require a careful consideration	49
38	navigate the challenge	71	88	essential to recognize	48
39	highlight the potential	69	89	validate the finding	48
40	pose a significant challenge	69	90	vital role in shaping	47
41	a unique blend	68	91	sense of camaraderie	47
42	a crucial development	68	92	influence various factors	47
43	various fields include	67	93	make a challenge	46
44	commitment to excellence	65	94	unwavering support	46
45	sent shockwaves through	65	95	importance of the address	46
46	emphasize the need	65	96	a significant step forward	46
47	despite the face	65	97	add an extra layer	45
48	understanding the fundamental	64	98	address the root cause	44
49	leave a lasting	63	99	a profound implication	44
50	gain a valuable	62	100	contributes to understanding	44

Table 8: GPTZero's Top AI Words and Phrases from January 2025 to March 2025

B Spearman ranking correlation

In this appendix, we present the Spearman rank correlations between the term rankings in each AI Vocabulary list (*October*, *Nov/Dec*, *Jan/Feb/Mar* and *All*) and the rankings of the same terms based on their frequency in the Ghostbuster corpus, considering the ChatGPT (*GPT*) and Claude (*Claude*) subsets separately, as well as the entire dataset (*All*).

AI Vocabulary	LLM	ρ	p
All	All	-0.064	0.316
Oct	All	-0.149	0.299
Nov/Dec	All	0.065	0.529
Jan/Feb/Mar	All	0.124	0.219
All	Claude	-0.170	0.007
Oct	Claude	0.501	0.000
Nov/Dec	Claude	0.045	0.660
Jan/Feb/Mar	Claude	0.476	0.000
All	GPT	-0.095	0.135
Oct	GPT	-0.243	0.088
Nov/Dec	GPT	0.211	0.039
Jan/Feb/Mar	GPT	-0.010	0.914

Table 9: Spearman ranking correlation coefficients and p -values between GPTZero’s AI Vocabulary terms and the odds ratios of those terms in LLM-generated terms from the Ghostbuster dataset (*All*, *Claude* or *GPT* only).

Paragraph-level Error Correction and Explanation Generation: Case Study for Estonian

Martin Vainikko¹, Taavi Kamarik², Karina Kert², Krista Liin¹,
Silvia Maine², Kais Allkivi², Annekatrin Kaivapalu³, Mark Fishel¹,

¹Institute of Computer Science, University of Tartu;

²School of Digital Technologies, Tallinn University;

³Department of Finnish, Finno-Ugrian and Scandinavian Studies, University of Helsinki

Correspondence: martin.vainikko@ut.ee, taavi.kamarik@tlu.ee, karina.kert@tlu.ee, krista.liin@ut.ee, silvia.maine@tlu.ee, kais.allkivi@tlu.ee, annekatrin.kaivapalu@helsinki.fi, mark.fisel@ut.ee

Abstract

We present a case study on building task-specific models for grammatical error correction and explanation generation tailored to learners of Estonian. Our approach handles whole paragraphs instead of sentences and leverages prompting proprietary large language models for generating synthetic training data, addressing the limited availability of error correction data and the complete absence of correction justification/explanation data in Estonian. We describe the chosen approach and pipeline and provide technical details for the experimental part. The final outcome is a set of open-weight models, which are released with a permissive license along with the generated synthetic error correction and explanation data.

1 Introduction

Language models with emergent abilities are increasingly showing capacity for performing natural language processing tasks via prompting (OpenAI et al., 2024; Grattafiori et al., 2024; DeepSeek-AI et al., 2025, etc.). However, it has been shown that targeted effort can result in surpassing the most advanced proprietary models with more task-oriented models, e.g., for grammatical error correction (Luhtaru et al., 2024a). Furthermore, hybrid combinations of large language model (LLM) prompting and tuning for synthetic data generation, as well as tuning for the final task, show even more promise (Luhtaru et al., 2024b).

Here we present a case study on the development of grammatical error correction (GEC) and grammatical error explanation (GEE) generation for learners of Estonian. The overall goal is to create task-specific models reliable enough to correct learners' grammar and justify the corrections. Most importantly, while we use proprietary LLMs in this work for data generation, the final result consists of independent open-weight models that can be used for both research and commercial purposes.

The central theme of all the presented work is dealing with data scarcity. The amount of training data for GEC has recently improved but still shows imbalance between English and other languages (Masciolini et al., 2025b) and Estonian is no exception. More specifically, there is a modest amount of Estonian GEC data but no data for GEE. We address both data deficiencies by utilising synthetic data, obtained by prompting OpenAI LLMs (detailed later in the paper) to either introduce grammatical errors into correct texts, in a manner characteristic for language learners (for GEC), or by generating and filtering explanations of gold-standard corrections (for GEE).

Below we describe the developed pipeline and details of generating the data and training the final models in Section 3. Then we present a comprehensive qualitative and quantitative evaluation of the results in Section 4. Finally, Section 5 describes the user feedback, collected from two groups of users: teachers and learners of Estonian as a second language (L2). Since the presented project is an ongoing effort, we finish with a brief description of lessons learned and future work in Conclusion 6.

2 Related Work

2.1 Grammatical Error Correction

The task of grammatical error correction (GEC) is to automatically detect and correct erroneous text. Bryant et al. (2023) argue that although the denomination of the task refers to grammatical errors, the scope of the task is not strictly limited to grammatical errors but other types of errors as well, such as spelling and fluency errors.

Recent approaches have moved from neural MT (Yuan and Briscoe, 2016) to LLM-based (Masciolini et al., 2025a). Even without downstream fine-tuning, LLMs have shown to generate grammatically correct text as an emergent ability (Cao et al., 2023; Coyne et al., 2023), but the edits tend to be

fluency edits as opposed to minimal edits (Fang et al., 2023; Davis et al., 2024).

Automatic error generation (AEG) is a widely applied approach in GEC, consisting of injecting automatically generated errors into correct sentences in order to generate synthetic GEC data. Approaches to AEG include rule-based (Sidorov et al., 2013; Ma et al., 2022), statistical methods (Felice and Yuan, 2014; Kasewa et al., 2018), and neural networks-based work (Grundkiewicz et al., 2019; Bout et al., 2023).

Korotkova et al. (2019) used neural MT for GEC for Estonian. Luhtaru et al. (2024b) used fine-tuned LLMs for both artificial error generation and correction. They used L2 essays for generating errors and evaluated the results on the Estonian learner language (EstGEC-L2) corpus¹. They concluded that using Llama-2-based fine-tuned models gave the most human-like distribution of generated errors. Another dataset, the EKI error-annotated L2 (EKI-L2) corpus², was released in 2024. The two corpora are included in the MultiGEC-2025 shared task (Masciolini et al., 2025b). The best results were achieved by the multilingual LLM based model of Staruch (2025), providing the whole essay at once for correction. Most GEC approaches and evaluation methods are sentence-based (Bryant et al., 2023), including previous work in Estonian GEC, which limits the system’s access to the broader context necessary for correctly detecting and correcting paragraph- or document-level errors.

2.2 Grammatical Error Explanation

Alongside GEC, the task of grammatical error explanation (GEE) has received increasing attention. Providing a reason for each correction helps language learners and other users to understand and learn from their errors.

Chen et al. (2017) extracted grammar patterns from a reference corpus to assist L2 learners of English in academic writing. E.g., the correction *chance for giving* → *chance to give* would be explained by the edit pattern *chance: N for -ing* → *N to v*. Lai and Chang (2019) also detected problem words co-occurring with grammar edits. They formulated feedback templates depending on the error type, classified by ERRANT (Bryant et al., 2017).

Another enhanced GEC system by Kaneko et al. (2022) presents related language examples based

on k -nearest-neighbour machine translation trained with incorrect-correct sentence pairs from English learner corpora. However, GEE-specific datasets allow to train models that give more detailed responses. Hanawa et al. (2021) experimented with neural retrieval and generation methods using L2 English essays manually annotated with feedback comments. Fei et al. (2023) introduced a dataset with error type and problem word annotations, using it for BERT-based token classification and error class prediction.

While it is costly to produce human-annotated or carefully engineered corpus-induced training data, the prompting of LLMs can offer a more accessible solution for GEE. Maity et al. (2024) prompted various LLMs in one-shot mode to correct erroneous Bengali sentences and obtain a brief explanation of each error. Song et al. (2024) achieved a better GEE performance with a two-step pipeline for explaining German and Chinese error corrections. First, they prompted and fine-tuned LLMs to extract atomic edits (insert, delete, replace, relocate). Then, explanations were generated by few-shot prompting GPT-4. This significantly improved the results compared to using only sentence pairs as input. Kaneko and Okazaki (2024) and Ye et al. (2025) similarly leveraged the in-context learning capabilities of the GPT models to synthesize English and Chinese error explanation data, respectively. Ye et al. (2025) used their dataset to fine-tune open-source LLMs both in a pipeline and multi-task setting, integrating GEC and GEE.

We adopt the LLM-based pipeline approach and include error types in addition to atomic edits. As a novel contribution, we provide each edit with two explanations of different detail levels: 1) a brief overview of the error cause and 2) a more comprehensive reasoning mainly aimed at advanced learners and teachers. We create synthetic data with both types of explanations by few-shot prompting GPT-4o and fine-tune a Llama-2-based LLM adapted for Estonian.

3 System Development

The system development consisted of data generation and fine-tuning for GEC and GEE. The resulting system pipeline consists of three steps: 1) grammatical error correction, 2) error tagging and 3) error explanation. The models, alongside the generated synthetic training datasets, are public and have a permissive license.

¹<https://github.com/tlu-dt-nlp/EstGEC-L2-Corpus/>

²<https://doi.org/10.15155/27bh-ny83>

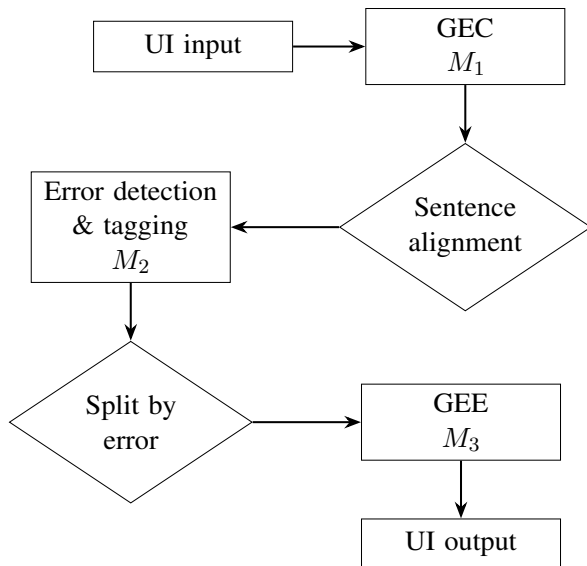


Figure 1: A high-level overview of the system. Each M denotes a model fine-tuned for the given task.

Figure 1 gives a high-level overview of the system (see a detailed example in Appendix A). The user’s input text, i.e., a paragraph, is passed to the first model M_1 as a whole, which then outputs the corrected text.³ The input and corrected text are split into sentences and aligned with the sentence aligner, resulting in input-output sentence pairs. If the input sentence is not equal to the output sentence, thus an error was corrected, the pair is passed to the second model M_2 , which outputs a list of tagged error corrections for each sentence pair.⁴ If input and output are equal, the following models are skipped. Otherwise, the sentence pair and the list of tagged errors are passed to the third model M_3 error by error, which explains the error correction.⁵

Next, we discuss these steps in detail. Fine-tuning is elaborated in Subsection 3.1. Subsections 3.2 and 3.3 delve into GEC and GEE model development, which involves experiments for data generation and finally generating the final datasets and fine-tuning the models. Although the GEC model works on paragraph level, GEE was performed on the sentence level due to context window limitations; to that extent, we performed sentence alignment, details of which are given in Subsection 3.4.

³<https://huggingface.co/tartuNLP/Llammas-base-p1-GPT-4o-human-error-mix-paragraph-GEC>

⁴<https://huggingface.co/tartuNLP/Llammas-base-p1-GPT-4o-human-error-pseudo-m2>

⁵<https://huggingface.co/tartuNLP/Llammas-base-p1-GPT-4o-human-error-explain-from-pseudo-m2>

3.1 Fine-tuning details

Base model For the base model we chose Llammas-base, which is a Llama 2 7B model that has been trained on Estonian texts in continued pre-training setting (Kuulmets et al., 2024) and has shown SOTA results by fine-tuning for Estonian sentence-level GEC (Luhtaru et al., 2024b).

Training parameters Apart from the maximum sequence length, which was 4096, 2048 and 4096 for M_1 , M_2 and M_3 , we used the same parameters as Luhtaru et al. (2024b) for all three models. For prompts, see Appendix B.

Hardware The models were trained on 2 AMD MI250X GPUs in the LUMI supercomputer,⁶ which totals 4 GPUS because AMD MI250X GPU is considered as two GPUs from both hardware and software perspectives in LUMI, each having access to 64 GB of memory.

3.2 Error correction and error type detection

For human-annotated training data, we used the EKI-L2 GEC corpus, which is part of MultiGEC-2025 (Masciolini et al., 2025b). It consists of 1,503 learner essays and 17,361 sentences, nearly 3/4 of which include at least one grammatical error. The dataset includes both minimal and fluency edits, we used the part with minimal corrections. Similarly to EstGEC-L2, its annotation follows the M^2 format and ERRANT error classification (Bryant et al., 2017) adapted for Estonian.

To increase the amount of pre-training data, we also generated synthetic data for Estonian. Moreover, as Luhtaru et al. (2024b) show that inserting errors into out-of-domain texts can actually hurt performance, we compare synthetic error addition to the original GEC data (corrected essays in EKI-L2), similar-domain human texts and synthetically generated texts. Human data from the similar domain consists of similar-sized excerpts of fiction, extracted from the Estonian National Corpus⁷.

We used OpenAI’s GPT-4o⁸ to generate essays based on EKI-L2. For each original corrected essay, we gave it as a 1-shot sample and prompted the model to generate a similar, correctly written essay given the original proficiency level. With temperature 1 the generated essays followed the argumentative structure of the sample with a new

⁶<https://www.lumi-supercomputer.eu/>

⁷<https://doi.org/10.15155/3-00-0000-0000-0000-08D17L>

⁸<https://platform.openai.com/docs/models/gpt-4o>

topic, at higher temperature levels the amount of noise rose sharply, so the essays were generated at 1.1. While we filtered out most of the noise, a few generated essays did include some grammatical errors.

GPT-4o few-shot prompting (at temperature 1.0) was then used to generate errors in these texts, sentence-by-sentence, given 5 randomly picked corrected-mistaken sentence pair samples from the original EKI-L2 corpus. Each of the four datasets was used as error correction training data to fine-tune a Llammas-base (Kuulmets et al., 2024) model for 3 epochs, which was then tested on 141 essays of the development set of EstGEC-L2 corpus (levels A2–C1).⁹ For training details, see Subsection 3.1. As some grammatical errors can only be detected when considering the context, the error correction models were given the whole essay as input, splitting only if the essay was too long to fit into the context window. The results can be seen in Table 1.

	P	R	F _{0.5}
Human errors			
EKI-L2	76.64	40.35	64.95
Synthetic errors			
EKI-L2	71.40	41.45	62.39
Fiction excerpts	70.55	46.55	63.96
Generated essays	69.70	49.19	64.33

Table 1: Error correction precision (P), recall (R) and F-score (F_{0.5}) after 3 epochs of training on different genres. Scores of the model trained on human errors versus models trained on synthetic errors generated into the listed datasets. EKI-L2 synthetic errors were generated into the target sentences, leaving with synthetic source sentences.

As automatically generated essays proved to yield good results compared to other training corpora, we 1) generated a 10 times larger set of essays, 2) introduced artificial errors to the generated essays and 3) employed a two-stage fine-tuning procedure for GEC. We first fine-tuned Llammas-base on a randomly shuffled 10:1 mixture of synthetic-human data. We then fine-tuned the best-performing checkpoint from the first stage on EKI-L2 human dataset. The checkpoint with the highest F_{0.5} score on the development set served as the base model for the second stage of the fine-tuning, which was fine-tuning the model again on the human dataset. The optimizer state was

⁹Originally 102 essays; longer essays were split.

reinitialized and the hyperparameters remained the same as in the first stage of fine-tuning. The third checkpoint of the final model served as the GEC model M_1 in the workflow.

For error detection and classification, we transformed EKI-L2 M^2 edits into simplified atomic edits with error type information, to be given as input for GEE. We fine-tuned a Llammas-base on the EKI-L2 set with atomic edits, resulting in the error tagging model M_2 in the workflow.

3.3 Error explanation

To generate training examples for GEE, we evaluated three approaches using OpenAI’s GPT models: 1) single-prompt parallel input, where the model was given original and corrected sentence pairs; 2) single-prompt error-tagged input, which provided correction edits and error-type information; and 3) prompt chaining with parallel input, which identified and explained corrections through separate prompts. These approaches were assessed using zero-shot and few-shot prompting.

The evaluation was based on 40 random sentence pairs from the EstGEC-L2 development set, including 10 pairs per proficiency level (A2–C1). In case of multiple error annotations, the first version was chosen. For each error, we requested either a single explanation or paired explanations: one brief and one more comprehensive. We rated their quality using colour codes based on traffic lights, so that green indicates good, yellow fair and red poor explanations. More precisely, green represents clear and sufficient information. Yellow denotes partial or nonfluent information that may still be helpful and does not mislead the user. Red explanations contain incorrect statements and terms, or simply describe the correction, but do not offer a justification. The explanation accuracy was defined as the percentage of good and fair explanations.

Annotators were three research group members with a linguistic background and previous experience in L2 error annotation. There was one annotator per each explanation. The annotation was reviewed by an L2 teaching expert participating in our project. The expert-guided evaluation principles were jointly discussed and specified throughout the evaluation process.

Initial experiments used Estonian and English zero-shot prompts and compared the performance of GPT-4o, GPT-4, and GPT-3.5 Turbo with Microsoft Azure’s default settings (temperature 0.7, top p 0.95) and reduced variability (lowering ei-

ther of the parameters). GPT-4o with default settings outperformed other models, producing fewer factual or logical errors, particularly in detecting Estonian case forms and sentence interpretation. Notably, Estonian prompts yielded more precise and fluent explanations. Requesting paired explanations provided higher-quality responses. In particular, comprehensive explanations could be considered more accurate and informative compared to single ones. Brief explanations were more problematic, often describing corrections (e.g., word x should be y) without any additional context.

The results generally improved by first generating the longer explanation instead of the shorter one. This way, the accuracy of long explanations increased from 29% to 64% with single-prompt parallel input and from 62.5% to 83% with error-tagged input. Brief explanation accuracy went from 0% to 18% with single-prompt parallel input and dropped from 54% to 48% with error-tagged input.

Adopting the paired explanation approach, we refined the best Estonian prompts to avoid redundant or insufficient information. For few-shot prompting, we constructed examples based on eight Estonian learner sentences, representing the 12 main error types and some combined errors (see (1) for an example of explanation input and output, translated into English). The single-prompt approach proved more effective with the few-shot method, whereas the prompt chaining did not yield better results. Its long explanation accuracy decreased from 58% to 36% and brief explanation accuracy from 56% to 44% compared to the zero-shot method. The main limitation was detecting atomic edits despite the few-shot examples. In a test where GPT-4o had to identify GEC edits three times per sentence pair and select the correct answer, it chose the right output for 28 out of 40 sentence pairs.

- (1) Source sentence: Head aega.
 Target sentence: Head aega! ('Goodbye!')
 Correction(s):
 1. incorrect punctuation: . -> !

 Explanation 1: . -> !
 Long: In Estonian, a greeting sentence ends with an exclamation mark, e.g., "Tere hommikust!" ('Good morning!'), "Head uut aastat!" ('Happy new year!').
 Brief: An exclamation mark is used after a greeting or wish.
 Error type: incorrect punctuation

Since the error-tagged input provided full alignment with actual edits and error types, we decided to use it for training data generation. In addition, this approach led to significantly higher accuracy in longer explanations (91% compared to 65% with parallel input). The accuracy of brief explanations was lower, equally 52%, mostly due to merely descriptive explanations. Therefore, we further improved the prompt to provide more meaningful clarifications. We synthesized error explanations based on the EKI L2 corpus from Subsection 3.2, using the 12,580 sentences that contain the atomic edits. We fine-tuned a Llammas-base model to generate explanations error-by-error on the synthesized dataset, resulting in model M_3 in the workflow.

3.4 Sentence alignment

As the error explanation model required input on sentence level, the essay from model M_1 error correction output had to be aligned with its input on sentence level. The same need came up when evaluating M_1 output. Complications rose when a sentence was split into several or several sentences joined as part of the correction, also when the M_1 model hallucinated new sentences, such as a greeting to start a letter. Sometimes a mismatch was caused by the sentence tokenizer mistaking the sentence boundaries in uncorrected text.

To solve this problem, we developed a simple many-to-many sentence aligner based on the Levenshtein distance. When aligning the gold standard and output essays during evaluation we considered the distance between corrected sentences of both essays, merging the gold M^2 representations as necessary. Testing on 400 essays of the training corpus, this rule-based aligner found correct alignments for 98% of the original sentences.

4 System Evaluation

4.1 Error correction performance

Error correction scores were automatically evaluated on the EstGEC-L2 development set using a modified version¹⁰ of the M^2 scorer (Dahlmeier and Ng, 2012). This yields error-level F-score comparing the output sentence with all given gold corrections, as well as a broad statistics of recall by error type. The modified version also takes into account that the word order error type ($R:WO$) used in train and test corpora can encompass other errors, as word order in Estonian tends to be rather

¹⁰<https://github.com/TartuNLP/estgec/>

free and the scope of that error type may include a large part of the sentence. The results on the development set of EstGEC-L2 corpus can be seen in Table 2, comparing the models trained on smaller or larger datasets of synthetic essays, and the latter model post-fine-tuned on human errors. While using a large number of generated essays did significantly raise recall, it is surprising that additional fine-tuning on human errors brought it down without much increase in precision. This may be partly due to the larger training set containing human errors already contributing to higher precision.

	P	R	F _{0.5}
GPT-4o	69.56	54.13	65.81
Synth _S	69.70	49.19	64.33
Synth _L -EKI-L2 Mix	75.61	45.68	66.85
+ EKI-L2 FT	76.45	42.45	65.90

Table 2: Scores of models trained on datasets with synthetic errors, based on the best F_{0.5} score across 3 epochs, compared to prompting GPT-4o at temperature 1 for GEC in a 1-shot setting (see Appendix B for details). Synth_S was trained on 1,503 generated essays with synthetic errors, while Synth_L-EKI-L2 Mix was trained on a dataset 10× larger, mixing synthetic and EKI-L2 errors. Synth_L-EKI-L2 Mix was also post-fine-tuned on EKI-L2.

While our F_{0.5} score is notably higher than the 49.44 reported by Staruch (2025), theirs was a multilingual system tested on a smaller subset of the EstGEC-L2 corpus. GPT-4o achieves a higher recall but a lower precision, resulting in a lower F_{0.5} score. We incorporated the EKI-L2 fine-tuned Synth_L-EKI-L2 Mix model trained for 3 epochs in our workflow as the final M₁ model, although its F_{0.5} score was better after epoch 1, recall was highest after 3 epochs.

The modified M² scorer shows recall by error type even if M₁ does not assign types. Considering the results (see Figure 2), the most difficult type by far is word order, mostly because the correction is considered accurate only if all possible encompassing errors are corrected as well. E.g., the phrase ‘*raamat loen ma*’ (‘*book-nom read I*’) should be corrected not only as ‘*ma loen raamat*’ (‘*I read book-nom*’), but also with the correct case ‘*ma loen raamatut*’ (‘*I read book-part*’). Note that the error correction model does not assign an error type, so even if it detects an error in the same scope as a nominal form error, it might try to replace or erase the whole word.

Leaving word order aside, the more difficult types to correct are word and punctuation choice (R:LEX, R:PUNCT), although missing punctuation marks (M:PUNCT) tend to be rather easy. This is somewhat expected as the choice of words for replacing an unsuitable one is rather large and not all suitable words are listed in human corrections. Inserting missing punctuation marks, correcting the capitalization (R:CASE) or whitespace, i.e., compounding errors (R:WS) as well as picking the right nominal or verb form (R:NOM:FORM, R:VERB:FORM) are all handled with slightly higher recall, as could be expected from a strong language model. As was seen from the evaluation scores, adding synthetic data helped raise recall. This seems to be mostly due to better detection of word order and compounding errors. The final model also detects capitalization errors noticeably better than the one trained on human errors, but if we consider corrections, then they are around the same level, as the model has trouble providing correct replacements. If we consider what may have contributed to the drop of precision in the final model, then most noticeable bottlenecks are detecting unnecessary words (U:LEX) and correcting complex errors where there are several mistakes in one word (e.g., wrong verb form with a spelling mistake – R:VERB:FORM:SPELL).

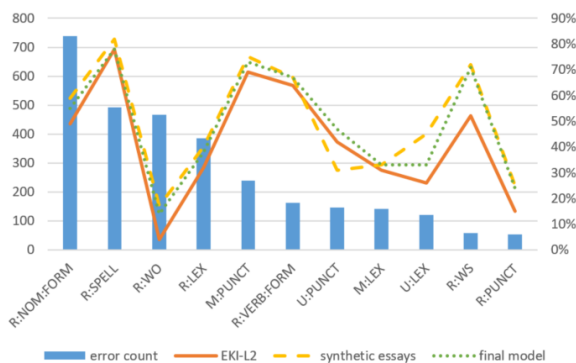


Figure 2: Recall by type for the 11 most frequent error types in the test corpus. Number of errors of each type present in test corpus is shown in columns for reference.

The test dataset has essays from proficiency levels A2–C1, whereas C1 was not present in the training corpus that contains K-12 student essays. When comparing error correction performance of the fine-tuned model across language proficiency levels (see Table 3), the results are quite uniform. The scores are a little better for A2, possibly because the sentences used are still rather simple. Recall is no-

	P	R	F _{0.5}
A2	74.12	48.99	67.22
B1	71.48	48.13	65.16
B2	74.00	43.32	64.82
C1	71.43	47.23	64.79
All	72.91	46.58	65.51

Table 3: GEC scores by language proficiency level.

ticeably lower for B2. The drop is most clear for two error types: word order and wrong punctuation mark. Out of 26 cases of wrongly chosen punctuation marks in B2 texts, none received the correct replacement, although a mistake was detected in more than third of these. B2 texts had more word order mistakes (196, compared to 127 in B1 texts and less than 100 on other levels), but even the recall of partial scope overlap was lower than for other levels: 32% compared to 47% or more. The distribution of more common error types and their detection rate can be seen in Figure 3. As for other proficiency levels, the model has relatively more difficulties detecting missing words in A2 texts, missing punctuation in B1 texts, and wrong capitalisation in C1 texts, although in the last case there were only 4 such mistakes present, which may be too few to draw conclusions.

4.2 Qualitative analysis of system output

For qualitative assessment of the three system components — corrector, error detector/classifier, and explainer — we randomly selected 40 sentences from the EstGEC-L2 test corpus, balanced for proficiency level (A2–C1). We compared two settings: a uniform 0.7 temperature and a varied higher temperature (1.0 for M_1 , 0.8 for M_2 , 0.9 for M_3) to encourage creativity.

In comparison with golden edits, we distinguished four correction types: necessary and suitable, necessary but incorrect, unnecessary but suitable, and unnecessary and incorrect. We calculated precision based on both types of suitable edits. The macro-averaged precision of error classification was assessed according to proposed changes, even if incorrect. Explanations were graded as good, fair, or poor, as described in section 3.2 (see translated examples in Appendix D). We separately evaluated explanations for necessary suitable corrections, since it is challenging or even futile to explain unnecessary or incorrect edits.

Lower temperature entailed higher correction

precision (89% vs. 76%) and fewer edits (63 vs. 71), while the number of suitable corrections was similar (56 vs. 54). The 0.7 setting also resulted in a greater overlap with reference edits (60% vs. 46%). However, the correction model then failed to detect word order errors. We suggest using an intermediate temperature for GEC. The average precision of error classification was comparable in the two conditions: 84% with higher and 87% with lower temperature.

The quality of explanations was generally better at the 0.7 temperature (see Table 4). Long explanations were more likely to be rated good or fair compared to the 0.9 temperature both in case of necessary corrections and all corrections, including optional and unjustified edits. Necessary brief explanations followed a different trend, being more accurate at the higher temperature. Nonetheless, when considering all system corrections, the proportion of good and fair explanations remained similar in the 0.7 setting, whereas radically dropping in the 0.9 setting. This refers to a better capability to justify optional edits at the lower temperature.

	Temp 0.7	Temp 0.9
Long explanations		
Necessary: good	51%	37%
Necessary: good/fair	66%	48%
All: good	51%	27%
All: good/fair	63.5%	37%
Brief explanations		
Necessary: good	45%	50%
Necessary: good/fair	70%	76%
All: good	46%	34%
All: good/fair	67%	51%

Table 4: GEE quality with two temperature settings.

In terms of GEE, our results can be compared with Maity et al. (2024) and Ye et al. (2025), who reported accuracy over 60%, and outperform Hanawa et al. (2021), who reached 40%–50% accuracy in explaining preposition errors and below 40% with various error types. One shortcoming was the inaccuracy of linguistic terms, such as using an existing term in a wrong context (e.g., false association of nominal case and word form) or forming a nonexistent term. This concerned 24% of long and 3% of brief explanations in the lower temperature setting. Furthermore, the prompt could be improved to explain context-dependent errors like grammatical form or word choice errors.

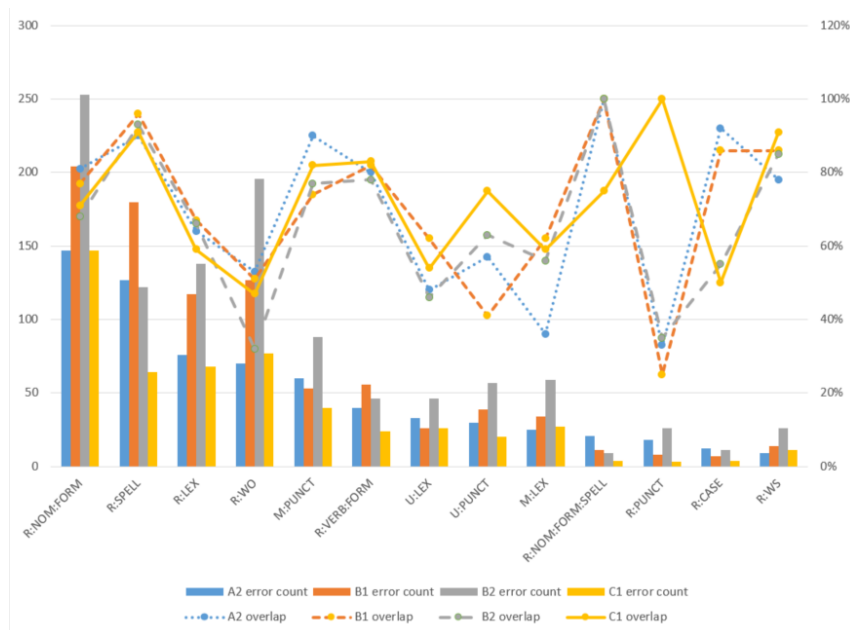


Figure 3: Error detection recall by language proficiency level for more common error types, considering at least partial overlap in scope.

5 User Feedback

5.1 Participants and questionnaire

The user test involved six learners and five teachers and/or testing experts of Estonian as L2. Volunteers were recruited using snowball sampling, considering their experience of learning and teaching at different proficiency levels. The teachers' expertise covered the whole range from pre-A1 to C1 level. Three teachers had taught adults and students from pre-A1- or A1- to C1-level. One had focused on adult learners with diverse backgrounds at levels A1 and A2. One had assessed exams at all tested levels but primarily at B2 and C1, prepared exam and screening test tasks, and briefly taught language courses. Three of the teachers were native Estonian speakers, one an Estonian-Russian bilingual and one a native Russian speaker who uses Estonian at home, at work and in daily life. The learners were Russian- and Ukrainian-speaking and bilingual (Ukrainian-Russian or Russian-Estonian). Three of them had lived in Estonia for many years or most of their lives and rated their language level as B2 and C1. The others reported their level to be A2 or B1, having spent 2.5–5.5 years in Estonia. Their exposure to Estonian ranged from rare use in lessons or grocery store to everyday use at work.

We used Google Forms to gather feedback through a semi-structured questionnaire. The respondents were given the option to answer the sur-

vey in English. We developed a demo application for testing (see Appendix C). First, we asked the users to assess the output for a sample B1-level text fragment. Repeated analysis of the same text may give varying results, so we presented a pre-given version of five corrections and explanations as screenshots to ensure response comparability. Subsequently, users interacted with the tool directly, correcting their own text or a student's writing, commenting on each correction and explanation and their general experience.

5.2 Results

Both teachers and learners found that the system makes most of the needed corrections and the majority of explanations could be useful in existing form or with some changes. 10 out of 11 test users considered the corrections somewhat useful or useful (corresponding to 4–5 on a 5-point Likert scale). Explanations were rated similarly by nine respondents. Three teachers and learners noted their plan to use the application in the future, one teacher and two learners would probably use it and one respondent from each group was not sure about it.

All corrections in the provided sample were considered appropriate, although two teachers noted that a lexical choice correction was not strictly necessary. Each explanation was rated on a three-point scale: useful – somewhat useful – not useful. Depending on the correction, long explanations

were found useful by 5–9 and brief explanations by 5–10 test users, averaging to 2/3 of the respondents. About 1/3 and 1/4 of the users, respectively, considered long and brief explanations somewhat useful. On average, one user did not think the shorter explanation was useful. The respondents agreed to defined error types, except for the case where word order and nominal form error occurred together but only the former was detected.

As expected, analysing user texts revealed more issues because these texts were generally longer than the sample and the system made more corrections in them. Teachers found an average of 82% and the learners 87% of corrections relevant. Both groups considered about 70% of long explanations at least somewhat useful, whereas brief explanations seemed useful to more than half of the learners and 3/4 of the teachers. Fixing and explaining structural errors in long complex sentences that contained numerous errors turned out to be challenging. The tool also had some trouble identifying and explaining combined errors. While our aim was to classify and explain all co-occurring error types, only one type may be detected and covered in the explanations (e.g., a spelling error is ignored alongside the choice of correct word form).

The explanations were said to give a comprehensive overview of the errors and help language learners notice errors they might be making systematically. Long explanations were rated higher in terms of content and wording, although there were also instances of complex language use or no added value compared to the shorter version. Users claimed that long explanations should complement short ones, while short explanations should still be informative. In some cases, two explanation layers may not be needed. Both teachers and learners recommended to put more emphasis on simple language comprehensible for A2- and B1-level learners, especially in brief explanations. Some suggestions were made to generalise or specify error classification and improve the user interface, however, the overall assessment was positive in both respects.

6 Conclusion

We trained a workflow of three fine-tuned models for GEC and GEE. Using synthesized L2 texts with introduced errors seems promising, but a larger training set might be necessary. Our first model corrects errors at the paragraph level and performs

well on L2 texts with a proficiency level not present in training data. The model yields higher precision with similar recall at lower temperatures but then struggles with word order errors, so we suggest using medium temperature.

We achieved better quality explanations in GEE by incorporating error types in addition to atomic edits in input and requesting two explanations (longer and shorter) for each error. This could be further improved as both LLM prompting and our fine-tuned model have low recall on detecting error types. It is also necessary to filter out low-quality explanations, such as including nonexistent nominal cases, by possibly using LLMs to evaluate the quality of a given explanation.

While our GEC model was paragraph-based, we used a sentence-based approach due to model limitations. In future work, we will apply a new methodology to preserve context and fit within hardware limits with the context window size. For each sentence in the essay, we will split essays into tuples of N consecutive sentences up to the given sentence. The new methodology could allow us to combine GEC and GEE into one model. In future work, we will also explore reversing the pipeline for fine-tuning for AEG.

Acknowledgments

The project “Autocorrect for Estonian as a 2nd language for learners and teachers” has been co-funded by Estonian Ministry of Education and Research and the European Union. We acknowledge University of Tartu, Estonia for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through University of Tartu, Estonia.

Limitations

Although our GEC system is based on paragraphs, the GEE pipeline, including both error detection and explanation, is based on sentence level due to context window limitations, limiting the model’s capability to explain errors at the document level. Future work needs to classify and explain errors with context, as well as combine related errors for explanation.

The GEE pipeline relies on atomic edits with error-type information, which we found necessary for reasonable explanations. However, atomic edits are based on M^2 , thus making it costly to obtain

new data. Future work should explore the automatic generation of atomic edits.

Additionally, GEC scores rely highly on the sentence alignment method since the M^2 scorer works on the sentence level. Poor sentence alignment affects the scores negatively.

References

- Kais Allkivi, Pille Eslon, Taavi Kamarik, Karina Kert, Jaagup Kippar, Harli Kodasma, Silvia Maine, and Kaisa Norak. 2024. [ELLE – estonian language learning and analysis environment](#). *Baltic Journal of Modern Computing*, 12(4):560–569.
- Andrey Bout, Alexander Podolskiy, Sergey Nikolenko, and Irina Piontkovskaya. 2023. [Efficient grammatical error correction via multi-task training and optimized training schedule](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5800–5816, Singapore. 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.
- Qi Cao, Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. 2023. [Unnatural error correction: GPT-4 can almost perfectly handle unnatural scrambled text](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8898–8913, Singapore. Association for Computational Linguistics.
- Jhih-Jie Chen, Jim Chang, Ching-Yu Yang, Mei-Hua Chen, and Jason S. Chang. 2017. [Extracting formulaic expressions and grammar and edit patterns to assist academic writing](#). In *EUROPHRAS 2017 - Computational and Corpus-based Phraseology: Recent Advances and Interdisciplinary Approaches*, pages 95–103.
- 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.
- 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.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*.
- 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](#). *Preprint*, arXiv:2304.01746.
- Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. [Enhancing grammatical error correction systems with explanations](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7489–7501, Toronto, Canada. Association for Computational Linguistics.
- Mariano Felice and Zheng Yuan. 2014. [Generating artificial errors for grammatical error correction](#). In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126, Gothenburg, Sweden. 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](#). *arXiv preprint arXiv:2407.21783*.
- 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.
- 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.

- 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.
- Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. [Interpretability for language learners using example-based grammatical error correction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.
- Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. [Wronging a right: Generating better errors to improve grammatical error detection](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983, Brussels, Belgium. Association for Computational Linguistics.
- Elizaveta Korotkova, Agnes Luhtaru, Maksym Del, Krista Liin, Daiga Deksnė, and Mark Fishel. 2019. Grammatical error correction and style transfer via zero-shot monolingual translation. *arXiv preprint arXiv:1903.11283*.
- Hele-Andra Kuulmets, Taido Purason, Agnes Luhtaru, and Mark Fishel. 2024. [Teaching llama a new language through cross-lingual knowledge transfer](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3309–3325, Mexico City, Mexico. Association for Computational Linguistics.
- Yi-Huei Lai and Jason Chang. 2019. [TellMeWhy: Learning to explain corrective feedback for second language learners](#). 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): System Demonstrations*, pages 235–240, Hong Kong, China. Association for Computational Linguistics.
- Agnes Luhtaru, Elizaveta Korotkova, and Mark Fishel. 2024a. [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.
- Agnes Luhtaru, Taido Purason, Martin Vainikko, Maksym Del, and Mark Fishel. 2024b. [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.
- 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.
- Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. 2024. [How ready are generative pre-trained large language models for explaining bengali grammatical errors?](#) *Preprint*, arXiv:2406.00039.
- Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025a. [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.
- Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, Robert Östling, Kais Allkivi, Špela Arhar Holdt, Ilze Auzina, Roberts Dargis, Elena Drakonaki, Jennifer-Carmen Frey, Isidora Glišić, Pinelopi Kikilintza, Lionel Nicolas, Mariana Romanyshyn, Alexandr Rosen, and 11 others. 2025b. [Towards better language representation in natural language processing](#). *International Journal of Learner Corpus Research*, 11(2):309–335.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Grigori Sidorov, Anubhav Gupta, Martin Tozer, Dolors Catala, Angels Catena, and Sandrine Fuentes. 2013. [Rule-based system for automatic grammar correction using syntactic n-grams for English language learning \(L2\)](#). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 96–101, Sofia, Bulgaria. Association for Computational Linguistics.
- 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.
- 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.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

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. Exgcec: A benchmark for edit-wise explainable chinese grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25678–25686.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 conference of the north American Chapter of the Association for computational linguistics: Human language technologies*, pages 380–386.

A System Overview

Figure 6 gives a detailed overview of the system with an example. The input paragraph translates to “I very much like classical music. We earlier bought the tickets. Sometimes already in January.” The explanations translate to “The correct spelling is “väga” (very)”, “Instead of the word “klassikat” (“classic” in the wrong case form), the word “klassikaline” should be used to show that the music is of classical type”, “In Estonian, the verb is generally before the adverb (adverb of time), for example, “ostsime piletid” (we bought the tickets) and “varem” (earlier). The original word order “varem ostsime piletid” is wrong.”

B Prompts

Tables 5, 6 and 7 display the prompts used for fine-tuning M_1 , M_2 and M_3 . The prompt format is from Luhtaru et al. (2024b), which in turn is loosely based on Alpaca (Taori et al., 2023) format. Table 8 shows the prompt used for GPT-4o for GEC.

C Demo application

For user testing, we developed a demo application¹¹ based on the Writing Evaluator proofreading tool of an Estonian language learning and analysis environment (Allkivi et al., 2024). The demo reuses existing interface components, such as approving or rejecting corrections and grouping errors by type.

After the user inserts their text, the back-end returns the corrected sentences along with error annotations, including error type and two accompanying

¹¹<https://elle.tlu.ee/corrector-test>

explanations. Users can interact with corrections in two ways: **inline view** — moving the cursor over a highlighted segment in the text triggers a popup displaying a short explanation and options to accept or reject the correction (See Figure 4); **sidebar view** — on the right side of the interface, corrections are grouped by type (see Figure 5). Clicking on a category reveals a list of related corrections with longer explanations.

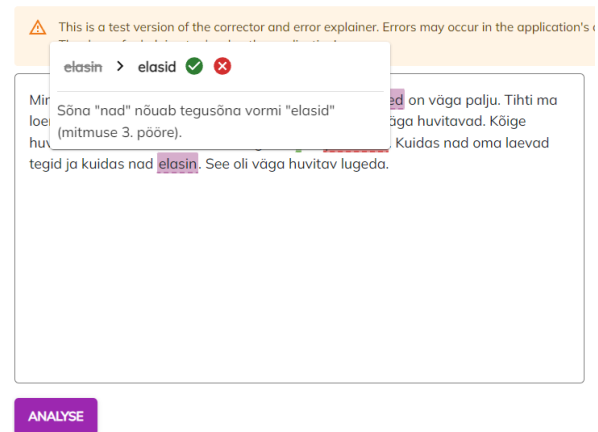


Figure 4: Popup view with a short explanation when hovering over a highlighted correction.

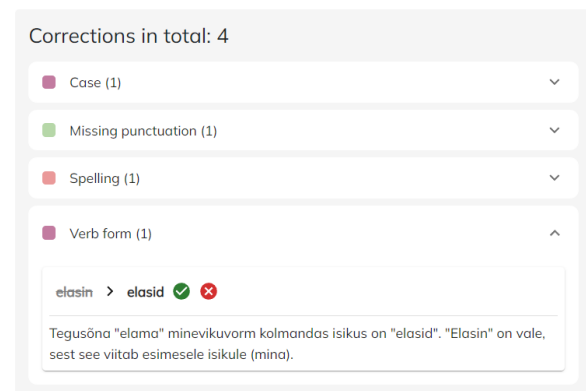


Figure 5: Sidebar displaying error categories and longer explanations under selected corrections.

D Example explanations

We rated the quality of system error explanations on the following scale: good — clearly presented and sufficient information; fair — partial or nonfluent but correct information; poor — use of incorrect statements and terms or edit description without additional context. Example (2) demonstrates a comprehensive and brief explanation annotated as good. Example (3) includes a brief explanation rated as fair and a longer explanation rated as poor.

```

### Instruction:
Reply with a corrected version of the input essay in Estonian with all grammatical and spelling errors fixed. If there
are no errors, reply with a copy of the original essay.

### Input:
{input}

### Response:
{correction}

```

Table 5: The GEC model M_1 fine-tuning prompt. The GEC instruction is a modification of the prompt by [Coyne et al. \(2023\)](#).

```

### Instruction:
Sa võrdled kahte eestikeelset lauset: keeleõppija kirjutatud algne lause ja parandatud lause. Genereeri vea kaupa
paranduste loend, kus on vealiik, algne tekst ja parandatud tekst.

### Input:
Algne lause: {original sentence}

Parandatud lause: {corrected sentence}

### Response:
Parandused: {list of atomic edits}

```

Table 6: The error detection and classification model M_2 fine-tuning prompt.

The correct word “terrassil” is in the adessive case form, not inessive, although both express location in Estonian. The brief explanation only considers the nominal form error, disregarding the spelling error (the base form is “terrass”, not “terrass”).

- (2) Source sentence: Pärast kontserdi me otsustasime juua kohvi restoranis ja koju minna jalgsi.
 Target sentence: Pärast kontserti me otsustasime juua kohvi restoranis ja koju minna jala.
 (‘After the concert, we decided to drink coffee in a restaurant and walk home.’)

Explanation: kontserdi -> kontserti
 Long: The word “päras” (‘after’) requires the partitive case, so the correct form is “kontserti”. The form “kontserdi” is in the genitive case and is not appropriate here.
 Brief: The word “päras” (‘after’) requires the partitive form “kontserti”.
 Error type: nominal form

- (3) Source sentence: Linnas ma istun terras ja joon siider.
 Target sentence: Linnas ma istun terrassil ja joon siidrit.
 (‘In the city, I sit on a terrace and drink

cider.’)

Explanation: terras -> terrassil
 Long: The inessive case form of the word “terrass” is “terrassil”. It expresses location (where?).
 Brief: The correct case form is “terrassil” (where?).
 Error type: nominal form

Instruction:
 Sa võrdled kahte eestikeelset lauset: keeleõppija kirjutatud algne lause ja parandatud lause. Sulle antakse paranduste loend, kus on vealiik, algne tekst ja parandatud tekst. Su ülesanne on selgitada ühte parandust. Selgita seda parandust, mis järgneb sõnale 'Selgitus'. Esiteks too välja põhjalikum selgitus, miks parandust vaja on. Teiseks anna lühike selgitus lihtsamas keeles. Pärast selgitust nimeta vealiik. Mitu vealiiki võivad kokku langeda. Omavahel seotud parandusi (näiteks sõnaühend, kus muutub mõlema sõna vorm) selgita koos. Sõnajärje parandusega kattuvaid muid parandusi selgita eraldi.

Input:
 {sentence pair, list of errors and input error to explain}

Response:
 {explanation for input error}

Table 7: The GEE model M_3 fine-tuning prompt.

Reply with a corrected version of the input text in Estonian with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the input text. There is one example of the task.

Input text: {example input paragraph}
 Corrected: {example corrected paragraph}

Input text: {input paragraph}
 Corrected:

Table 8: The 1-shot prompt used for GEC with GPT-4o. The example was randomly sampled from the EKI-L2 set. The GEC instruction is a modification of the prompt by [Coyne et al. \(2023\)](#).

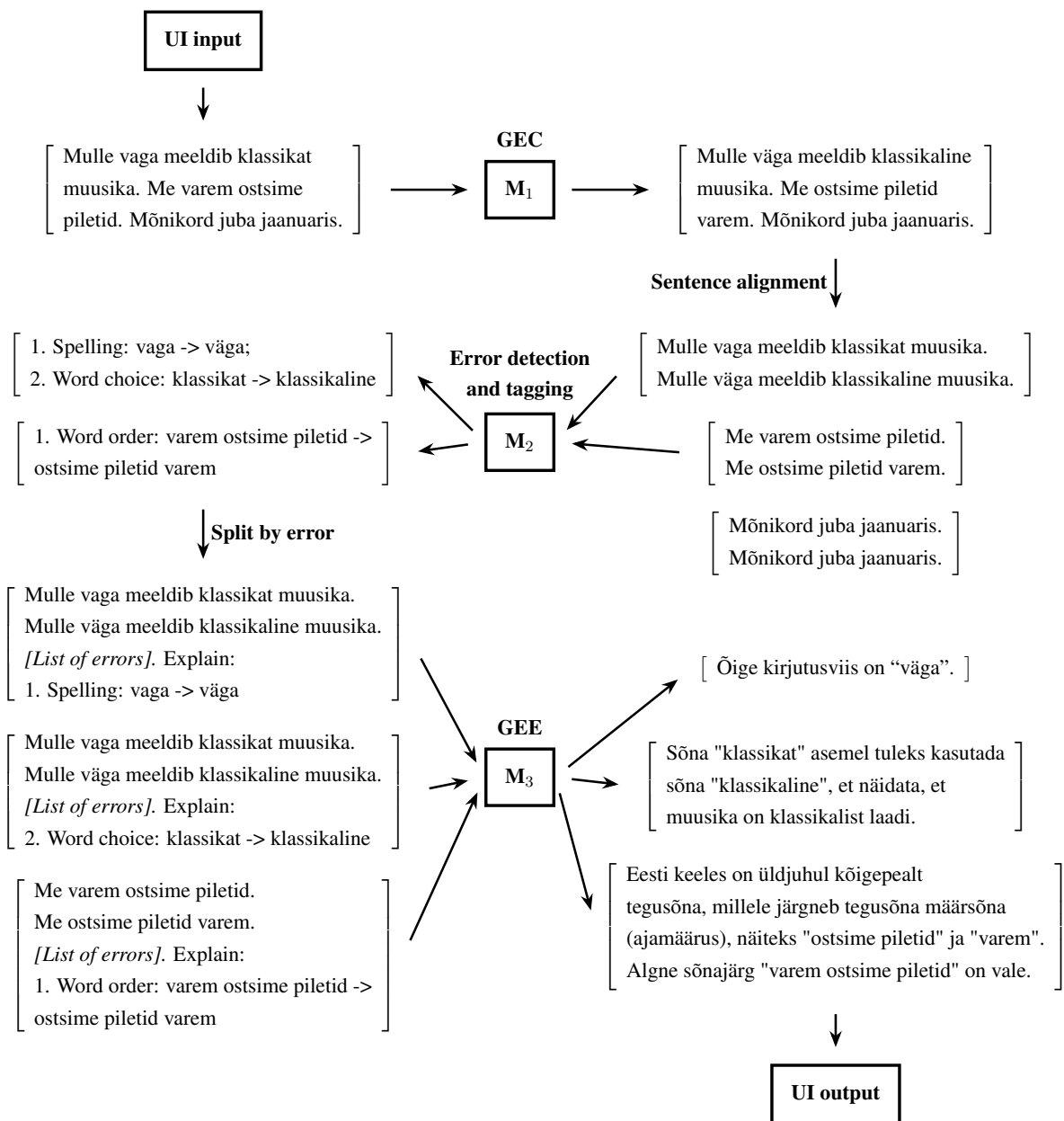


Figure 6: A detailed overview of the grammatical error correction and explanation system with an example. M denotes model.

End-to-End Automated Item Generation and Scoring for Adaptive English Writing Assessment with Large Language Models

Kamel Nebhi, Amrita Panesar, Hans Bantilan

Education First

Selnaustrasse 30

8001 Zürich, Switzerland

{kamel.nebhi, 09panesara, hansbantilan}@gmail.com

Abstract

Automated item generation (AIG) is a key enabler for scaling language proficiency assessments. We present an end-to-end methodology for automated generation, annotation, and integration of adaptive writing items for the EF Standard English Test (EFSET), leveraging recent advances in large language models (LLMs). Our pipeline uses few-shot prompting with state-of-the-art LLMs to generate diverse, proficiency-aligned prompts, rigorously validated by expert reviewers. For robust scoring, we construct a synthetic response dataset via majority-vote LLM annotation and fine-tune a LLaMA 3.1 (8B) model. For each writing item, a range of proficiency-aligned synthetic responses, designed to emulate authentic student work, are produced for model training and evaluation. These results demonstrate substantial gains in scalability and validity, offering a replicable framework for next-generation adaptive language testing.

1 Introduction

The demand for scalable, authentic, and adaptive English proficiency assessments has grown rapidly in recent years, as language learning expands across global and digital platforms. This surge has compelled test developers to explore advanced Natural Language Processing (NLP) and Machine Learning (ML) solutions that can deliver reliable and fair measurement at scale. The EF Standard English Test (EFSET)¹ exemplifies recent innovation in this space, having introduced performance-based Writing and Speaking tasks that leverage state-of-the-art NLP and ML methods for both test delivery and automated scoring (Nebhi and Szaszák, 2023; Williams et al., 2022).

Despite these advances, item generation remains a major challenge for adaptive assessment. Creating high-quality prompts that are valid across a

range of topics, calibrated for all proficiency levels, and secure from test exposure is both resource-intensive and psychometrically complex (Zhang et al., 2022; Brown, 2023; Gierl and Haladyna, 2012). As the development and deployment of adaptive language tests like EFSET increases, scalable and robust methods for generating, validating, and securing writing assessment items are crucial for the advancement of fair and accurate proficiency measurement.

To address this issue, we present a novel pipeline for generating and incorporating new items into the EFSET writing assessment scoring process. Our method uses Large Language Models in Automatic Item Generation (AIG) and Synthetic Data Generation for Student Responses for scalable adaptive writing assessment. First, we generate new assessment items using a few-shot learning strategy applied to LLMs, systematically exploring multiple prompting combinations. Human evaluators then verify item quality, ensuring appropriate difficulty, clarity, and topic relevance.

In order to then integrate these validated, newly generated items into our existing automated assessment pipeline, we fine-tune a LLaMa-3.1 8B model via ORPO (Optimized Reward Preference Optimization) (Hong et al., 2024) to generate realistic student-like responses for these items across different proficiency levels. The fine-tuning relies on real test data combined with systematically generated synthetic annotations obtained via consensus annotation (majority vote) from three distinct LLM annotators. These item-response pairs then allow use to train our existing RoBERTa-based Transformer model for proficiency scoring on these new writing prompts. This synthetic annotation approach ensures scalable yet reliable response-label assignment without intensive human labor.

A summary of the main contributions of this paper is as follows: (1) we introduce an automated item generation (AIG) pipeline for adaptive writ-

¹<https://www.efset.org/en/>

ing assessment that leverages state-of-the-art large language models and few-shot prompting to create high-quality, proficiency-aligned prompts; (2) we propose and validate a synthetic data augmentation process based on fine-tuned LLMs and consensus annotation via majority voting, resulting in robust and reliable datasets for model training; and (3) we develop and empirically evaluate a fully automated scoring framework based on Transformer models (RoBERTa), demonstrating significant gains in accuracy and consistency through extensive testing on EFSET items and a carefully calibrated validation set.

In the following sections, we first review the state of the art in automated writing assessment and item generation. We then detail our methodology for prompt generation, dataset construction, and automated evaluation. Next, we present empirical results illustrating the validity and reliability of our approach on both the EFSET validation set and a dedicated calibration dataset. Finally, we discuss the implications and potential extensions of this framework for scalable, adaptive language proficiency assessment.

2 Related Work

This section synthesizes two key developments in recent research on automated language assessment. First, we review state-of-the-art approaches to item generation that leverage large language models (LLMs), prompt engineering, and few-shot learning to efficiently produce diverse and high-quality assessment prompts. Second, we examine emerging methods for synthetic data annotation, with a particular focus on the use of LLMs to simulate candidate responses and facilitate reliable labeling at scale for proficiency scoring tasks.

2.1 Automated Item Generation with LLMs and Prompting

The automated generation of test items, especially for language assessment, has evolved considerably in recent years. Early systems used template-based approaches, in which test developers designed fixed “item shells” and populated them with variable linguistic elements—such as word lists or grammatical forms—to produce items at scale (Bejar et al., 2003). For example, thousands of cloze items, exercises where words are removed from a passage for the student to fill in the gaps, could be created programmatically by instantiating such templates

with preselected vocabularies and distractors, providing structural consistency and psychometric control. However, content diversity and authenticity remained limited by the template bank, and extensive manual authoring was needed to cover new topics or scales. These constraints have since led to the exploration of more flexible, data-driven methods, most notably involving large language models (LLMs).

The advent of large pre-trained language models (LLMs) has fundamentally shifted automated item generation toward more data-driven, scalable, and flexible paradigms. Models such as GPT-3 and GPT-4 have been shown to generate diverse assessment items—including reading, writing, and cloze tasks—by leveraging few-shot prompting, where only a handful of examples guide the model’s output (Brown et al., 2020). Educational evaluation shows that LLM-generated items are closely aligned to human-authored items, with Zhang et al. (2022) reporting that over 80% of reading comprehension questions automatically generated by GPT-3 were rated as valid by expert reviewers, and prompt appropriateness and difficulty levels closely aligned to human-authored items. Similarly, Kurdi (2023) found that LLMs could create contextually relevant language assessment prompts, achieving human-likeness scores above 4/5 on standard rubrics. Research by Brown (2023) and Zhai et al. (2023) supports that such approaches not only accelerate item production and reduce costs, but also enable rapid adaptation to new topics and test formats, with acceptance rates for LLM-generated prompts ranging from 60–95% after light expert editing.

However, even high-performing LLM-generated items require careful evaluation and annotation before they can be reliably used in machine learning-based assessment pipelines to ensure that they are well-calibrated and capable of distinguishing student ability. Recent work has shown that using synthetic annotation, consensus labeling strategies (e.g., majority voting among multiple LLMs), or semi-automatic calibration processes significantly improves dataset consistency and psychometric validity (Liu, 2023; Mai, 2022; Clark, 2021; Yao et al., 2024).

2.2 Synthetic Data Annotation for Language Assessment

A persistent challenge in automated educational assessment is the limited availability of high-quality

annotated data required to train and validate modern NLP models for predicting student proficiency. Recent advances have addressed this by not only generating novel assessment prompts, but also leveraging LLMs to simulate candidate responses and assign linguistic proficiency or accuracy labels at scale (Yao et al., 2024; Wang et al., 2024; Brown, 2023).

For instance, Clark (2021) showed that the scarcity of labeled data for language tasks can be mitigated by generating synthetic examples, improving model robustness and generalization. Moreover, ensemble annotation methods—where multiple LLMs independently label each sample and a majority vote is used—have demonstrated increased labeling reliability, especially when compared to standard human annotation benchmarks (Liu, 2023). These synthetic annotation strategies make it possible to rapidly construct large, diverse, and reliable datasets matched to newly generated items.

Integrating both automated item generation and synthetic annotation creates a complete pipeline for adaptively expanding and enhancing machine learning-based scoring systems. Not only does this combination facilitate the inclusion of new item types without costly manual labeling, but it also supports the continual improvement of model accuracy, as shown by transformer-based scoring systems trained on such enriched datasets (Mayfield and Black, 2020; Mai, 2022). This integrated approach forms the basis for recent innovations in fully automated and scalable assessment frameworks.

3 Proposed Approach

This section presents our integrated pipeline for adaptive writing assessment, encompassing automated item generation, synthetic training data creation, LLM-based response simulation, and transformer-based scoring. Our approach is designed to efficiently generate, validate, and psychometrically calibrate novel test items, ensuring both robustness and scalability for deployment in real-world language proficiency testing environments.

3.1 Overview of the Integrated Pipeline Approach

Figure 3.1 presents the end-to-end pipeline developed for adaptive writing assessment. The process begins with the generation of new writing prompts,

using prompt engineering and few-shot learning with a single LLM (GPT-4o) to produce candidate items focused on specific topics and proficiency levels. All generated prompts undergo human expert review, where only validated items are retained for integration into the assessment bank.

We fine-tune a LLaMA 3.1 (8B) model—leveraging ORPO optimization—on a custom instruction dataset to generate synthetic student responses for each validated item reflecting varying proficiency levels. This instruction dataset is created by collecting real candidate answers, prompting several LLMs to annotate each response for accuracy using the few-shot paradigm, and applying a majority voting scheme to select the final label. Only samples with strong inter-model agreement are retained, ensuring high label reliability and calibration.

The resulting synthetic dataset, containing approximately 200 responses per new item, is then used to further train and fine-tune a RoBERTa-based transformer scoring model. This updated scoring engine is evaluated both on existing and new items to ensure seamless integration and consistency. Throughout the pipeline, quality is maintained through a combination of automated filtering and targeted human-in-the-loop validation, enabling scalable, reliable item generation and robust scoring for real-world proficiency assessment.

3.2 Phase 1: Automated Creation and Validation of Writing Items

The aim of this first phase is to automate the generation of writing prompts intended for students to write an essay about, each of which is targeted at a specific proficiency level and topic. We leveraged a large language model (LLM)—specifically, GPT-4o (OpenAI, 2024)—to achieve this. To ensure that the generated writing prompts were tailored to adaptive assessments, we provided the LLM with a set of representative triplets of writing prompt, proficiency level, and topic. We then guided the LLM to generate new writing items with the intended form, content scope, and level-appropriateness by few-shot prompting the LLM with examples of the target structure and the level of complexity required.

In order to guide the LLM, we first carefully curated a small set of ≈ 20 examples explicitly designed to match the communicative demands of English writing proficiency tests. Each writing example consisted of a proficiency level and a suc-

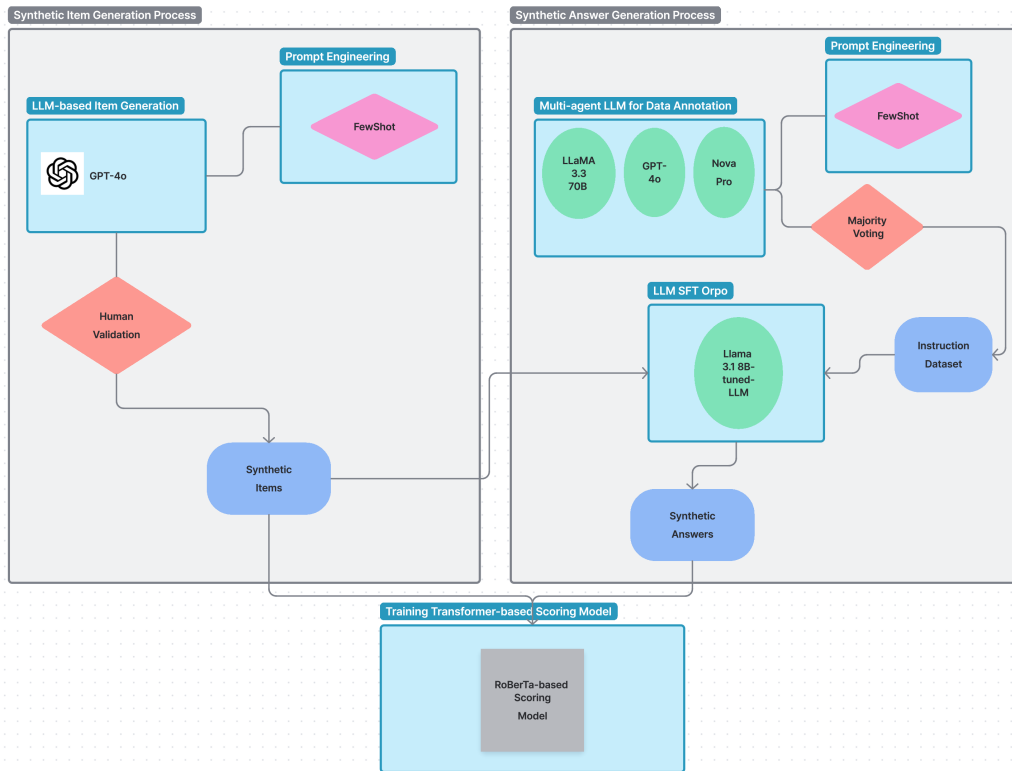


Figure 1: Pipeline for automated item and synthetic answer generation in the adaptive writing assessment

cinct writing prompt (no more than 25 words). To augment these examples with the topic as further context in the few-shot prompt, GPT-4o was applied to generate the topic for these hand-picked examples. The resulting triplets of example writing prompt, proficiency level and topic constituted our set of few-shot examples.

To generate a new writing item, we specified the desired topic(s) and proficiency level as part of the input for the LLM. A few semantically similar but diverse set of examples were chosen from our manually curated collection using LangChain’s Max-MarginalRelevanceExampleSelector (LangChain, 2025). These chosen examples were then passed into the LLM as a part of the input prompt to generate a new item based on the requested topic and proficiency level, as shown in Figure 2.

We ensured a wide coverage of content and linguistic complexity across all proficiency bands by systematically generating new items across multiple combinations of topic and proficiency level.

Five language assessment experts were asked to judge a sample of 100 prompts based on appropriateness for assessment regarding the following metrics: clarity, curriculum fit, and difficulty level. In addition, to further evaluate item quality, we com-

pared expert annotations on both difficulty level and topic with GPT-4o’s predictions, finding a correlation of nearly 0.9. This high level of agreement suggests that GPT-4o is able to closely approximate expert judgment in these qualitative aspects.

The use of LLMs and enabled rapid and scalable item generation, whilst retaining strict quality control through expert review, allowing the assessment to expand to new topics and levels efficiently and reliably, as recommended in recent work on few-shot prompting in language assessment contexts (insert citation).

3.3 Phase 2: Synthetic Generation of Training Data

In this phase, the aim was to generate a high-quality training set to generate responses for the new items produced in Phase 1. The use of ORPO in the next stage requires pairs of good and bad student responses for each item, and hence we require a way to assess the quality of generated responses to produce these pairs of examples. To do so, we first evaluated several available LLMs of different architectures and sizes for its ability to rate student responses. Each model was assessed for its consistency and reliability in assigning grammatical

```

Prompt:
Imagine you are a language teacher writing essay question prompts for an English
level written test. Given the level and topic, write a short prompt (max 25 words)
for the student. The prompt should be succinct and appropriate.

Examples:
Topic: Daily life
Level: 4
Prompt: Describe your daily routine.

Topic: Work, Company policies
Level: 6
Prompt: Your boss has asked for your help with the office dress code policy.
What rules do you suggest?
...

```

Figure 2: Illustration of a few-shot prompting template used for automated writing prompt generation.

accuracy scores to (item, response) pairs using a calibration set drawn from real test data. We measured agreement between each model and expert human annotations using Cohen’s Kappa statistic. Based on these preliminary experiments, we then selected the three LLMs that demonstrated the highest inter-annotator agreement with human raters as well as amongst themselves to perform a majority vote over the quality of the synthetic student response. This enabled us to then produce a larger dataset of pairs of student responses that can be used in the next phase of response generation.

For the annotation process, each selected LLM was first provided with a few-shot prompt comprising the grammatical accuracy scale (0–4) and multiple labeled examples. Each model independently assigned an accuracy score to every response, leveraging the internalized patterns from the few-shot instruction. Majority voting was then applied to the three scores produced for each sample, retaining the class most frequently assigned as the final label.

To ensure the highest possible data quality, we filtered the resulting dataset to retain only the samples where annotator agreement was strongest—either full consensus or clear majority among the three LLMs. This approach allowed us to construct a robust, reliable, and well-calibrated instruction dataset for producing realistic student responses via subsequent model fine-tuning and evaluation.

To evaluate the quality and reliability of the annotation process, we created an evaluation set (gold standard) consisting of approximately 200 (item, response) pairs. Each of these samples was independently annotated for grammatical accuracy, on a scale from 0 to 4, by five expert human raters. Only those samples with an inter-annotator agreement

above 70% were retained, ensuring a high level of reliability in the ground truth annotations.

This gold standard dataset was then used to benchmark each candidate LLM’s annotation performance. For the comparison, we calculated the Cohen’s Kappa score between the accuracy levels assigned by each LLM and the ground truth established by human annotations. The LLMs evaluated in this process included Llama 3.3 70B (Touvron et al., 2024), Nova Pro and Nova Small (AWS proprietary models²), Mistral Small and Mistral Large (Jiang et al., 2023), Claude Opus (Anthropic, 2024), and GPT-4o (OpenAI, 2024).

This systematic comparison enabled us to identify the models with the highest alignment to expert human judgments, guiding the selection of annotators for the synthetic data generation pipeline.

LLM Candidate	CK
LLaMA 33 70B	0.86
GPT-4o	0.87
Nova Pro	0.78
Mistral Small	0.18
Mistral Large	-0.01
Claude Opus	-0.08
Ensemble	0.89

Table 1: Cohen’s Kappa agreement between each LLM and gold standard human annotation

We selected the three best-performing LLMs, LLaMA 33 70B, GPT-4o, and Nova Pro as shown in Table 1, to form an ensemble for the majority voting procedure. Notably, this combination achieved a Cohen’s Kappa of 0.89 with the human-annotated gold standard, outperforming any individual model. This result demonstrates that major-

²Technical details available in the AWS Bedrock documentation: <https://docs.aws.amazon.com/bedrock/latest/userguide/model-parameters-nova.html>.

ity voting among top-performing models further increases annotation reliability and brings machine annotation closer to human-level agreement.

Finally, we generated our synthetic dataset of 20,000 pairs of responses, employing our ensemble of LLMs for the rating process to ensure that each pair consisted of one higher accuracy response and one lower accuracy response with the associated grammatical accuracy and proficiency level.

3.4 Phase 3: Fine-tuning of LLM-Based Response Generator

In this phase, we focused on enhancing the quality and proficiency alignment of synthetically generated responses by fine-tuning a large language model. We selected LLaMA 3.1 (8B parameters) as the base model for fine-tuning, utilizing the Optimized Reward Preference Optimization (ORPO) technique. The training data comprised approximately 20,000 synthetic samples generated during Phase 2, each annotated for grammatical accuracy and proficiency level.

Fine-tuning was conducted using a distributed training setup on the SageMaker infrastructure, employing the following configuration:

- **Instance type:** ml.g5.12xlarge
- **Environment:** PyTorch 2.5.1, GPU, CUDA 12.4, Ubuntu 22.04
- **Batch size:** 8 per device
- **Gradient accumulation steps:** 1
- **Learning rate:** 2×10^{-4}
- **Number of epochs:** 3
- **LoRA settings:** $r = 8$, $\alpha = 16$, dropout=0.1
- **Seed:** 42 (for reproducibility)

Model training was orchestrated with distributed computing support (Torchrun) to fully leverage available GPU resources, and checkpointing mechanisms were in place to ensure reliability.

Through this fine-tuning process, the LLaMA 3.1 model was adapted to generate candidate responses at specific proficiency levels, closely mimicking real student outputs in both accuracy and variety. The resulting model serves as a robust response generator for subsequent scoring model development and evaluation within the adaptive writing assessment pipeline.

3.5 Phase 4: Training Adaptive Transformer-Based Scoring Model

In the final phase, we aimed to robustly integrate the newly generated writing items into our automated scoring pipeline. To achieve this, we focused on the domain of education, and manually composed approximately thirty new writing prompts covering a broad range of proficiency levels, as generated during Phase 1.

For each prompt, the fine-tuned LLaMA 3.1 (8B) response generator was used to synthesize approximately 200 sample answers at varying proficiency levels. This resulted in a substantial and well-stratified dataset representing a wide spectrum of student abilities.

We then fine-tuned our RoBERTa-based transformer scoring model, training it on a combination of both initial (pre-existing) and newly generated items and responses. This approach was designed to ensure a smooth integration of new items into the scoring system while maintaining performance on established items.

The model was trained using the following hyperparameters with the TrainingArguments setup:

- **Evaluation and save strategy:** every 200 steps
- **Batch size:** 16 per device
- **Learning rate:** 2×10^{-5}
- **Warmup ratio:** 0.1
- **Epochs:** 6
- **Weight decay:** 0.01
- **Learning rate scheduler:** linear
- **Mixed precision training:** enabled (fp16)

This fine-tuning procedure enables the scoring model to generalize to new adaptive items and proficiency levels while ensuring reliable and consistent automated assessment performance.

4 Experiment

4.1 Psychometric Analysis

4.1.1 Dataset

We constructed a dataset containing 500 responses to approximately 50 different writing prompts, with

Level	Prompt	Synthetic Response	Label (proficiency)
2	What activities do you do at school?	I am studying english at university. I love english and talk with my friend, we share knowledge.	1
3	Describe your teaching style.	I like to teach students about english and use example and video for help them learn easy.	1
3	What do you think makes a good teacher?	A good teacher possesses patience, empathy, and effective communication skills. They foster a supportive environment, encourage critical thinking, and adapt teaching methods to cater to diverse learning styles, promoting academic growth and personal development in their students.	4
5	Explain the importance of extracurricular activities in a student's overall development.	Ggghhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh	0
5	Describe a typical school day, including your classes, activities, and any special events you participate in.	My school day starts at 8am with class. I join clubs after and do volunteer work. Math is my favorite.	3

Table 2: Examples of synthetically generated items and responses on the topic of education, illustrating different levels of language proficiency.

an equal split between human-generated and automatically generated answers (250 each). Each response is scored on an ordinal scale from 0 to 4, providing a rich basis for psychometric analysis. This dual-sourced dataset enables us to directly compare human performance and large language model (LLM) behavior under similar assessment conditions.

4.1.2 Evaluation

To better understand the scoring dynamics and the comparability between synthetic and authentic responses, we conducted an Item Response Theory (IRT) analysis. Figure 4.1.2 presents average Item Characteristic Curves (ICCs) derived from the dataset, shown separately for human and LLM-generated responses. Each curve reflects the probability of achieving at least a given score threshold as a function of modeled proficiency, averaged across all items.

The ICCs for both human and synthetic responses reveal similar shapes and threshold spacing, indicating that LLM-generated answers closely emulate the probability distributions observed in real student performance. This suggests that synthetic responses can serve as effective proxies for actual learner data in calibrating and evaluating automated scoring models.

4.2 Pre-piloting Study

To validate the integration and quality of the newly generated items, a pre-piloting study was conducted with approximately 250 participants in Rwanda. The main objective of this phase was

to compare the performance and acceptability of the newly generated items. Participants completed a test composed of a balanced mix of traditional (previously validated) items and automatically generated new items. The distribution of items was designed to ensure diversity in both content and difficulty levels.

4.2.1 Comparative Analysis

To assess the effectiveness of the automatically generated items, we conducted a comparative analysis of success rates between old and newly generated items using statistical significance testing. For each item, we computed the difference in success rates and tested for significance using a z-test for proportions. To ensure a sufficient number of items per analysis group, we grouped the original 16 difficulty levels into 6 broader categories, thereby increasing the number of items per test group for more robust statistical analysis.

Table 3: Statistical Comparison: z-test and p-value for Each Item Level

Item-Level Group	z-score	p-value
1	4.35	0.001*
2	1.33	0.182
3	0.99	0.323
4	1.08	0.285
5	-1.96	0.056
6	-1.86	0.060

* Statistically significant at $p < 0.05$.

The results presented in Table 3 show that although most items did not show significant differences, the only statistically significant effects ($p < 0.05$) were observed between items in dif-

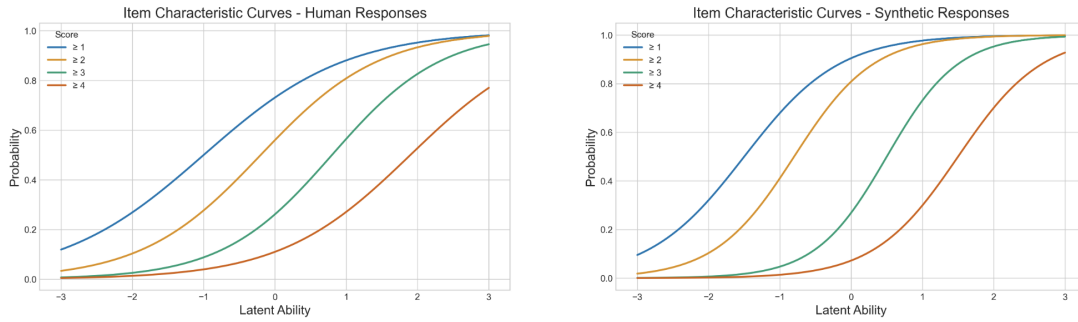


Figure 3: Average Item Characteristic Curves from IRT analysis of human and synthetic responses.

ficulty group 1 (the easiest items). For items belonging to difficulty group 5, p-values were close to the significance threshold, indicating borderline significance. These observations underscore the importance of careful quality control, particularly at both extremes of item difficulty, when integrating automatically generated items into assessments.

4.2.2 Item Characteristic Curve Analysis

To further assess the psychometric properties of both traditional and automatically generated items, we performed an Item Characteristic Curve (ICC) analysis using the Two-Parameter Logistic (2PL) model from Item Response Theory (IRT) (Baker and Kim, 2004). The 2PL model estimates two main parameters for each item: the difficulty parameter (indicating the level of ability required for a 50% probability of a correct response) and the discrimination parameter (reflecting how well the item differentiates between participants of differing ability levels).

For each item, we fitted the 2PL model using the participants' response data. T

Figure 4 displays the ICCs for three representative synthetic items extracted from the assessment. Each curve presents the probability of a correct response ($P(\theta)$) as a function of participant ability (θ), and the items were chosen to illustrate a range of difficulty and discrimination parameters.

- **Easy item:** This item is answered correctly by participants even at lower ability levels. The steep, less rounded shape of the ICC indicates a high discrimination parameter, meaning the item sharply differentiates between participants just below and just above its difficulty threshold.
- **Medium item:** This item requires a higher ability level for a 50% probability of a correct answer, suggesting moderate difficulty. It also

exhibits high discrimination, as seen in the sharp transition.

- **Difficult item:** This item is considerably harder and is only likely to be answered correctly by participants with the highest abilities. The more gradual slope of its ICC suggests a lower discrimination parameter compared to the other items.

These three examples demonstrate both the range of difficulty present in the test and the variation in item discrimination. Such diversity ensures that the assessment can reliably differentiate participants across a broad spectrum of ability levels.

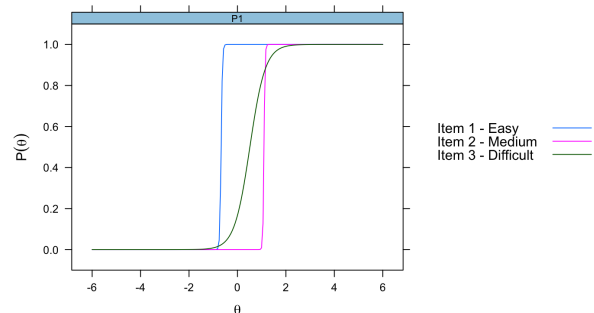


Figure 4: Item Characteristic Curves for 3 synthetic items.

4.3 Scoring Model Evaluation

4.3.1 Dataset

To assess the impact of synthetic items on model training, we conducted controlled experiments using both authentic and synthetic data. The evaluation was performed on a fixed test set of 800 samples that included a balanced selection of prompts and responses, covering all proficiency levels and a wide range of topics.

4.3.2 Evaluation

We evaluated the proficiency classification task using three distinct transformer-based models, each fine-tuned on our training data. First, we included two widely used traditional encoders: **BERT-base-uncased** (Devlin et al., 2018) and **RoBERTa-base** (Liu et al., 2019). Both are pre-trained bidirectional transformers and have served as robust baselines for a range of NLP classification tasks. Second, we fine-tuned **Flan-T5 Base** (Longpre et al., 2023), an instruction-tuned sequence-to-sequence model with strong generalization abilities for text-to-text tasks, adapting it specifically for multi-class classification by framing the label prediction as sequence generation.

Table 4 summarizes the precision, recall, and F1-scores obtained by each model, macro-averaged across proficiency levels. RoBERTa shows the strongest overall performance (macro F1-score of 0.82), illustrating the benefits of its more advanced pre-training. BERT achieves good results but slightly lower than RoBERTa, consistent with prior findings in classification tasks. Notably, Flan-T5 Base also provides competitive performance (macro F1-score of 0.80), demonstrating the viability of adapting generative models to classification through prompt engineering and sequence-based fine-tuning.

Model	Precision	Recall	F1-score
BERT-base-uncased	0.80	0.77	0.78
Flan-T5 Base	0.81	0.80	0.80
RoBERTa-base	0.83	0.81	0.82

Table 4: Macro-averaged precision, recall, and F1-score for each fine-tuned model on proficiency classification.

5 Conclusion

In this work, we introduced an end-to-end automated pipeline for adaptive English writing assessment, leveraging recent advancements in large language models for both item generation and synthetic data annotation. Our methodology utilizes few-shot prompting, robust majority-vote labeling, and transformer-based scoring to efficiently generate, calibrate, and evaluate new writing tasks within a psychometrically-sound framework. Extensive experiments demonstrate that the proposed system achieves high agreement with expert evaluations, ensuring both the validity and scalability required for operational proficiency testing. We anticipate that this approach will provide a solid foundation

for future research on data-driven adaptive assessment and the broader application of LLMs in language testing.

Limitations

Automated scoring models risk perpetuating biases, particularly across demographic groups, language proficiencies, or socio-cultural contexts. The use of synthetic data and automated generation may also introduce or reinforce unintended patterns, potentially affecting educational fairness. To mitigate these risks, it is vital to incorporate diverse training data, implement human-in-the-loop evaluations, and regularly audit system performance. We regularly monitor test quality through ongoing psychometric analyses and expert human evaluation. This process ensures that both automated item generation and scoring maintain high standards of validity and reliability over time.

Furthermore, the introduction of a substantial number of new items into the assessment pool needs large-scale psychometric analysis to fully evaluate their functioning and impact. We acknowledge this as an essential next step, and plan to conduct comprehensive studies to further validate the psychometric properties of these newly introduced items across diverse populations and contexts.

References

- Anthropic. 2024. Introducing claude 3. <https://www.anthropic.com/news/claude-3-family>.
- Frank B Baker and Seock-Ho Kim. 2004. *Item response theory: Parameter estimation techniques*. CRC press.
- Isaac I. Bejar and 1 others. 2003. A template-based approach to the generation of test items. In *Principles and Practice in Automated Item Generation*.
- Tom B. Brown and 1 others. 2020. [Language models are few-shot learners](#). *NeurIPS*.
- Tom V. et al. Brown. 2023. [The duolingo english test interactive writing task](#). In *Proc. BEA Workshop at ACL 2023*.
- Christopher et al. Clark. 2021. Generating synthetic data for improved accuracy in educational nlp tasks. *arXiv preprint arXiv:2106.05071*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Mark J. Gierl and Thomas M. Haladyna. 2012. *Item Generation for Test Development*. Routledge.
- Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*.
- Zheng-Xin Jiang and 1 others. 2023. [Mistral: A simple yet effective baseline for instruction-tuned large language models](#). *arXiv preprint arXiv:2310.06825*.
- Maryam et al. Kurdi. 2023. [Controlled generation of assessment items for educational applications using language models](#). In *Proc. BEA Workshop at ACL 2023*.
- LangChain. 2025. Few shot prompt template. https://python.langchain.com/api_reference/core/prompts/langchain_core.prompts.few_shot.FewShotPromptTemplate.html.
- Ximing et al. Liu. 2023. [G-eval: Nlg evaluation using gpt-4 with better human alignment](#). *arXiv preprint arXiv:2303.16634*.
- 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*.
- Shayne Longpre, Christopher Patil, Abhay Rao, Albert Webson, Le Hou, Pengfei Liu, and 1 others. 2023. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Zeyi et al. Mai. 2022. [Towards generalizable essay scoring with language models and augmented data](#). In *NAACL 2022*.
- Elizabeth Mayfield and Alan W. Black. 2020. [Automated scoring of written essays with transformer models](#). In *Proc. BEA at ACL 2020*.
- Kamel Nebhi and György Szaszák. 2023. Automatic assessment of spoken english proficiency based on multimodal and multitask transformers. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 769–776.
- OpenAI. 2024. [Gpt-4o technical report](#). *arXiv preprint arXiv:2405.16408*.
- Hugo Touvron, Thibaut Lavril, and 1 others. 2024. [Llama 3: Open foundation and instruction models](#). *arXiv preprint arXiv:2404.14219*.
- Yizhong Wang, Zhiwei Zhang, Zexuan Zhong, Zhe Gan, Jingjing Liu, and Noah A. Smith. 2024. [Simulating candidates in educational assessment with large language models](#). *arXiv preprint arXiv:2401.07043*.
- Luke Williams and 1 others. 2022. [The duolingo english test: 2022 technical report](#). In *arXiv preprint arXiv:2206.01056*.
- Xuechen Yao, Ankur P. Parikh, Noah Constant, Dwi Susanti, and Heng Ji. 2024. [Leveraging llm-respondents for item evaluation: a psychometric analysis](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1602–1628.
- Xu Zhai, Xiang Wan, Qian Jin, and Eduard Hovy. 2023. [Automatic generation of language assessment tasks using large language models](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 340–350.
- Diyi Zhang and 1 others. 2022. [Automatic generation of factual reading comprehension questions with large language models](#). In *Proceedings of ACL 2022*.

A Framework for Proficiency-Aligned Grammar Practice in LLM-Based Dialogue Systems

Luisa Ribeiro-Flucht^{1,2} Xiaobin Chen^{1,2} Detmar Meurers^{1,3}

¹ LEAD Graduate School and Research Network, University of Tübingen, Germany

² Hector Research Institute of Education Sciences and Psychology,
University of Tübingen, Germany

³ Leibniz-Institut für Wissensmedien, Tübingen, Germany
{luisa.ribeiro-flucht,xiaobin.chen}@uni-tuebingen.de
d.meurers@iwm-tuebingen.de

Abstract

Communicative practice is critical for second language development, yet learners often lack targeted, engaging opportunities to use new grammar structures. While large language models (LLMs) can offer coherent interactions, they are not inherently aligned with pedagogical goals or proficiency levels. In this paper, we explore how LLMs can be integrated into a structured framework for enabling goal-oriented, grammar-focused interaction, building on an existing dialogue system. Through controlled simulations, we evaluate five LLMs across 75 A2-level tasks under two conditions: (i) grammar-targeted, task-anchored prompting and (ii) the addition of a lightweight post-generation validation pipeline using a grammar annotator. Our findings show that template-based prompting alone substantially increases target-form coverage up to 91.4% for LLaMA 3.1-70B-Instruct, while reducing overly advanced grammar usage. The validation pipeline provides an additional boost in form-focused tasks, raising coverage to 96.3% without significantly degrading appropriateness.

1 Introduction

Second language acquisition (SLA) is driven by frequent and meaningful language use (Behrens, 2009; Ellis, 2002; Canale and Swain, 1980). While second language (L2) learners develop comprehension skills through input-rich activities, many struggle to find opportunities outside of the classroom to meaningfully produce what they’ve learned, especially in the early stages (Ortega and DeKeyser, 2007). This is particularly true for A2-level learners, who are using the L2 in social interaction more consistently, as described by the Common European Framework of Reference Companion Volume (CEFR; Council of Europe, 2020).

Grammar often represents an obstacle in L2 production, with certain forms proving persistently

difficult to master (Ellis, 2017). As a result, learners tend to avoid challenging forms in spontaneous communication, where conveying meaning quickly takes priority (Lyser and Sato, 2013). To mitigate this issue, research emphasizes the importance of communicative practice that targets learners’ specific needs and occurs iteratively, rather than relying on decontextualized drills. Without such targeted support, many A2 learners tend to plateau, struggling to transfer classroom grammar knowledge to real-life communication (Richards, 2008; Mirzaei et al., 2017; Lightbown, 2007).

Conversational agents, or chatbots, have been proposed as a way to offer students language production opportunities (Sydorenko et al., 2018). Early rule-based systems enabled predictable, level-appropriate dialogues, but their design was labor-intensive and resulted in rigid, limited interactions (Bibauw et al., 2022). More recently, LLMs have emerged as a promising alternative, as they are capable of generating coherent and fluent language without the need for manual scripting. However, their stochastic nature often leads to inconsistent pedagogical alignment (Zhou et al., 2023; Benedetto et al., 2025).

While several studies have explored the use of out-of-the-box LLMs for educational purposes, such as linguistic feedback, role play, and adaptivity (Borchers and Shou, 2025; Gervás et al., 2025; Fincham and Alvarez, 2024), their inherent unpredictability poses challenges for aligning output with pedagogical frameworks, adaptive logic, and real-world scalability. In the absence of a systematic mechanism for constraining both communicative context and grammatical targets, learners may engage in practice that lacks the individualization and progression necessary for effective development (Ruiz et al., 2023).

To bridge this gap, we introduce a parametric framework for goal-oriented, CEFR-aligned grammar practice through LLM-mediated dialogue.

Building on AISLA, a rule-based system developed for grammar instruction among German seventh-graders (Chen et al., 2022), this A2-level extension links each task to the English Grammar Profile (EGP; O’Keeffe and Mark, 2017). EGP targets are embedded into dynamically generated prompt templates that pair specific grammatical forms with communicative scenarios, enabling repeated practice of the same structure across varied, context-rich tasks. This approach aims to maintain pedagogical consistency while leveraging the flexibility and fluency of LLMs.

In this paper, we investigate the effectiveness of our approach by asking the following research questions:

1. Can out-of-the-box LLMs generate goal-oriented dialogues that spontaneously target specific grammatical structures?
2. How effective is task-supported prompting in guiding LLMs to produce multi-turn, A2-level outputs aligned with target grammar?
3. To what extent can LLMs of different sizes maintain coherence and target specific grammatical structures in task-supported dialogues?
4. Does incorporating a grammar validation component improve target structure usage and CEFR alignment?

2 Goal-oriented Grammar Practice

While theoretical perspectives in SLA vary, many contemporary approaches increasingly view grammar as a functional and adaptive component of language use, rather than a fixed body of rules (Diessel, 2019; Dik, 1981). Larsen-Freeman (2003) advocates for the reframing of grammar as a dynamic skill encompassing three interrelated dimensions: (i) form, structural features of language; (ii) meaning, the semantic or propositional content these forms convey; and (iii) use, the pragmatic and discourse functions that guide when and why particular forms are selected in context.

To support this dynamic view of grammar, instructional design should align the learning environment with real communicative demands. Transfer-appropriate processing (TAP; Lightbown, 2007) reinforces this idea, suggesting that learning is most effective when the cognitive processes involved

during learning mirror those required during retrieval and use. Grammar practice then should not be decontextualized, rather, learners need iterative opportunities to use target structures in activities that closely resemble authentic language use.

Furthermore, research shows that specific grammar structures are best acquired through activities that naturally elicit their use (Loschky and Bley-Vroman, 1993; Faitaki and Murphy, 2019; Lyster and Sato, 2013). Frameworks such as Task-Based Language Teaching (TBLT; Nunan, 2004) and Task-Supported Language Teaching (TSLT; Ellis, 2024) operationalize this idea by embedding language practice within communicative tasks. In TSLT, for example, the syllabus is organized around specific linguistic units, which are practiced through meaning-oriented tasks: activities in which language is used to achieve a non-linguistic goal, such as comparing options or making plans. Unlike traditional drills, such tasks promote interaction in which the target structure is functionally relevant. Our system builds on this approach, using task templates that integrate grammar targets with communicative goals (Bear et al., 2024).

3 Controlled Text Generation

To integrate a multidimensional view of grammar into LLM-based applications, developers must find ways to control the output of these models in order to deliver scaffolded, targeted practice without sacrificing meaning or use. Although emerging approaches offer potential solutions, such as prompting techniques and finetuning techniques, they typically lack explicit goal orientation and have not been systematically applied to grammar-focused learning tasks.

In the context of open-ended dialogue, some approaches have aimed to implicitly steer LLMs toward producing predetermined grammatical structures. For instance, Okano et al. (2023) compare reinforcement learning-based fine-tuning of DialoGPT with few-shot prompting of GPT-3, finding that both methods can enable grammatical control, with reinforcement learning achieving greater precision. Similarly, Glandorf et al. (2025) evaluate prompting, fine-tuning and decoding strategies for the inclusion of predetermined EGP structures during open-ended chat, showing that grammar-controlled decoding with LLaMA 3.3 effectively targets specific forms, albeit with a slight reduction in response quality. However, both studies focus

exclusively on the inclusion of target structures in the next response only, not evaluating model performance across multi-turn interactions.

Engaging LLMs in multi-turn conversations introduces additional challenges, as the model must track and integrate longer contextual information to maintain coherence and relevance across turns (Yi et al., 2024). This challenge becomes more complex when there is a pedagogical task to adhere to and a grammar structure to target. While some recent work explores different applications of LLM-mediated language learning (Tyen et al.; Méndez and Bautista, 2025), no approach, to our knowledge, has attempted to integrate LLMs within goal-oriented dialogue systems for systematic, targeted grammar practice.

Our work therefore intends to move towards bridging these divides by integrating a CEFR-aligned proficiency framework, generating task-based dialogue data and embedding real-time grammar scaffolding into an LLM-powered dialogue system. In doing so, we aim at combining the naturalness of large-scale language models, with the pedagogical basis of goal-oriented, task-supported instruction.

4 Implementation

4.1 System Description

The AISLA system was built using a Java-based backend with a PostgreSQL¹ database and an Android-based frontend². Its backend follows a modular, service-oriented design for functionalities such as text-to-speech and automatic speech recognition. The chatbot functionality is handled via AWS Lex³, a slot-filling dialogue management service, which requires manual dialogue scripting. Although effective for rule-based interactions, especially in school contexts, where content control is a priority (Wilske, 2015), this configuration presents limitations for the integration of personalized and adaptive features.

To support the requirements of this research, several major architectural changes were introduced. First, the following changes were made to accommodate LLM-based interactions: AWS Lex was replaced by LLM APIs, and a DialogueManager class was added to orchestrate prompt chaining and dialogue state management. Second, the Android

frontend was replaced with an Ionic⁴ one to ensure broader accessibility across platforms, thereby increasing inclusivity in participant recruitment and usage scenarios. Additionally, the EGP was integrated in the grammar task design.

4.2 Task Bank

To support modularity and future scalability, a task bank was implemented as a database table. Each entry is linked to a target grammar structure and a communicative purpose, including fields such as the EGP structure's guideword, can-do statement, the name and format of the task and its instructions. Task names refer to real-life situations where grammar structures are employed for communicative purposes, for instance, "Discussing cultural differences between two countries", "Telling someone about a historical monument" or "Picking between two places to go to".

The task design is based on the Gramming framework (Larsen-Freeman, 2003), accounting for the three dimensions of grammar. Accordingly, three task types were developed:

Q&A (form-focused): These tasks are meant to provide high-frequency exposure to a target grammar structure in each of the model's turns. It aims to use and elicit the structure in short question-answer exchanges (e.g. answering questions about one's daily routine with frequency adverbs).

Information gap (meaning-focused): These tasks emphasize the meaningful application of grammar structures, encouraging learners to make decisions about where and how to use the target structure in context, usually leveraging external resources like tables, charts and images (e.g. reporting on what was said in an interview with reported speech, explaining what someone looks like using adjectives).

Role play (use-focused): These tasks situate grammar practice in realistic scenarios (e.g. giving a friend advice with modal verbs, asking for directions with prepositions). They are designed to simulate real-life situations where the structure must be used appropriately within a given social or functional context.

¹<https://www.postgresql.org/>

²<https://www.android.com/>

³<https://aws.amazon.com/lex/>

⁴<https://ionicframework.com/>

4.3 Dialogue management

When a task is initiated by the student, information from the task bank is dynamically retrieved from the database and inserted into a template-based LLM prompt. An example prompt schema can be found in Appendix A. Task duration is currently managed via turn count. This means that each dialog task spans a predetermined number of turns by default, after which the learner is given the option to either conclude the task or continue practicing.

5 Method

Since the purpose of this study was to evaluate how well different LLMs perform in task-supported dialogues, conducting a user study was considered premature. Instead, to simulate varied learner interactions, each model was paired with ChatGPT-4o, using a temperature setting of 0.5 to introduce some content variability on the student side.

To test the robustness of the model acting as the tutor, three learner behavior patterns were implemented in every task: (1) in the first run, the model was instructed to make grammatical mistakes; (2) in the second run, a hard-coded clarification request ("What does that mean?") was injected; and (3) in the third run, a misunderstanding was introduced via the phrase "I don't know" at the second turn (c.f. Appendix F for snippets of different runs and task type). Each task was limited to 10 turns to ensure comparability across conditions.

We selected 75 tasks targeting 15 grammar super-categories drawn from the EGP, namely, adjectives, adverbs, clauses, determiners, future, modality, passives, past, prepositions, present, pronouns, verbs, questions, negation, and reported speech. The tasks were equally divided into Q&A, information gap, and role play formats. To account for output variability, each task was run five times, resulting in 375 dialogues, and 1875 messages per model⁵.

5.1 Experiment 1

We evaluated five large language models (LLMs) spanning a wide parameter range: Llama 3.1 8B-Instruct, Mistral-Small 3.1 24B-Instruct (Mistral AI, 2024), Llama 3.3 70B-Instruct (Meta, 2024), DeepSeek-V3 685B (DeepSeek AI, 2024), and GPT-4o (OpenAI, 2024), whose exact parameter count is undisclosed. Each model acted as the tu-

⁵All experiments and data mentioned in this work can be found in <https://github.com/luisards/grammar-practice-framework>

tor in the 75 tasks. The decoding temperature was fixed at 0.0 for decreased variability.

To isolate the effect of explicit grammatical scaffolding (RQ1), we first used *task-only prompting*, satisfying the TSLT requirement of a clear non-linguistic goal. In this setting, the prompt contained only the task name plus minimal instructions, with no mention of the target grammar structure (c.f. Appendix B).

The second part of Experiment 1 introduced our template-based prompt that embeds the communicative goal together with A2-level grammar cues. We ask whether this prompt improves alignment and whether model size modulates any gain (RQ2-RQ3).

5.2 Experiment 2

Experiment 2 adds a lightweight control pipeline. We integrated POLKE (Sagirov and Chen, 2025), an EGP-based grammar annotator, as a post-generation validator. For every tutor turn, POLKE tagged all grammar structures and their CEFR level; a one-shot rephrase is triggered when (i) any structure above B1 is present (Appendix C) or (ii) in Q&A tasks, the required target structure is missing (c.f. Appendix D). Only one rewrite pass is allowed to bound latency and prevent loops. The control loop was applied only to the three best-performing models from Experiment 1 (Llama 3.3 70B, DeepSeek V3, GPT-4o).

6 Evaluation

We combine two quantitative metrics, obtained via POLKE annotations, with one qualitative metric obtained from human ratings.

Target structure presence A binary metric which measures whether the tutor turn contains the grammar form specified in the task (crucial for Q&A).

Proficiency alignment Defined here as the use of grammar within the target CEFR range. This metric refers to the proportion of structures above the B1 ceiling (i.e. B2, C1, C2).

Response quality Appropriateness on a 5-point scale (factual accuracy + contextual coherence). Fifteen native or near-native English speakers recruited through Prolific⁶ rated six dialogues per model (450 tutor turns). The rubric and anchors appear in Appendix E.

⁶<https://www.prolific.com>

To probe rubric interpretability, GPT-4o scored the same 30 dialogues. Its item-level scores correlate moderately with the human mean (Spearman $\rho = .49, p < .01$) and reproduce the system rank order ($\rho = .67$). A separate GPT-4o pass over the full 75-task set is released for replication in the shared repository.

7 Results

In this section, we report the results of three experimental conditions: baseline task-only (B), prompt + grammar scaffold (P) and prompt + scaffold + validation (P+V), distributed by metric.

7.1 Dialogue-quality ratings

Table 1 shows the mean human appropriateness ratings. Larger models outperform smaller ones across all conditions. Prompting incurs a small drop for every model (max 0.6 points for Mistral-Small). Validation restores or slightly improves quality for the top systems. Inter-rater agreement was found to be moderate (Krippendorff $\alpha_{\text{ordinal}} = .42$; rises to .45 with GPT-4o added).

Smaller models were excluded on the basis of a post-hoc, exploratory cutoff: any model whose mean human appropriateness rating fell below 4.0, corresponding to the “somewhat appropriate” anchor on our 5-point rubric was deemed pedagogically unviable and therefore did not get included in experiment 2.

Model	B	P	P+V
Llama 3.1	3.9	3.6	–
Mistral-Small	3.5	3.3	–
Llama 3.3	4.9	4.3	4.7
DeepSeek V3	4.7	4.4	4.4
GPT-4o	4.5	4.5	4.6

Table 1: Mean human appropriateness ratings (1-5).

7.2 Proficiency alignment

Table 2 reports the share of grammar at or below B1. At baseline, a chi-square test across the five models is significant ($\chi^2 = 59.4, df = 4, p < 10^{-11}$) but the practical effect is small (Cramér $V = .04$). Prompting pushes every model above 98% basic grammar; validation halves the residual advanced usage.

7.3 Target-structure coverage (form-practice tasks)

Prompting boosts target-structure inclusion from roughly 30% to 70-91% (Table 3). Validation yields a further 5-11-point gain (96.3 % for Llama 3.3 70B, 95.0% for DeepSeek V3, 91.5% for GPT-4o). A Pearson chi-square on the Q&A subset confirms significant model differences at the prompt stage ($\chi^2(2, N = 1,875) = 33.1, p < 10^{-7}, V = .13$). After validation the gap narrows but remains significant ($\chi^2 = 19.5, p < .001, V = .10$).

8 Discussion

Our findings reveal that while large language models (LLMs) are capable of generating fluent, goal-oriented dialogues, they do not reliably produce the intended grammatical structures without explicit guidance. Answering RQ1, at baseline, models demonstrated stronger appropriateness, with bigger models reaching average ratings between 4.5 and 4.8, but the presence of the target grammatical structure was limited, appearing in only 28-39% of tutor turns in form-practice tasks.

RQ3 explored how model size influence the ability to maintain coherence and grammatical focus in task-supported dialogues. Our findings suggest that model capacity matters. Larger models (70B-685B) retained higher appropriateness (4.3 - 4.7) and needed fewer rewrites, confirming that scale confers stronger control. Yet the scaffolded prompt significantly narrowed the grammar gap, even though their final appropriateness remained lower ($\approx 3.6 - 3.3$). This trade-off invites a cost-benefit choice: institutions with limited resources may be able to achieve near-large-model grammar fidelity at a fraction of the compute cost, accepting a decrease in perceived dialogue polish.

Concerning RQ4, a post-hoc validation pass halved the residual advanced grammar usage ($\chi^2(2) = 35.1, V = .10$), confirming its value as a safety net when level control is non-negotiable. However, quality gains plateau once a strong prompt is in place, suggesting diminishing returns for additional automated checks.

Finally, it is important to acknowledge the fact that strict structure enforcement must be balanced against the spontaneity of genuine dialogue: real learners will redirect, clarify and digress. Designing tasks that preserve communicative authenticity while guaranteeing exposure to a focal form remains an open challenge, especially at higher or

Model	B		P		P+V	
	\leq B1	$>$ B1	\leq B1	$>$ B1	\leq B1	$>$ B1
Llama 3.1	92.4	7.6	98.6	1.4	–	–
Mistral-Small	93.6	6.4	92.8	7.2	–	–
Llama 3.3	93.0	7.0	98.6	1.4	99.4	0.6
DeepSeek V3	91.5	8.5	98.2	1.8	99.1	0.9
GPT-4o	92.4	7.6	98.9	1.1	99.4	0.6

Table 2: Percentage of grammar structures at or below B1 (\leq B1) and above B1 ($>$ B1).

Model	B	P	P+V
Llama 3.1	39.0	78.4	–
Mistral-Small	28.0	69.4	–
Llama 3.3	37.0	91.4	96.3
DeepSeek V3	34.2	87.8	95.0
GPT-4o	35.5	80.5	91.5

Table 3: Tutor turns that contain the requested grammar structure (25 form-practice tasks).

lower CEFR targets where our A2-centric template may not directly transfer.

9 Conclusion and Outlook

This paper explores the integration of LLMs into a goal-oriented dialogue system for A2-level grammar practice. Our results suggest that when dialogues are grounded in pedagogically designed prompts, proficiency alignment converges across models of different sizes. While these findings are promising, they remain preliminary and based on controlled simulation rather than real learner input.

Larger models (e.g., LLaMA 3.3 70B, GPT-4o, DeepSeek V3) maintained grammatical focus and dialogue coherence more reliably, particularly under conversational pressure. However, prompting alone was sufficient to bring smaller models (e.g., Mistral-Small) closer to the target structure usage rates observed in larger systems. This indicates that instructional framing, not just model capacity, plays a critical role in shaping output toward pedagogical goals.

We also explored the impact of a lightweight post-generation validation step using POLKE, an EGP-based grammar annotator. While this step did not significantly alter overall CEFR alignment (which was already high under prompt conditions), it provided additional gains in target-form inclusion, particularly in Q&A tasks, where an increase

of approximately 10% was observed. These findings highlight validation as a useful safeguard for scenarios where structure-specific exposure is pedagogically important.

Taken together, our findings point toward two tentative design guidelines for developers of intelligent tutoring systems that incorporate LLMs: (i) Combining pedagogically-grounded prompts with CEFR-based post-generation validation may offer a feasible path toward controllable, targeted grammar practice; (ii) Model scaling should be guided by observable convergence in dialogue coherence and target-form density, which, based on our exploratory experiments, occurred at around 70 billion parameters.

Furthermore, because our evaluation relies on open CEFR descriptors and a publicly available annotator, the method remains applicable as new models are released. To support continued re-evaluation, we release all core components of our setup: the selected task bank, template prompt, and scoring script.

Beyond targeting grammatical forms, our results underscore the value of contextual control: grammar structures should appear not only accurately, but also in varied, goal-relevant settings. Our template-based prompting framework sets to achieve this by scaffolding interaction around communicative goals, potentially making it easier to support iterative practice, interest-driven adaptation and integration of learner modeling.

In future work, we intend to perform user studies and log learner use of support tools (e.g., on-demand L1 translation), engagement with different contexts and its interaction with learning outcomes. Finally, over time, interaction data collected from users will allow for the creation of authentic data, enabling LLM fine-tuning grounded in authentic learner behavior.

Limitations

While our framework demonstrates the potential of LLMs for proficiency-aligned grammar practice, several limitations must be acknowledged. First, our grammar validation relies on an automatic annotator, the robustness and coverage of which varies across structures. In experiment 2, the same annotator was also used for both controlling and evaluating the output, which could introduce bias into the results.

In addition, the system currently does not perform a formal grammar accuracy check beyond target form detection, meaning that in case the LLMs make errors, those may go unnoticed. Similarly, vocabulary control, although implicitly restricted through task design, is not externally validated against level-specific lexicons, which may impact lexical appropriateness for A2 learners.

Lastly, our evaluation remains system-focused and has not included learner interaction data. Without a user study, we cannot yet assess the pedagogical effectiveness, learner engagement or practical impact of the system in real-world settings. These areas will be addressed as the next step in this project.

Acknowledgments

This work was supported by the German Ministry of Education and Science (BMBF) under funding number 01IS22076. We also acknowledge the valuable contribution of the English Grammar Profile (EGP), which provided an invaluable resource used in this study.

References

- Elizabeth Bear, Xiaobin Chen, Daniela Verratti Souto, Luisa Ribeiro-Flucht, Björn Rudzewitz, and Detmar Meurers. 2024. *Designing a task-based conversational agent for EFL in German schools: Student needs, actions, and perceptions*. *System*, 126:103460.
- Heike Behrens. 2009. *Usage-based and emergentist approaches to language acquisition*. *Linguistics*, 47(2):383–411.
- Luca Benedetto, Gabrielle Gaudeau, Andrew Caines, and Paula Buttery. 2025. *Assessing how accurately large language models encode and apply the common european framework of reference for languages*. *Computers and Education: Artificial Intelligence*, 8:100353.
- Serge Bibauw, Thomas François, and Piet Desmet. 2022. *Dialogue systems for language learning: Chatbots and beyond*. In Nicole Ziegler and Marta González-Lloret, editors, *The Routledge Handbook of Second Language Acquisition and Technology*, page 15. Routledge, New York.
- Conrad Borchers and Tianze Shou. 2025. *Can large language models match tutoring system adaptivity? a benchmarking study*. *arXiv preprint arXiv:2504.05570*. Accepted as full paper to the 26th International Conference on Artificial Intelligence in Education (AIED 2025).
- Michael Canale and Merrill Swain. 1980. *Theoretical bases of communicative approaches to second language teaching and testing*. *Applied Linguistics*, 1(1):1–47.
- 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 *Proceedings of the International Conference on Artificial Intelligence in Education*, pages 582–585. Springer.
- Council of Europe. 2020. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment – Companion Volume*. Council of Europe Publishing, Strasbourg.
- DeepSeek AI. 2024. *DeepSeek-V3: Model Overview and API*. Software.
- Holger Diessel. 2019. *Usage-based construction grammar*, pages 50–80. De Gruyter Mouton, Berlin, Boston.
- Simon C. Dik. 1981. *Functional Grammar*, volume 7 of *Publications in Language Sciences*. De Gruyter, Berlin, Boston.
- 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. 2017. *Salience in usage-based SLA*. In Susan M. Gass, Patti Spinner, and Jennifer Behney, editors, *Salience in Second Language Acquisition*, page 20. Routledge, New York.
- Rod Ellis. 2024. *Task-based and task-supported language teaching*. *International Journal of TESOL Studies*, 6(4).
- Faidra Faitaki and Victoria A. Murphy. 2019. *Oral language elicitation tasks in applied linguistics research*. In Jim McKinley and Heath Rose, editors, *The Routledge Handbook of Research Methods in Applied Linguistics*, page 10. Routledge, London.
- Nicholas X. Fincham and Alejandro Arronte Alvarez. 2024. *Using large language models (LLMs) to facilitate L2 proficiency development through personalized feedback and scaffolding: An empirical study*. In *Proceedings of the International CALL Research Conference, 2024*, pages 59–64.

- Pablo Gervás, Carlos León, Mayuresh Kumar, Gonzalo Méndez, and Susana Bautista. 2025. Prompting an LLM chatbot to role play conversational situations for language practice. In *International Conference on Computer Supported Education, CSEdu-Proceedings*, volume 2, pages 257–264.
- Dominik Glandorf, Peng Cui, Detmar Meurers, and Mrinmaya Sachan. 2025. [Grammar control in dialogue response generation for language learning chatbots](#). *arXiv preprint arXiv:2502.07544*. Accepted to NAACL 2025.
- D. Larsen-Freeman. 2003. *Teaching Language: From Grammar to Grammaring*. Newbury House teacher development. Thomson/Heinle.
- Patsy Martin Lightbown. 2007. *Transfer Appropriate Processing as a Model for Classroom Second Language Acquisition*, pages 27–44. Multilingual Matters, Bristol, Blue Ridge Summit.
- Lester Loschky and Roman Bley-Vroman. 1993. Grammar and task-based methodology. In Graham Crookes and Susan M. Gass, editors, *Tasks in Language Learning*, pages 123–167. Multilingual Matters, Clevedon.
- Roy Lyster and Masatoshi Sato. 2013. [Skill acquisition theory and the role of practice in l2 development](#). In María del Pilar García Mayo, María Juncal Gutiérrez Mangado, and María Martínez-Adrián, editors, *Contemporary Approaches to Second Language Acquisition*, volume 9 of *AILA Applied Linguistics Series*, pages 71–92. John Benjamins Publishing Company, Amsterdam.
- Gonzalo Méndez and Susana Bautista. 2025. Configuring an LLM chatbot as practice partner for language learning. In *Advances in Artificial Intelligence-IBERAMIA 2024: 18th Ibero-American Conference on AI, Montevideo, Uruguay, November 13–15, 2024, Proceedings*, volume 15277, page 458. Springer Nature.
- Meta. 2024. [LLaMA 3, Model Card and Documentation](#). Software.
- Mehdi Mirzaei, Masoud Zoghi, and Haniyeh Davatgari Asl. 2017. [Understanding the language learning plateau: A grounded-theory study](#). *Teaching English Language*, 11(2):195–222.
- Mistral AI. 2024. [Mistral 7B and Mistral Small API Documentation](#). Software.
- David Nunan. 2004. *Task-based language teaching*. Cambridge university press.
- Yuki Okano, Kotaro Funakoshi, Ryo Nagata, and Manabu Okumura. 2023. [Generating dialog responses with specified grammatical items for second language learning](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 184–194.
- OpenAI. 2024. [GPT-4o API documentation](#). Software.
- Lourdes Ortega and Robert DeKeyser. 2007. *Meaningful L2 practice in foreign language classrooms: A cognitive-interactionist SLA perspective*, page 180–207. Cambridge Applied Linguistics. 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.
- Jack Croft Richards. 2008. *Moving beyond the plateau: From Intermediate to Advanced Levels in Language Learning*.
- 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*, page 25. Routledge, New York.
- 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.
- Tetyana Sydorenko, Phoebe Daurio, and Steven L. Thorne. 2018. [Refining pragmatically-appropriate oral communication via computer-simulated conversations](#). *Computer Assisted Language Learning*, 31(1-2):157–180.
- Gladys Tyen, Andrew Caines, and Paula Buttery. [LLM chatbots as a language practice tool: A user study](#). In *Swedish Language Technology Conference and NLP4CALL*, pages 235–247.
- Sabrina Wilske. 2015. *Form and Meaning in Dialog-Based Computer-Assisted Language Learning*. Dissertation, Universität des Saarlandes, Saarbrücken. Co-supervised by Prof. Dr. Manfred Pinkal and Prof. Dr. Detmar Meurers.
- Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. [A survey on recent advances in LLM-based multi-turn dialogue systems](#). *arXiv preprint arXiv:2402.18013*. 35 pages, 10 figures, submitted to ACM Computing Surveys.
- Houquan Zhou, Yang Hou, Zhenghua Li, Xuebin Wang, Zhefeng Wang, Xinyu Duan, and Min Zhang. 2023. How well do large language models understand syntax? an evaluation by asking natural language questions. *arXiv preprint arXiv:2311.08287*.

A Template-based Prompt

"I am a [level] [language] student. I want to practice [EGP structure information] by [task_name]. I want to ensure that I [can_do_statement]. For example, [examples]. Let’s perform a [task_type]

exercise. You are a [system_role]. [system_instructions]. Please keep your messages short and use easy words. Output only your next turn."

B Baseline Condition Prompt

"You are a friendly English tutor. I want to practice by [task name]. Please use direct, short and simple sentences and easy words. Output only your next turn."

C Simplification Rephrase Prompt

"The learning objective is [task name] (e.g., [examples]). Simplify ONLY the advanced grammar constructs while carefully preserving the learning objective in the following text. Advanced grammar to simplify: [advanced structure] IN [sentence] Text to rephrase: [advanced response] Please provide only the rephrased text without explanations."

D Target Structure Inclusion Rephrase Prompt

"The student is practicing [task name] (e.g., [examples]) through dialogue. Please rephrase the following response to contain that grammar structure while keeping the meaning: [sentence]"

E Response Appropriateness Rubric

1. Completely inappropriate: Off-topic and incoherent throughout
2. Mostly inappropriate: Some relevance, but major issues in logic or coherence
3. Somewhat appropriate: Largely on-topic, but difficult to follow due to coherence issues
4. Appropriate: On-topic and generally clear, with minor coherence issues
5. Fully appropriate: Clear, coherent, and consistently on-topic

F Dialogue Snippets

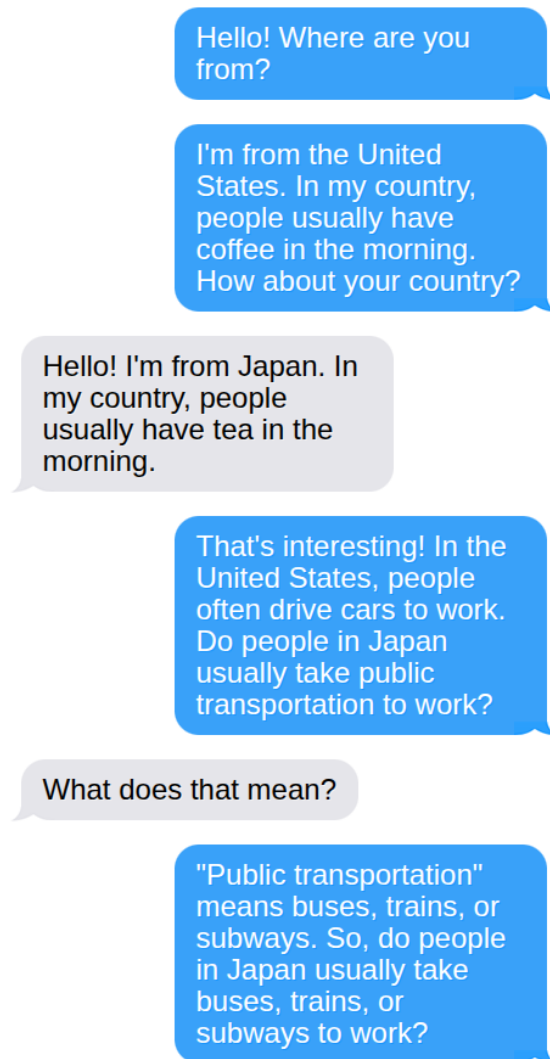


Figure 1: Example output from GPT-4o acting as the tutor under the template-based prompt condition. The Q&A task shown is "Talking about common habits in your country." This snippet is from the second run, which includes an injected clarification request. Blue bubbles indicate tutor messages.

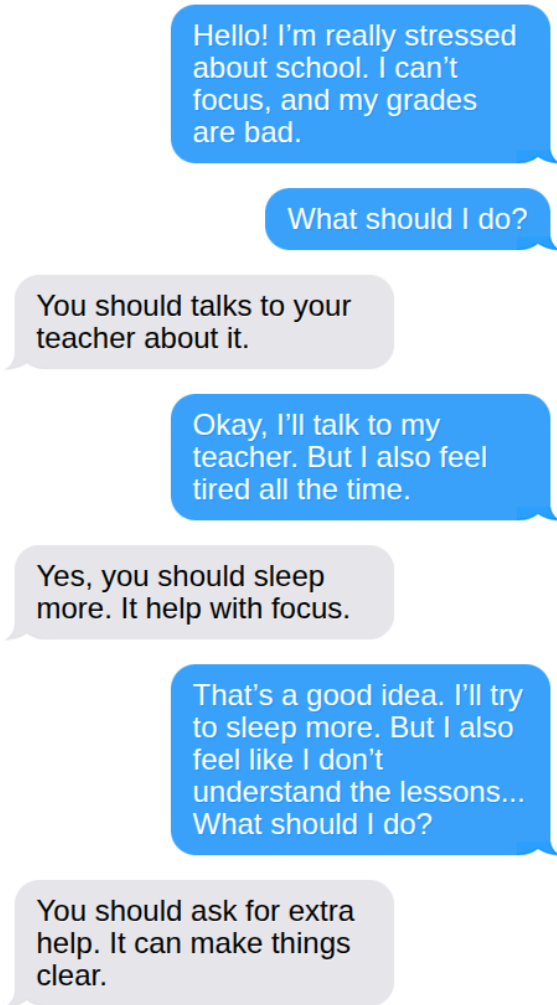


Figure 2: Example output from Deepseek-V3 acting as the tutor, under the template-based prompt condition. The role-play task shown is "Giving a friend advice about school". This is a snippet from the first task run, in which grammar errors are instructed. Blue bubbles indicate tutor messages.

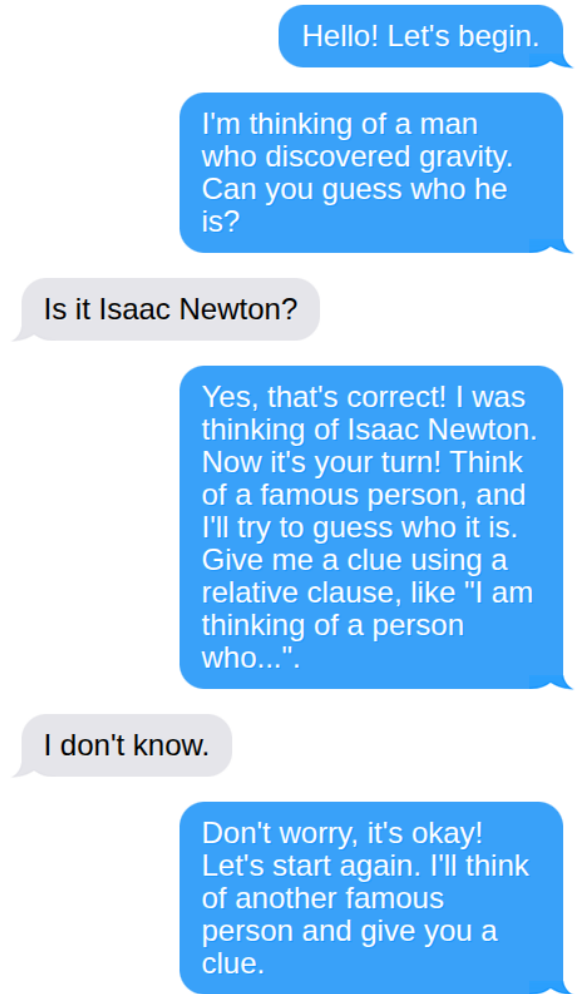


Figure 3: Example output from LLaMa 3.3-70B-Instruct acting as the tutor, under the template-based prompt condition. The information-gap task is "Playing a guessing game about famous scientists". This is a snippet from the third task run, in which a misunderstanding indication is injected. Blue bubbles indicate tutor messages.

Can LLMs Reliably Simulate Real Students' Abilities in Mathematics and Reading Comprehension?

KV Aditya Srivatsa

Kaushal Kumar Maurya

Ekaterina Kochmar

Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

{vaibhav.kuchibhotla, kaushal.maurya, ekaterina.kochmar}@mbzuai.ac.ae

Abstract

Large Language Models (LLMs) are increasingly used as *proxy students* in the development of Intelligent Tutoring Systems (ITSs) and in piloting test questions. However, *to what extent these proxy students accurately emulate the behavior and characteristics of real students remains an open question*. To investigate this, we collected a dataset of 489 items from the National Assessment of Educational Progress (NAEP), covering mathematics and reading comprehension in grades 4, 8, and 12. We then apply an *Item Response Theory (IRT)* model to position 11 diverse and state-of-the-art LLMs on the same ability scale as real student populations. Our findings reveal that, without guidance, strong general-purpose models consistently outperform the average student at every grade, while weaker or domain-mismatched models may align incidentally. Using grade-enforcement prompts changes models' performance, but whether they align with the average grade-level student remains highly model- and prompt-specific: no evaluated model-prompt pair fits the bill across subjects and grades, underscoring the need for new training and evaluation strategies. We conclude by providing guidelines for the selection of viable proxies based on our findings.¹

1 Introduction

Large language models (LLMs) are capable of generating fluent and coherent text and excelling at many complex tasks (Chang et al., 2024; Zhao et al., 2024). Their rise offers new opportunities for educational technology, notably in (i) intelligent tutoring systems (ITS; Wang et al., 2024) and (ii) *piloting assessments* before they go live (Liu et al., 2025; Grohs et al., 2024). ITS provides targeted feedback and adaptive instruction, while reliable

¹All related code and data is available at <https://github.com/kvadityasrivatsa/IRT-for-LLMs-as-Students>

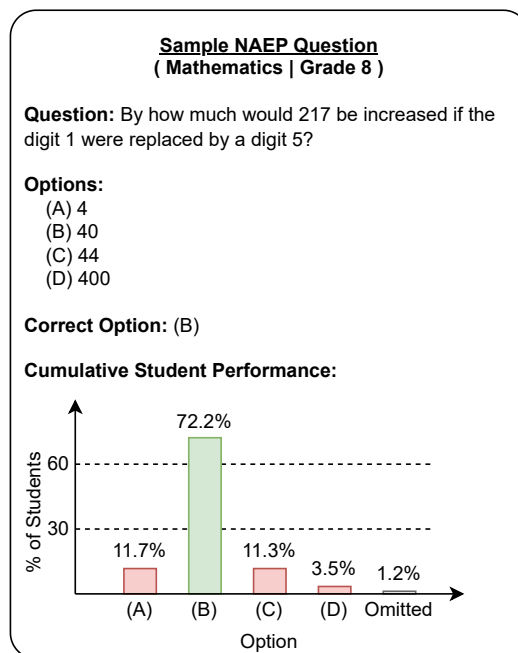


Figure 1: Sample NAEP question from grade 8 mathematics.

assessments track learning without bias. Yet both require understanding how real students would interact with them, which is extremely challenging to verify.

Ideally, tutors and test forms should be vetted on representative student samples across skill levels. This, however, is resource-intensive, especially in regions already short on teachers and infrastructure (UNESCO, 2023; Woolf et al., 2013). Teacher-led evaluations (e.g., Macina et al., 2023) and static logs similarly fail to scale or capture the dynamics of new items and adaptive strategies (Belz et al., 2023; U.S. Department of Education, 2023). These constraints motivate alternatives that enable rigorous, equitable evaluation at scale.

An emerging approach is for *simulate* students with LLMs (Mollick et al., 2024; Sonkar et al., 2024). Proxy models can be conditioned on at-

tributes such as grade level, offering fast, repeatable tests of tutor features or item quality. However, current evaluations are based on an expert judgment of plausibility (Macina et al., 2023), leaving open the question of how closely such proxies match real student performance. Similarly, in psychometrics, LLMs have been used as *synthetic examinees*: e.g., Liu et al. (2025) show that GPT-3.5/4 answer sets yield item statistics that mirror a 50-student pilot, and Grohs et al. (2024) demonstrate that ChatGPT can pre-flag weak or biased items. However, these studies treat LLMs as single test-takers and do not look into whether persona prompts can tie their abilities to specific grade bands.

Our approach: We apply IRT (Baker, 2001) to measure how 11 diverse LLMs and real students perform on the same grade-level questions. Using data from the National Assessment of Educational Progress (NAEP) (National Center for Education Statistics, 2022) for mathematics and reading in grades 4, 8, 12, we evaluate whether the LLM responses (both *generic* and *grade-conditioned*) align with authentic student response patterns. Specifically, we address the following research questions: **RQ1** – Under standard prompting, how do LLMs compare with real students across grades and subjects? **RQ2** – When asked to act as an average student in a given grade: How does LLM performance change? (**RQ2.1**) Does the shift match real grade-level patterns? (**RQ2.2**)

The main contributions of our work are as follows:

- We compile and release a dataset² sourced from NAEP of real student responses to subject-specific, grade-targeted questions, covering two subjects (mathematics and reading assessment) and three grade levels (4, 8 and 12).
- We adapt Item Response Theory (IRT) to assess the alignment between LLM-generated responses and actual student performance patterns.
- We conduct an evaluation of 11 diverse LLMs, examining how well they approximate student responses under both *unenforced* (generic) and *grade-enforced* prompts.

²<https://github.com/kvadiyasrivatsa/IRT-for-LLMs-as-Students>

2 Related Work

Simulated Students in Intelligent Tutoring Systems Early simulated-student work relied on production rule *apprentices* that learn from worked examples and then reproduce step-level behavior inside an ITS. The *SimStudent / Apprentice-Learner* family shows that such models can generate realistic error types and serve as policy learners to hint (Matsuda et al., 2023; Smith et al., 2024). More recent studies graft LLMs onto this pipeline: it has been shown that GPT-4 “think-aloud” traces improve bug-library discovery and fine-grained skill tagging, while LLM agents at dialogue level can populate entire synthetic classroom cohorts (Mollick et al., 2024). These approaches demonstrate that generative text can complement symbolic learner models, yet they rarely test whether the *ability* distribution of synthetic learners matches that of real students – a gap we address through our analysis.

LLM-Generated Responses for Item Calibration Psychometric studies have begun to treat LLM outputs as *synthetic examinee responses*. Liu et al. (2025) show that the GPT-3.5/4 answer sets yield 3PL item statistics that match a 50-student baseline, reducing the pretest costs. Grohs et al. (2024) use ChatGPT to filter out low information or biased items. He-Yueya et al. (2024) further adapt IRT to align LLM and human response patterns, while Zelikman et al. (2023) simulate K-12 students. However, these works produce only aggregate correlations; they do not examine whether an LLM’s *latent ability* aligns with a particular grade band or whether persona prompts shift that ability in predictable ways. Our work closes this gap with an IRT model that maps LLM performance to grade-level performance.

Persona-Conditioned Prompting and Alignment Prompting a model with an explicit role (e.g., “*You are a 4th-grade student*”) can change both reasoning depth and surface style. Benedetto et al. (2024) find that a one-sentence student-level prompt lets GPT-4 imitate weak, average, and strong test-takers across subjects, although adherence to the target level is uneven. Broader evaluations such as CharacterEval (Tu et al., 2024) measure persona consistency in dialogues, while Kim et al. (2024) show that role prompts can either help or hurt accuracy depending on task characteristics. None of these efforts connect persona adherence to *quan-*

titative grade-level ability estimates, nor do they compare default and persona-conditioned ability curves within a unified IRT framework.

Together, these strands indicate that (i) LLMs are already employed as simulated students and psychometric stand-ins, and (ii) persona prompts shift model behaviour without a principled link to grade-level metrics. Our study unifies the two directions by applying an IRT model to quantify how default and persona-conditioned LLM outputs align with average student performance at grades 4, 8, and 12.

3 NAEP Data

3.1 Source and Composition

We prepared our dataset using publicly available items and student response data from the National Assessment of Educational Progress (NAEP) (National Center for Education Statistics, 2022),³ a large-scale assessment program administered by the National Center for Education Statistics (NCES). NAEP periodically assesses student achievement across the United States in key subject areas, including mathematics and reading. These assessments are conducted in grades 4, 8, and 12, offering a cross-sectional perspective on student proficiency throughout K–12 education.

3.2 Coverage and Educational Context

We source questions from both the *mathematics* and *reading comprehension* assessments at the three grade levels, capturing a broad spectrum of student performance and cognitive development throughout different educational stages. We focus on these two subjects for two reasons: (1) numeracy and literacy are considered fundamental skills (e.g., Williams, 2003); and (2) NAEP data cover three grades for these subjects, while many other subjects only cover one or two grades. Math questions span topics such as measurement, algebra, geometry, and probability and statistics, with overall difficulty scaling with grade level. Reading comprehension items are based on passages whose average length increases with grade. The corresponding questions shift from direct factual queries in lower grades to those requiring interpretation and reflection at higher levels.

Each record contains the original question, multiple choice options, the correct annotated answer,

and anonymized aggregate response patterns. For each item, the dataset reports the percentage of students who selected each option or omitted the question. Figures 1 and 4 show representative examples from the grade-8 mathematics and grade-12 reading subsets, respectively.

Since NAEP is a continually administered assessment, this dataset can be periodically updated with newly released items. This makes it a dynamic resource that can evolve along with changes in educational standards and student performance distributions, offering long-term utility for evaluating automated student proxies and similar tasks.

3.3 Preprocessing and Filtering Criteria

NAEP assessments encompass a variety of question types, modalities, and response formats. Given that this is a preliminary effort to develop a quantitative and interpretable framework for aligning LLM performance with real student behavior, the inclusion of diverse modalities can introduce confounding factors that obscure analysis. For example, items that involve images, diagrams, or tables introduce the additional variable of visual comprehension, making it difficult to isolate language understanding as the primary factor in model performance. Similarly, free-form responses present evaluation challenges: gold-standard answers are often limited in number and may not capture the full range of acceptable responses. Assessing these reliably often requires expert judgment, which undermines the feasibility of scalable, LLM-based evaluation.

In contrast, multiple choice questions offer clearly defined answer sets, enabling a more straightforward and objective evaluation, which is crucial for both quantitative benchmarking and interpretability via Item Response Theory (IRT). Consequently, we apply two main filtering criteria when constructing our dataset.

- *Text-only content:* We exclude any items that involve diagrams, tables, or multimedia elements, retaining only questions and instructions presented in text.
- *Multiple-choice format:* We include only multiple choice questions (MCQs), which support standardized evaluation and facilitate downstream processing, such as answer extraction and IRT-based analysis.

After filtering, our final dataset consists of 489 multiple-choice items in English: 249 from mathe-

³<https://nces.ed.gov/nationsreportcard/>

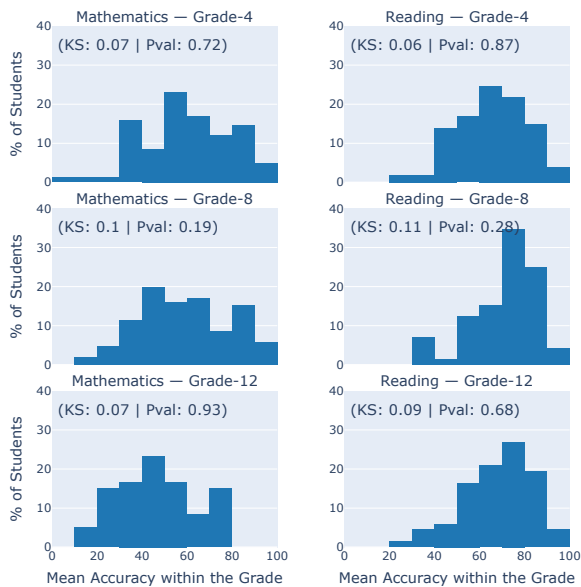


Figure 2: Distribution of question-level accuracy in NAEP assessments across grades and subjects. KS statistics and corresponding p-values are reported to assess normality; distributions with $p > 0.05$ are considered consistent with a normal distribution.

mathematics and 240 from reading. Table 3 summarizes key statistics from the dataset. Figure 2 shows the distribution of questions answered by students with varying accuracy for each subject and grade. We calculate the Kolmogorov-Smirnov (KS) statistic for each distribution to test for normality, and all subsets sufficiently ($p\text{-value} > 0.05$) follow a normal distribution.

4 Proposed Methodology

Our goal is to evaluate LLMs alongside human students recorded in the NAEP dataset by estimating their answering ability on a shared scale. To do this, we draw on Item Response Theory (IRT; Baker (2001)), a well-established framework in educational measurement. IRT enables us to jointly model the ability of test takers and the difficulty of individual test items using probabilistic principles.

4.1 Estimating Student (LLM) Ability

We begin with the Rasch model (Rasch, 1960), a widely used and interpretable form of IRT. It assumes that the probability of a correct response depends only on the difference between a participant’s ability and an item’s difficulty. This model uses a single parameter per item, that is, difficulty b_i , and one ability parameter θ_i per participant.

The Rasch model defines the probability that participant i correctly answers item j as:

$$P(X_{ij} = 1) = \frac{e^{\theta_i - b_j}}{1 + e^{\theta_i - b_j}}, \quad (1)$$

where $\theta_i \in \mathbb{R}$ is the ability of the participant i , $b_j \in \mathbb{R}$ is the difficulty of item j , and \mathbb{R} is the common scale of difficulty or ability.

To estimate b_j , we use the empirical proportion p_j of correct responses for each item in the population. A simple approximation is:

$$b_j \approx \log\left(\frac{1 - p_j}{p_j}\right), \quad (2)$$

which reflects that the harder items (with lower p_j) have higher difficulty values (Bond and Fox, 2015). Once the item difficulties are known, the ability of each participant θ_i can be estimated using marginal maximum likelihood or Bayesian inference based on their response pattern.

4.2 Grade-Alignment Prompting

Our first research question (RQ1) investigates how an LLM’s problem-solving ability compares to that of average students at different grade levels, specifically grades 4, 8 and 12, based on the NAEP dataset. To measure this, we begin with a minimal zero-shot prompt, which we refer to as **UNENFORCED** (see Appendix Figures 5 and 9 for exact prompt templates). This prompt simply presents the question to the model without any added instructions or persona guidance.

Our second research question (RQ2) explores whether LLMs can align their responses with the answering patterns and performance levels of students in specific grades. To probe this, we design a set of increasingly guided zero-shot prompts that aim to steer the model toward grade-level reasoning.

1. **GRADEENFORCEDMINIMAL**: Identical to the Unenforced prompt, but with the added instruction that the model should act as an average student from a specific grade (4, 8, or 12). The exact prompts are presented in Appendix Figures 6 and 10.
2. **GRADEENFORCEDBASICCOT**: Builds on the minimal version by prompting the model to consider what an average student at the specified grade would likely choose and why. This prompt encourages brief, grade-aware reasoning and reflects the student’s typical reasoning ability and common error patterns.

See Figures 7 and 11 in the Appendix for the exact prompts.

3. **GRADEENFORCEDFULLCOT**: Adds further scaffolding by dividing the reasoning process into two steps. First, the model is instructed to reflect on whether an average student at the given grade level would be likely to answer the question correctly. Second, based on that reflection, the model either justifies a correct answer or, if the student is unlikely to succeed, selects and explains the most plausible incorrect answer. See Figures 8 and 12 in the Appendix for the exact prompts.

The design of the GRADEENFORCEDBASIC-COT and GRADEENFORCEDFULLCOT prompts is inspired by Benedetto et al. (2024), who developed similar prompts to simulate student reasoning across skill levels on exam-style questions of varying difficulty. Their work informed our decision to incorporate reasoning about the ability of a student and the likelihood of error into our prompt design.

Our aim is not to claim these are optimal prompts or to exhaustively search for the best possible formulations. Instead, we adopt straightforward, representative prompting strategies aligned with popular practices to focus our investigation on whether such methods meaningfully promote grade-level alignment in model behavior. This may limit the scope of our findings, but it allows us to isolate and evaluate the effects of targeted prompting on grade-sensitive reasoning.

5 Experimental Setup

5.1 Task Setup

We design our experiments based on the framework described in Section 4. The evaluation is conducted in two phases:

1. **Problem-Solving**: LLMs answer questions in a standard problem solving setting, without specific instructions on how to mimic human behavior. Their performance is compared to that of average students in different grade levels.
2. **Grade-Level Mimicking**: LLMs are explicitly instructed to emulate an average student of a specific grade level and respond as such.

In both phases, we apply a Rasch model to assess performance. Each question is treated as

an individual item j , and each LLM is treated as an in-distribution test-taker i . The binary response of LLM i to the question j is represented as $s_{ij} \in \{0, 1\}$, where $s_{ij} = 1$ indicates a correct answer.

LLM	Open Source?	Parameter Count	Fine-Tuned?	Benchmark Scores	
				GSM8K (%)	MLU (%)
LlAMA2-13B (Touvron et al., 2023)	✓	13B	✗	28.7	54.8
LlAMA2-70B (Touvron et al., 2023)	✓	70B	✗	56.8	68.9
LlAMA3.1-8B (Touvron et al., 2023)	✓	8B	✗	84.5	73.0
LlAMA3.1-70B	✓	70B	✗	95.1	86.0
Mistral-7B (Jiang et al., 2023)	✓	7B	✗	52.1	60.1
Qwen2.5-7B (Yang et al., 2024)	✓	7B	✗	85.4	74.2
Qwen2.5-Math (Yang et al., 2024)	✓	7B	✓	91.6	67.8
GPT-3.5-Turbo (OpenAI, 2023)	✗	-	✗	57.1	70.0
o3-Mini (OpenAI, 2025)	✗	-	✗	89.9	85.2
SocraticLM (Liu et al., 2024)	✓	7B	✓	60.6	-
LearnLM-1.5-Pro (Modi and the LearnLM Team, 2024)	✗	-	✓	-	-

Table 1: List of LLMs evaluated in our study, along with key descriptors about each model, i.e., open source availability, parameter size, whether the model is fine-tuned (as opposed to pretrained or instruction-tuned), and scores on reasoning and comprehension benchmarks GSM8K and MMLU (we omit scores that have not been released publicly by the respective model’s paper or technical report).

5.2 Models

We select a diverse set of 11 LLMs (see Table 1) to ensure broad coverage across access types (open vs. closed), model sizes and training paradigms (pretrained vs. domain-finetuned). Our goal is to capture a range of capabilities relevant to reasoning and comprehension, as reflected in benchmarks like GSM8K (Cobbe et al., 2021) and MMLU (Hendrycks et al., 2020). We include both general-purpose models and those finetuned to specific domains. GPT-3.5-Turbo is included based on Benedetto et al. (2024), who suggest that it can adapt responses to the levels of student ability instructed. SocraticLM and LearnLM-1.5-Pro are fine-tuned on pedagogical data; therefore, they might have more accurate insights into the performance of students at different grade levels.

5.3 Evaluation

Measuring Problem-Solving Correctness All problems in our dataset are multiple choice questions (MCQs), which simplifies the evaluation of correctness: *a model’s response is considered correct if the selected option matches the correct answer provided with the dataset*. This binary distinction between correct and incorrect responses makes the data well-suited for dichotomous (i.e.,

the answer can only be correct or incorrect) Item Response Theory (IRT) models. We found that model responses vary in structure and require a unified follow-up prompt to extract the predicted choice from each model response (see Figure 13 in the Appendix).

Estimating Grade-Level Alignment To address both research questions, we estimate how closely a model’s performance aligns with that of an average student at a given grade level. To pin the origin and unit of the Rasch ability scale during marginal maximum-likelihood estimation, we follow the standard convention of treating examinee abilities as standard-normal, $\theta \sim \mathcal{N}(0, 1)$. Although not theoretically necessary, Embretson and Reise (2000) note that this assumption is a reasonable way to identify the latent trait because it fixes the zero point and variance without constraining the shape of the data.

Therefore, the average student has an ability parameter of zero ($\theta_{\text{avg}} = 0$).

The estimated ability parameter θ_i for a model i can be interpreted in relation to this benchmark. *The closer θ_i is to zero, the more the model’s performance aligns with that of the average student.* We can also express this alignment using the percentile rank, computed via the cumulative distribution function (CDF) of the standard normal distribution, denoted by $\Phi(\theta)$:

$$\text{Percentile Rank} = \Phi(\theta) \times 100 \quad (3)$$

A percentile rank of 50 corresponds to the average student. Higher percentile ranks indicate higher levels of ability relative to the population. We use percentile rank as our main metric to measure LLM or student ability, as it has a fixed range (0-100 and is centered at 50), which allows for easier comparison of LLM alignment across subjects and grade levels.

6 Results & Discussion

In Table 2, we report each LLM’s percentile rank in grade-level mathematics and reading comprehension questions under three conditions: unenforced prompting (P_U), grade-enforced prompting (P_E) and their difference (Δ). We report P_U for the best prompt for each LLM per subject (see Table 5 in the Appendix for the best prompts for each setting) which maximizes grade alignment, i.e., when percentile rank is closest to 50 (average).

Avg. Deviation records the mean absolute deviation from $P = 50$ for the corresponding prompt settings. The baseline Random Choice reports the percentile scores achieved with a randomized option chosen for each problem. This setup allows us to:

- address **RQ1** by comparing model percentiles to the student mean (50th percentile) across grades and subjects, and
- address **RQ2** by (i) quantifying the effect of grade enforcement on LLM performance (**RQ2.1**) and (ii) evaluating whether these shifts mirror human student response patterns (**RQ2.2**).

For further context, Table 4 presents the accuracy of each LLM under the unenforced condition.

6.1 RQ1: Alignment under Unenforced Prompting

We ask whether the unenforced problem solving prompt generates outputs that align with that of the average student in each grade (P_U).

Mathematics. Most models, especially those scoring well on GSM8K, e.g., LLaMA3.1-70B, Qwen2.5-Math, o3-Mini, and SocraticLM, achieve high percentiles in every grade, overshooting all benchmarks and showing no alignment with any specific grade. This is also reflected in the high average deviation of 40.5, 35.0, and 32.9 percentile points, respectively, from the optimal $P=50$ mark. In contrast, smaller models with relatively poorer benchmark performance, such as LLaMA2-13B and Mistral-7B, exhibit lower percentiles and show better alignment between grades.

Reading. Similar to mathematics, the models demonstrate high average percentile scores in reading for grades 4 and 8, proving unsuitable for faithful student mimicking. The models in our pool align better with grade 12, with relatively lower average P_U values. Fine-tuned models (not tuned for grade-alignment), e.g., Qwen2.5-Math, SocraticLM – Qwen2.5-Math further tuned on pedagogical data, have a poorer overall performance, resulting in better alignment across grades.

Across grades and subjects, all models score well above the Random Choice baseline. Without enforced instructions, LLMs rarely self-calibrate to grade difficulty. They overshoot when capacity is high and align only when under-powered or off-domain.

LLM	Mathematics									Reading								
	Question Grade 4			Question Grade 8			Question Grade 12			Question Grade 4			Question Grade 8			Question Grade 12		
	P_U	P_E	Δ	P_U	P_E	Δ	P_U	P_E	Δ	P_U	P_E	Δ	P_U	P_E	Δ	P_U	P_E	Δ
LLaMA2-13B	63.7	66.1	+2.4	52.6	50.8	-1.8	48.6	66.5	+17.9	99.7	95.5	-4.2	94.9	87.6	-7.3	80.9	58.7	-22.3
LLaMA2-70B	85.5	33.5	-52.0	23.9	31.6	+7.6	42.4	57.8	+15.4	88.6	79.8	-8.8	72.3	69.1	-3.2	71.6	58.7	-12.9
LLaMA3.1-8B	96.8	85.5	-11.3	85.2	60.0	-25.2	79.5	69.3	-10.2	99.7	77.9	-21.8	96.7	92.7	-4.0	86.7	83.9	-2.8
LLaMA3.1-70B	99.6	97.6	-2.0	98.9	98.0	-0.9	96.1	97.0	+0.9	99.9	99.9	0.0	96.7	92.7	-4.0	83.9	80.9	-3.0
Mistral-7B	63.7	58.9	-4.8	43.5	49.0	+5.4	63.7	57.8	-5.9	94.3	67.9	-26.4	84.8	69.1	-15.7	83.9	55.5	-28.4
Qwen2.5-7B	99.6	18.5	-81.2	99.3	22.5	-76.7	96.1	30.3	-65.8	98.2	5.2	-93.1	98.2	7.8	-90.4	77.9	30.2	-47.7
Qwen2.5-Math	99.8	70.7	-29.1	98.5	96.1	-2.4	97.8	97.8	0.0	72.0	69.9	-2.0	36.5	29.4	-7.1	43.4	43.4	0.0
GPT-3.5_Turbo	89.0	44.7	-44.3	70.8	11.7	-59.1	79.5	45.5	-34.0	99.7	61.8	-37.8	98.2	65.9	-32.3	68.3	43.4	-24.9
o3-Mini	98.9	98.3	-0.6	98.5	99.5	+1.0	95.1	99.3	+4.2	99.3	99.9	+0.6	99.1	98.2	-0.9	86.7	86.8	+0.1
SocraticLM	99.3	96.8	-2.6	99.7	99.3	-0.5	97.0	98.4	+1.4	59.8	63.9	+4.1	34.0	39.1	+5.0	32.6	49.3	+16.7
LearnLM-1.5-Pro	99.8	75.3	-24.6	99.7	93.6	-6.2	98.9	98.4	-0.5	99.9	24.5	-75.4	94.9	53.3	-41.6	65.1	58.7	-6.4
Avg. Deviation	40.5	27.5	-13.0	35.0	30.2	-4.8	32.9	28.8	-4.2	41.9	30.6	-11.3	37.8	27.5	-10.3	25.4	15.2	-10.2
Random Choice	6.1			1.4			6.7			4.12			4			0.9		

Table 2: LLM percentile scores on grade-level questions from mathematics and reading without grade enforcement (P_U – shaded blue), with grade enforcement (P_E – shaded green), and their difference ($\Delta = P_U - P_E$ – shaded yellow). Darker hues for P_U and P_E denote closer alignment to the average score of 50 and larger absolute change in Δ . **Boldface** highlights the best model (i.e., closest to 50) in each setting. Avg. Deviation records the mean absolute deviation from $P=50$ for corresponding prompt settings. The Random Choice baseline reports the percentile scores attained with a randomized option chosen for each problem.

6.2 RQ2.1: Effect of Grade-Level Prompts

We test whether prompting a model to “think like an average grade g student” changes its performance (Δ in Table 2), regardless of the resultant alignment.

Drops: Overall, we note that upon prompting models to mimic the average student of grades 4, 8, or 12, percentile scores generally drop (see Avg. Deviation Δ in Table 2), with a greater drop for a lower target grade. Qwen2.5-7B records the highest drop of 93.1 percentile points for reading grade 4 as well as the greatest average drop.

Gains: We observe that grade-specific prompting can also increase model performance. For example, several settings with LLaMA2-13B/70B for mathematics and all grade settings with SocraticLM for reading result in higher P_E than P_U .

Stable: Some models, such as LLaMA3.1-70B and o3-Mini for subjects and SocraticLM for mathematics, show little to no change between their values of P_U and P_E values, despite their respective P_U values having a high deviation from the target $P=50$.

Prompt Strength: Among the three grade enforcement prompts, the most detailed GRADEENFORCEDFULLCOT prompt (with explicit instruc-

tion to consider the probability that an average student of the target grade will get the given problem right) causes the largest changes (Figure 3a). This shows that grade-level cues can markedly increase or lower scores depending on the model and prompt strength, although a few models remain robust.

6.3 RQ2.2: Alignment Under Enforced Prompts

We investigate whether grade-specific prompts move the model performance closer to the average student (ideal $P = 50$). We find that the results are spread across the following categories:

(1) Aligned P_U and aligned P_E : Some models that are close to the 50th percentile without grade-specific prompting maintain good alignment after prompting (for example, LLaMA2-13B / 70B for mathematics and SocraticLM for reading). These models can act as “proxy students” out of the box for particular pairs of subjects’ grades.

(2) Misaligned P_U and misaligned P_E : Other models’ percentile scores can range far above or below the median despite grade-specific prompting (for example, P_U s and P_E s for o3-Mini across subjects and grades stay far above 50). We did not observe any model that consistently scored below the median percentile.

(3) Misaligned P_U and aligned P_E : In some cases, prompting can help induce grade alignment (P_E) when unenforced alignment is poor (P_U). For example, Mistral-7B’s percentile range on reading problems moves from $P_U \in [83.9 - 94.3]$ to $P_E = [55.5 - 69.1]$; GPT-3.5-Turbo shows similar gains in most tasks. Such cases demonstrate the desired effect of grade-specific prompting.

(4) Aligned P_U and misaligned P_E : In contrast, grade-specific prompting can cause models to overshoot. Qwen2.5-7B in grade 4 reading drops from 98.2 to 5.2 ($\Delta = -93.1$), overshooting the target.

Prompt design matters. Figure 3b shows that the GRADEENFORCEDFULLCOT template changes scores the most. However, it is not always the most optimal prompt setting to achieve better grade alignment (lower percentile deviation from 50).

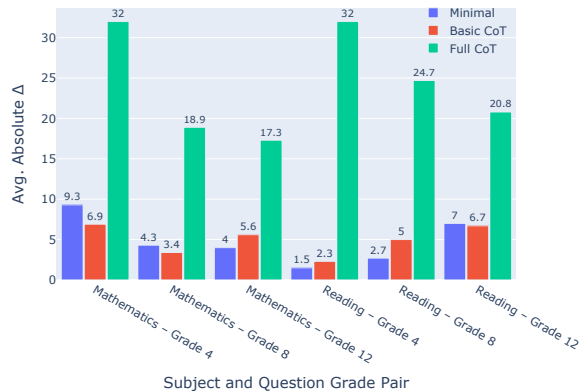
Fine-tuned models. Pedagogically tuned models (LearnLM-1.5-Pro, and SocraticLM) are not better aligned than general LLMs (such as Mistral-7B), with or without prompts, indicating that faithful grade-level emulation probably needs explicit alignment objectives.

Thus, grade alignment is model-prompt specific; no single prompt works everywhere. Reliable grade-level emulation will require tailored prompting that does not ensure generalization to other grades or subjects.

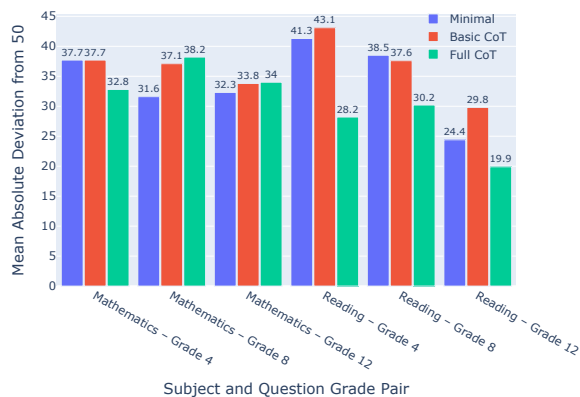
6.4 Guidelines for Selecting Viable LLM “Proxy Students”

Our experiments confirm that no single model-prompt pair reliably matches average student performance in every grade and subject. Before an LLM can stand in for real students, e.g., to trial new test items or train a model for an ITS, it should pass the following baseline checks:

- Grade alignment.** The model’s ability estimate (θ_n) in a representative item set must fall within the normative band of the grade: core average ± 1 logit (percentile: 15.9 to 84.1), extended ± 1.5 , outlier $\geq \pm 2$ (Bond and Fox, 2015). Models such as GPT-3.5-Turbo stayed in the core range with appropriate prompts for most grades.
- Developmental ordering.** Ability should rise monotonically with grade, mirroring trends in NAEP reading (217, 262, and 285 according



(a) On Δ



(b) On grade-alignment. We report mean absolute deviation from an average grade-level student’s percentile score (i.e., 50). Thus, greater the deviation, poorer the average alignment.

Figure 3: Impact of grade-enforcing prompting

to NAEP’s own cumulative scoring scale for grades 4, 8, 12 respectively; National Center for Education Statistics, 2022). Several pairs violated this; e.g., Mistral-7B’s P_U was 38.3, 17.0, 40.5 for the same grades.

- Prompt stability.** Grade-enforcing prompts can improve or harm performance. An unenforced prompt should be used if the model is already aligned; otherwise, one should verify that enforcement is equally accurate across all grades.

These criteria are necessary, but not sufficient. We believe that more accurate guidelines for faithful student mimicking will emerge with richer evaluation datasets.

7 Conclusion

In this paper, we investigate whether LLMs’ regular problem-solving performance aligns with that of an average student of a given grade, and whether

explicit prompting to act like the average student makes a difference and improves this alignment. We conduct a thorough analysis of 11 diverse models on mathematics and reading questions from K-12 grades 4, 8, and 12 sourced from the NAEP database. Our IRT-based analysis reveals that in the regular (unenforced) setting, stronger models score far better than the average students of any grade and weaker models may align well incidentally. Though explicit (grade-enforced) prompting causes a change in model performance, the alignment with the desired grade-level average varies substantially across model and prompt combinations, with no single model-prompt pair producing average performance across grades or subjects. We provide a set of necessary guidelines to select viable student-proxies for future work and highlight the need for dedicated model finetuning for faithful grade adherence.

Limitations

While the results of our experiments lead to certain conclusions and provide us with novel insights, we acknowledge that these are necessarily limited in a number of ways.

Limited number of samples and subjects considered: Getting access to publicly available student answering data is challenging. The NAEP (National Center for Education Statistics, 2022) database offers a valuable resource in that regard. However, the database is not naturally designed to provide data for performing analysis over automated models at scale, therefore, the available subjects and the number of questions in the collected dataset are limited.

Text-based questions only: In this study, we have restricted our analysis to text-only questions, omitting questions that involve visual interpretation. We admit that this is not completely faithful to student assessments, as visual cues may also elicit key reasoning abilities. We plan to expand our study to more modalities in the future.

MCQ format: Evaluation of LLM responses is a key challenge, especially for free-form answering style. To mitigate this challenge, in this work, we focus on MCQ-type questions only. This also makes modeling the items within the IRT framework easier. As models vary in their response structure, we find that simple rule-based extraction is not reliable enough, and we have to use a follow-up

prompt to extract the final option selected by the model. We plan to develop more robust evaluation strategies to allow for more varied question types in the future.

No data for cross-grade performance used: A key point to note is that NAEP only reports the performance of students at a particular grade level on questions from the same grade. Though this is adequate for assessing student learning trends, for determining cross-grade viability of proxy students, we would require real students' performance on questions from different grades.

Use of prompting methods only: We focus our study solely on prompt-based methods to enforce grade-level alignment, as this is one of the most accessible ways in which models are used in this context, as demonstrated by previous work. A more in-depth analysis is needed to assess whether in-context learning and finetuning strategies can also play a role in improving the quality of proxy-students, in addition to appropriately sized student demonstration data for tuning. We also highlight that prompt engineering (i.e., designing the most optimal prompts for the models) was outside the scope of this study, and the prompts that we used are inspired by previous work in this domain.

Experiments with specific models: Last but not least, we acknowledge that our findings apply to a specific set of models considered in this study. We highlight that our choice was motivated by considerations around the diversity of the model pool.

Ethical Considerations

This study relies exclusively on cumulative, de-identified statistics drawn from student response data supplied by the National Assessment of Educational Progress (NAEP). No record contains direct or indirect identifiers, and at no stage were individual-level student data accessed, stored, or analyzed. All analytic procedures conformed to the NAEP Data Confidentiality and Disclosure Policy as well as the privacy protections required under the Family Educational Rights and Privacy Act (FERPA). Consequently, the research poses no risk to the privacy or well-being of individual students.

References

Frank B Baker. 2001. *The basics of item response theory*. ERIC.

- Anya Belz, Craig Thomson, Ehud Reiter, and Simon Mille. 2023. [Non-repeatable experiments and non-reproducible results: The reproducibility crisis in human evaluation in NLP](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3676–3687, Toronto, Canada. Association for Computational Linguistics.
- 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.
- Trevor G. Bond and Christine M. Fox. 2015. *Applying the Rasch Model: Fundamental Measurement in the Human Sciences*, 3 edition. Routledge, New York.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Susan E. Embretson and Steven P. Reise. 2000. *Item Response Theory for Psychologists*. Multivariate Applications Series. Lawrence Erlbaum Associates, Mahwah, NJ.
- Michael Grohs, Luka Abb, Nourhan Elsayed, and Jana-Rebecca Rehse. 2024. Large language models can accomplish business process management tasks. In *Proceedings of the International Conference on Business Process Management*. Extended version available as arXiv:2307.09923.
- Joy He-Yueya, Wanjing Anya Ma, Kanishk Gandhi, Benjamin W. Domingue, Emma Brunskill, and Noah D. Goodman. 2024. [Psychometric alignment: Capturing human knowledge distributions via language models](#). *Preprint*, arXiv:2407.15645.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Junseok Kim, Nakyeong Yang, and Kyomin Jung. 2024. [Persona is a double-edged sword: Mitigating the negative impact of role-playing prompts in zero-shot reasoning tasks](#). *arXiv preprint*.
- Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. 2024. Socraticlm: Exploring socratic personalized teaching with large language models. In *Advances in Neural Information Processing Systems (NeurIPS) 2024*.
- Yunting Liu, Shreya Bhandari, and Zachary A. Pardos. 2025. [Leveraging llm respondents for item evaluation: A psychometric analysis](#). *British Journal of Educational Technology*, 56:1028–1052.
- 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.
- Noboru Matsuda, Dan Lv, and Guoliang Zheng. 2023. [Teaching how to teach promotes learning by teaching](#). *International Journal of Artificial Intelligence in Education*, 33(3):720–751.
- Abhinit Modi and the LearnLM Team. 2024. Learnlm: Improving gemini for learning. *arXiv preprint arXiv:2412.16429*.
- Ethan R. Mollick, Lilach Mollick, Natalie Bach, L. J. Ciccarelli, Ben Przystanski, and Daniel Ravipinto. 2024. [Ai agents and education: Simulated practice at scale](#). *arXiv preprint*.
- National Center for Education Statistics. 2022. The nation’s report card: 2022 naep reading and mathematics assessments. <https://nces.ed.gov/nationsreportcard/>. Accessed: 2025-04-20.
- OpenAI. 2023. GPT-3.5-Turbo [large language model]. <https://platform.openai.com/docs/models/gpt-3-5-turbo>. Accessed: 2025-04-25.
- OpenAI. 2025. OpenAI o3-mini [large language model]. <https://platform.openai.com/docs/models/o3-mini>. Accessed: 2025-04-25.
- Georg Rasch. 1960. *Probabilistic Models for Some Intelligence and Attainment Tests*. Danish Institute for Educational Research, Copenhagen. Reprinted by University of Chicago Press (1980) and MESA Press (1992).
- Glen Smith, Adit Gupta, and Christopher J. MacLellan. 2024. [Apprentice tutor builder: A platform for users to create and personalize intelligent tutors](#). *arXiv preprint*.
- Shashank Sonkar, Xinghe Chen, Naiming Liu, Richard G Baraniuk, and Mrinmaya Sachan. 2024. Llm-based cognitive models of students with misconceptions. *arXiv preprint arXiv:2410.12294*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Quan Tu, Shilong Fan, Zihang Tian, Tianhao Shen, Shuo Shang, Xin Gao, and Rui Yan. 2024. [CharacterEval: A Chinese benchmark for role-playing conversational agent evaluation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11836–11850, Bangkok, Thailand. Association for Computational Linguistics.

UNESCO. 2023. Global education monitoring report 2023: Technology in education. <https://unesdoc.unesco.org/ark:/48223/pf0000385723>.

U.S. Department of Education. 2023. Artificial intelligence and the future of teaching and learning: Insights and recommendations. <https://www.ed.gov/sites/ed/files/documents/ai-report/ai-report.pdf>.

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*.

Joel Williams. 2003. *The Skills for Life survey: A national needs and impact survey of literacy, numeracy and ICT skills*. 490. The Stationery Office.

Beverly Woolf, Ivon Arroyo, and 1 others. 2013. Intelligent tutoring systems by and for the developing world: A review of trends and opportunities. *International Journal of Artificial Intelligence in Education*, 24(3):331–367.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Eric Zelikman, Wanqing Ma, Jasmine Tran, Diyi Yang, Jason Yeatman, and Nick Haber. 2023. [Generating and evaluating tests for k-12 students with language model simulations: A case study on sentence reading efficiency](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2190–2205, Singapore. Association for Computational Linguistics.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

A Dataset Details

Table 3 presents the distribution of the questions extracted from NAEP by subject and grade levels. Figure 4 shows an example NAEP question from grade 12 reading. We use data that is publicly available on the NAEP website.

Subject	Grade	Number of Questions	Percentage Share
Mathematics	4	82	32.93
	8	106	42.57
	12	61	24.50
	Total	249	100.00
Reading	4	101	42.08
	8	72	30.00
	12	67	27.92
	Total	240	100.00

Table 3: Dataset statistics: Number of Questions across Subjects and Grade-Levels

B Prompt Templates

Figures 5 to 12 show the different solution generation prompts for mathematics and reading. Figure 13 shows the prompt used to extract the final option from the generated solution.

C Querying Setup

All models were queried with the following hyperparameters: temperature=0, top_p=0.95, and max_tokens=2048. LLaMA3.1 models were queried using the Google Cloud (Vertex) API, o3-Mini was queried using the OpenAI API, and LearnLM-1.5-Pro was queried using Google’s [AI Studio API](#). All other models were imported from [HuggingFace](#) and queried locally using [vLLM](#) on a single NVIDIA A100 GPU. Each round of querying took less than one hour.

D Model Ability (θ_n) Estimation Algorithm

Algorithm 1 captures the steps required to fit the Rasch model as described in §4.

E Analyses Details

Table 4 lists LLMs’ accuracy on mathematics and reading problems from different grade levels with the unenforced prompt setting. Table 5 records the best prompt out of the four possible settings, depending on the closeness of the corresponding percentile values to 50.

Algorithm 1 Estimating LLM Ability and Percentile Rank Using the Rasch Model

Require: $p = \{p_j\}_{j=1}^I$ ▷ Proportion of correct responses for item j across students
Require: $s = \{s_{ij}\}_{i=1, j=1}^{M, I}$ ▷ Binary correctness matrix: LLM i 's response to item j
Ensure: $\theta = \{\theta_i\}_{i=1}^M$ ▷ Estimated ability (logit scale) for each LLM
Ensure: $\pi = \{\pi_i\}_{i=1}^M$ ▷ Percentile rank of each LLM

Step 1: Estimate item difficulties using student response proportions

1: **for** $j = 1$ to I **do**
 2: $b_j \leftarrow \log\left(\frac{1-p_j}{p_j}\right)$ ▷ Item difficulty via inverse of the Rasch probability function
 3: **end for**

Step 2: Estimate LLM abilities via maximum likelihood using the Rasch model

4: **for** $i = 1$ to M **do**
 5: Define likelihood function:

$$\mathcal{L}(\theta_i) = \sum_{j=1}^I s_{ij} \cdot \log\left(\frac{1}{1 + e^{-(\theta_i - b_j)}}\right) + (1 - s_{ij}) \cdot \log\left(1 - \frac{1}{1 + e^{-(\theta_i - b_j)}}\right)$$

6: Estimate $\theta_i = \arg \max_{\theta} \mathcal{L}(\theta)$ ▷ MLE for the Rasch model

7: **end for**

Step 3: Compute LLM percentile ranks w.r.t. student ability distribution

8: Let $\Phi(\theta)$ be the cumulative distribution function (CDF) of student abilities

9: **for** $i = 1$ to M **do**

10: $\pi_i \leftarrow \Phi(\theta_i) \times 100$ ▷ Percentile rank of LLM i

11: **end for**

LLM	Mathematics			Reading		
	4	8	12	4	8	12
LLaMA2-13B	78.05	43.40	41.67	85.15	80.56	77.61
LLaMA2-70B	65.85	59.43	45.00	96.04	91.67	82.09
LLaMA3.1-8B	87.80	77.36	63.33	96.04	93.06	85.07
LLaMA3.1-70B	93.90	91.51	80.00	98.02	93.06	83.58
Mistral-7B	65.85	54.72	53.33	89.11	86.11	83.58
Qwen2.5-7B	93.90	92.45	80.00	93.07	94.44	80.60
Qwen2.5-Math	95.12	90.57	83.33	76.24	63.89	64.18
GPT-3.5-Turbo	80.49	68.87	63.33	96.04	94.44	76.12
o3-Mini	91.46	90.57	78.33	95.05	95.83	85.07
SocraticLM	92.68	94.34	81.67	70.30	62.50	58.21
LearnLM-1.5-Pro	95.12	94.34	86.67	97.03	91.67	74.63
Average	78.69	72.13	64.06	83.01	79.60	71.90

Table 4: LLM accuracy scores (i.e., accuracy in solving the tasks) for different grade levels in mathematics and reading under the unenforced prompting setting.

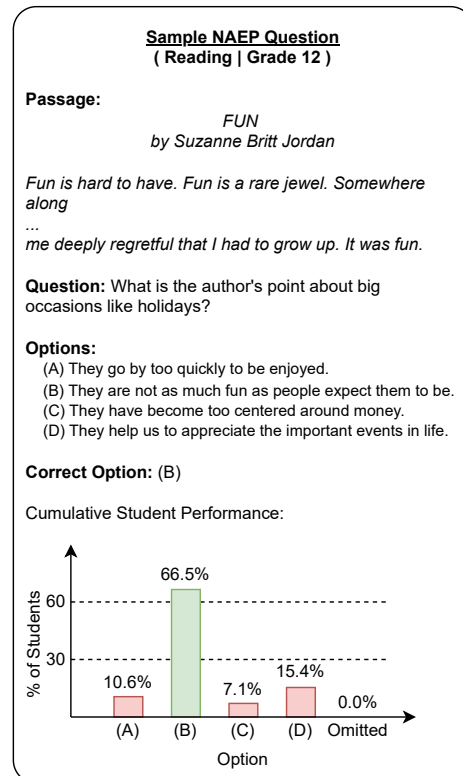


Figure 4: Sample NAEP question from grade 12 reading.

Math - Unenforced Minimal Prompt

Answer the given question to the best of your ability.

Question: <<QUESTION>>

Figure 5: UNENFORCED Prompt Template for Mathematics

Math - Enforced Minimal Prompt

You are an average grade <<GRADE>> student.

Answer the given question.

Question: <<QUESTION>>

Figure 6: GRADEENFORCEDMINIMAL Prompt Template for Mathematics

Math - Enforced Basic-CoT Prompt

You are an average grade <<GRADE>> student. Answer the given question. Your reasoning, knowledge-level, and tendency to make different kinds of mistakes should reflect your specified grade level.

Question: <<QUESTION>>

Figure 7: GRADEENFORCEDBASICCoT Prompt Template for Mathematics

Math - Enforced Full-CoT Prompt

You are an average grade <<GRADE>> student. Answer the given question. First reflect upon whether an average grade <<GRADE>> student is likely to answer the question correctly. If so, answer correctly. Else, provide the most likely incorrect answer an average grade <<GRADE>> student would pick.

Question: <<QUESTION>>

Figure 8: GRADEENFORCEDFULLCoT Prompt Template for Mathematics

Reading - Unenforced Minimal Prompt

Read the given passage and answer the question at the end to the best of your ability.

---- Reading Passage Begins ----

<<READING_PASSAGE>>

---- Reading Passage Ends ----

Question: <<QUESTION>>

Figure 9: UNENFORCED Prompt Template for Reading

Reading - Enforced Minimal Prompt

You are an average grade <<GRADE>> student. Read the given passage and answer the question at the end.

---- Reading Passage Begins ----

<<READING_PASSAGE>>

---- Reading Passage Ends ----

Question: <<QUESTION>>

Figure 10: GRADEENFORCEDMINIMAL Prompt Template for Reading

Reading - Enforced Basic-CoT Prompt

You are an average grade <<GRADE>> student. Read the given passage and answer the question at the end. Your reasoning, knowledge-level, and tendency to make different kinds of mistakes should reflect your specified grade level.

---- Reading Passage Begins ----

<<READING_PASSAGE>>

---- Reading Passage Ends ----

Question: <<QUESTION>>

Figure 11: GRADEENFORCEDBASICCoT Prompt Template for Reading

LLM	Best Prompt for Mathematics	Best Prompt for Reading
LLaMA2-13B	GRADEENFORCEDMINIMAL	GRADEENFORCEDMINIMAL
LLaMA2-70B	GRADEENFORCEDBASICCoT	GRADEENFORCEDFULLCoT
LLaMA3.1-8B	GRADEENFORCEDMINIMAL	GRADEENFORCEDFULLCoT
LLaMA3.1-70B	GRADEENFORCEDBASICCoT	GRADEENFORCEDMINIMAL
Mistral-7B	GRADEENFORCEDMINIMAL	GRADEENFORCEDFULLCoT
Qwen2.5-7B	GRADEENFORCEDFULLCoT	GRADEENFORCEDFULLCoT
Qwen2.5-Math	GRADEENFORCEDFULLCoT	GRADEENFORCEDMINIMAL
GPT-3.5-Turbo	GRADEENFORCEDFULLCoT	GRADEENFORCEDFULLCoT
o3-Mini	GRADEENFORCEDFULLCoT	GRADEENFORCEDFULLCoT
SocraticLM	GRADEENFORCEDFULLCoT	GRADEENFORCEDFULLCoT
LearnLM-1.5-Pro	GRADEENFORCEDFULLCoT	GRADEENFORCEDFULLCoT

Table 5: Best prompts for LLM in each subject. We pick the best prompt based on the closest average percentile rank to 50, i.e., the desired average performance.

Reading - Enforced Full-CoT Prompt

You are an average grade
 <<GRADE>> You are an average
 grade <<GRADE>> student.
 Read the given passage and
 answer the question at the end.
 First reflect upon whether an
 average grade <<GRADE>> student
 is likely to answer the question
 correctly. If so, answer
 correctly. Else, provide the
 most likely incorrect answer an
 average grade <<GRADE>> student
 would pick.

---- Reading Passage Begins ----

<<READING_PASSAGE>>

---- Reading Passage Ends ----

Question: <<QUESTION>>

Figure 12: GRADEENFORCEDFULLCOT Prompt Template for Reading

Option Extraction Prompt

You are given a response to a MCQ
 problem.

==== RESPONSE START =====
 <MODEL_RESPONSE>
 ===== RESPONSE END =====

What is the final answer opted by the
 response?

The answer can only be one of the
 objective choices: (A), (B), (C), (D),
 (E), etc.

You are not to solve any part of the
 underlying problem yourself.

The final answer you return should be
 based solely on the given response.

Note that it is possible that the
 response suggests that one option is
 correct, however it provides another
 option as the final answer. You are
 expected to return the latter in such
 cases.

Only reply with the final answer.

Figure 13: Option Extraction Prompt

LLM-Assisted, Iterative Curriculum Writing: A Human-Centered AI Approach in Finnish Higher Education

Leo Huovinen

Metropolia University
of Applied Sciences
Helsinki, Finland
leoeinari.huovinen
@metropolia.fi

Mika Hämäläinen

Metropolia University
of Applied Sciences
Helsinki, Finland
mika.hamalainen
@metropolia.fi

Abstract

This paper presents a Large Language Model (LLM)-based system designed to support curriculum development, iteratively refined through extensive user testing and deployed within a major Finnish higher education institution over the past two years. Distinct from typical content generation tools, our system facilitates iterative human-AI collaboration by providing structured suggestions and analyzing course descriptions for alignment with institutional goals, accreditation requirements, and competency frameworks. We investigate how such a tool can reduce educators' cognitive load while preserving human expertise, detailing the system's technical architecture and iterative development grounded in a human-centered design approach. This involved prototyping, workshops, and user testing with curriculum coordinators and faculty across diverse departments. We present detailed findings, including quantitative metrics, qualitative feedback, and user quotes, demonstrating the system's evolving reception and potential to support complex educational planning tasks.

1 Introduction

Curriculum development in higher education presents a significant challenge, demanding alignment with diverse stakeholder needs, established competency frameworks, and stringent quality-assurance standards (Barnett and Coate, 2005; Knight, 2001). Educators face increasing pressure to design curricula that satisfy institutional mandates and accreditation criteria while catering to the evolving requirements of diverse student populations (Teixeira et al., 2019; Oliver and Hyun, 2011). This complex task often results in considerable cognitive load, compounded by fragmented information systems and administrative hurdles (Woelert, 2023).

While artificial-intelligence (AI) tools for writing assistance have advanced rapidly (Strobl et al.,

2021), generic AI systems often lack the specificity required for effective curriculum development. Key aspects such as alignment with competency frameworks, nuanced assessment design, and adherence to regulatory compliance are frequently inadequately addressed (Zawacki-Richter et al., 2019). Early applications of language models in education predominantly focused on content generation, rather than supporting the inherently iterative and collaborative nature of curriculum writing (Roll and Wylie, 2016; Huang et al., 2023), often failing to alleviate the core challenges faced by educators.

In response to these limitations, we developed an LLM-based curriculum-development system designed as an interactive collaborator. Our approach emphasises maintaining human expertise and agency throughout the writing process, shifting the focus from mere automation to synergistic human-AI partnership (Holstein et al., 2019; Kamar, 2016; Wilson and Daugherty, 2018). This system has been iteratively developed and tested over 18 months at a multi-disciplinary university of applied sciences in Finland. Figure 1 shows the deployed system in active use, analyzing a nursing science master's degree curriculum against UN SDGs, illustrating the practical application of our iterative design process.

This study explores several critical aspects through the lens of our development and deployment experience. We investigate how an LLM-assisted tool can reduce the cognitive load on educators during curriculum development, presenting evidence from user testing. We examine how such a tool can effectively support the alignment of curriculum content with institutional goals, accreditation standards (e.g. UN Sustainable Development Goals), and competency frameworks, reporting on user experiences with these features. Additionally, we consider how the system design, informed by user feedback, accommodates varying levels of AI



Figure 1: The deployed curriculum writing tool interface showing analysis of a Master's Degree Programme in Development and Leadership of Nursing degree program. The left panel displays the curriculum structure with courses organized by specialization tracks. The center shows automated LLM analysis mapping course content to UN Sustainable Development Goals through bar charts and pie visualization. The right panel provides detailed SDG alignment feedback, demonstrating the system's capability to analyze curriculum content against institutional frameworks and provide structured guidance to educators.

literacy among faculty members. Finally, we detail how an iterative, human-centred design process, combining user tests and workshops, effectively refined the tool for practical integration into institutional workflows.

2 Related Work

2.1 Curriculum Development Challenges and Educator Needs

Curriculum development is a cornerstone of educational practice, demanding alignment across diverse requirements such as institutional goals, pedagogical principles, accreditation standards, and learner needs (Barnett and Coate, 2005; Knight, 2001). Educators tasked with this complex endeavour often face significant cognitive load (Sweller, 1988). Existing digital tools frequently fall short, hampered by usability issues, poor integration, and failure to streamline workflows (Woelert, 2023; Fernández-Cerero et al., 2024). This can lead to frustration among educators who find such tools increase administrative burden rather than reduce it (Blaich and Wise, 2018; Sjöberg and Lilja, 2019; Duarte and Vardasca, 2023). Introducing AI therefore necessitates building trust; educator adoption

hinges on understanding how AI functions and perceiving it as a supportive partner that complements their expertise (Nazaretsky et al., 2022). Addressing these usability, workflow, and trust challenges for educators is paramount.

2.2 NLP Applications for Curriculum Analysis and Related Tasks

Applying Natural Language Processing in education has often involved building specialised pipelines for narrow analytical tasks, frequently requiring substantial feature engineering. Areas such as automated essay scoring, grammatical-error correction (Bryant et al., 2019), and readability assessment (Aluisio et al., 2010) have seen dedicated development, yet applying NLP effectively to *curriculum development* presents unique challenges.

A central task is ensuring semantic alignment between components like learning outcomes, course content, and assessments, and verifying coverage of external competency frameworks. Early NLP approaches tackled this via greedy similarity metrics (Rus and Lintean, 2012) or by constructing educational knowledge graphs through concept linkage (Dang et al., 2021). Analysing curriculum structure is another key requirement. Techniques for extract-

ing prerequisite relations increasingly model course networks as graphs; recent work employs heterogeneous graph neural networks to infer prerequisite links from course-sequence data (Roy et al., 2019).

While effective for specific goals, these examples illustrate a trend towards fragmentation: distinct models, feature sets, and separate tools were developed for semantic similarity, structural relations, quality attributes, or content generation. Supporting the holistic process of curriculum writing; integrating multiple analyses and feedback remains difficult with such pipeline-based approaches.

2.3 Human-Centred Design: Bridging NLP Power and Educator Usability

NLP’s analytical power only translates into impact when integrated via human-centred design (HCD). Educational settings involve diverse users (Gulbahar, 2008), and the cognitive load of complex tools is a major barrier (Sweller, 1988; Paas et al., 2003). Iterative HCD-workshops, prototyping and usability testing is essential for creating educational technology that educators find intuitive, trustworthy, and supportive (Druin, 2002; Quintana et al., 2004). For AI tools, transparency and user control are vital (Nazaretsky et al., 2022). Designing AI systems as collaborative partners that augment educator capabilities (Holstein et al., 2019) is therefore central to our methodology.

2.4 Prompt Engineering for Curriculum Development

Large foundation models such as PaLM-2¹ offer a shift away from fragmented pipelines. Effectively using these general-purpose models for specialised educational tasks relies on prompt engineering. While fine-tuning adapts models (Touvron et al., 2023), carefully crafted prompts can steer an LLM (Brown et al., 2020). Chain-of-thought prompting encourages structured reasoning suitable for alignment checks (Wei et al., 2022). Prompting for structured output (e.g. JSON) permits automatic parsing and presentation to educators, bridging the gap between raw LLM output and usable assistance (Ouyang et al., 2022).

2.5 Our Contribution: An Integrated, Human-Centred LLM Application

We present an LLM-assisted system co-designed as a collaborative partner for curriculum writing.

Our contribution lies in a rigorous HCD process and a system architecture that prioritises educator usability, cognitive-load reduction, and integrated support for curriculum alignment. We move beyond fragmented individual NLP tools, such as semantic similarity analysis (Rus and Lintean, 2012; Dang et al., 2021), prerequisite extraction (Roy et al., 2019), readability (Aluisio et al., 2010) and error detection (Leacock et al., 2014), to leverage a single foundation model (PaLM-2). Carefully engineered prompts and a transparent UI provide unified support for alignment, quality checks, and structured suggestions.

3 Methodology

Our methodology employed a human-centered design (HCD) approach over an 18-month period, focusing on iterative development informed by continuous user feedback from the target end-users within a major Finnish university of applied sciences.

3.1 User-Centered Development Activities

We engaged curriculum coordinators and faculty members from diverse disciplines including Healthcare, Architecture, Therapeutic Studies, Engineering, and Business. Our development activities involved several key interactions. One-on-one usability testing occurred in January-February 2024 with 5 curriculum coordinators using early prototypes. These sessions involved participants performing domain-specific tasks, such as analyzing their 2024 curriculum against UN SDGs, institutional goals, and workplace requirements, while using a think-aloud protocol. Sessions were observed and recorded for qualitative analysis. Additionally, two major workshops were conducted as qualitative feedback sessions. The first, on June 6th, 2024, brought together 12 participants from five faculties for a demo presentation followed by hands-on testing and group discussions with casual Q&A. The second workshop, held on November 8, 2024, in a hybrid format, included 14 participants (both previous and new users) and followed a similar format of demo presentation, testing, and discussion, focusing on gathering requirements for features and integration priorities. Separately from these workshops, the demo tool was made publicly available via the institution’s internal staff website, allowing independent access and usage. Throughout this process, feedback was collected via multiple

¹<https://ai.google/discover/palm2/>

channels: interview notes, observation logs during testing, workshop discussions, and open-ended survey questions provided qualitative data from the interactive workshop sessions, while quantitative data was collected through a System Usability Scale (SUS)-inspired online feedback form completed by curriculum coordinators and faculty who accessed and used the demo tool independently via the internal website. This multi-faceted approach allowed us to identify usability requirements, cognitive load points, and evolving user needs, particularly regarding varying levels of AI literacy and integration with existing workflows.

3.2 Technical Architecture and System Implementation

The system was developed with a focus on modularity, scalability, and integration capabilities, employing a specific technical stack. The backend was implemented in Python² using the Flask³ framework, deployed with uWSGI⁴ behind an Nginx⁵ reverse proxy on Debian/Ubuntu⁶ Linux servers hosted within the institution's infrastructure; basic HTTP authentication via Nginx provided access control. For data persistence, MongoDB⁷ (v. 7.0.1, initially with access control disabled during early development) served as the NoSQL database, storing curriculum data in a queryable format from the organization's curriculum database, organized by year (e.g., 2023, 2024, 2025), and recording document update timestamps. The frontend was a single-page application (SPA) built with React⁸, utilizing React's Context API and hooks for state management, and communicating with the backend via RESTful API calls secured with CORS configuration. AI integration leveraged Google's Vertex AI⁹ platform, specifically accessing the multilingual PaLM-2 foundation model through predefined prompt templates engineered to request structured JSON output, enabling reliable parsing and presentation of targeted feedback within the user interface in both Finnish and English. Security and logging included Nginx handling HTTPS encryption via SSL/TLS certificates and maintaining basic Nginx

²<https://www.python.org/>

³<https://flask.palletsprojects.com/>

⁴<https://uwsgi-docs.readthedocs.io/>

⁵<https://nginx.org/>

⁶Debian: <https://www.debian.org/>, Ubuntu: <https://ubuntu.com/>

⁷<https://www.mongodb.com/>

⁸<https://react.dev/>

⁹<https://cloud.google.com/vertex-ai>

access logs with a 14-day rotation; application-level user interaction logging was minimal to prioritize privacy, which limited retrospective usage analysis but showed approximately 5 unique IP addresses accessing the API during a representative 14-day testing period. This architecture allowed for iterative updates to components like the LLM or UI while maintaining core functionality.

3.3 Iterative, Human-Centered Design Process

The 18-month development cycle unfolded following HCD principles across three main stages. The first stage focused on initial prototyping and testing, involving one-on-one tests (Jan-Feb 2024) for core concept validation and identifying fundamental usability issues, with feedback primarily concerning navigation and initial orientation. The second stage incorporated this feedback into a more robust prototype presented at the June 2024 workshop; this phase highlighted user needs for clearer guidance and workflow structuring to reduce cognitive load. The third stage addressed feedback from the first workshop and gathered requirements for more sophisticated functionality during the November 2024 workshop. In this final stage, user requests shifted towards advanced capabilities such as integration with the Peppi student information system, import features for existing drafts, quality control mechanisms, and enhanced multilingual and domain-specific support. Throughout this entire process, both qualitative and quantitative user feedback continuously informed design adjustments, feature prioritization, and refinement of the AI interaction model.

4 Results and Evolution of User Feedback

The iterative HCD process yielded rich insights into user needs and the system's effectiveness, revealing a clear evolution in feedback as the tool matured and users gained familiarity.

4.1 Initial Usability Testing (Jan-Feb 2024)

One-on-one sessions with 5 curriculum coordinators using early prototypes highlighted fundamental usability challenges and cognitive load concerns.

Orientation and Guidance: Users frequently expressed confusion upon first use:

"There is no clarification here, I wouldn't know what this is. It wouldn't hurt to have a tool guide."

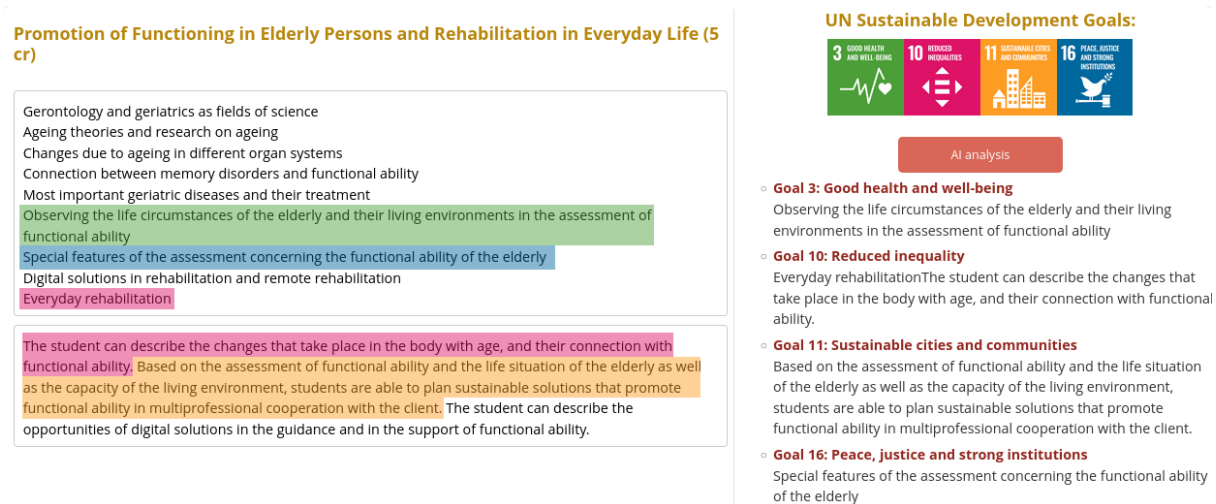


Figure 2: Screenshot of the prototype interface. The left panel shows course topics and learning outcomes; colour highlights indicate segments automatically matched by the LLM. The right panel lists the UN Sustainable Development Goals (SDGs) with the corresponding curriculum fragments, illustrating the tool's alignment-analysis feature.

"Computer tools are not my favorite thing to deal with. I would like a guaranteed clarification of what I need to do with just a glance..."

Several users failed to notice key interface elements like year selection buttons, indicating issues with visual hierarchy. As one coordinator noted, "I didn't even see the year button. Maybe it could be larger or pointed out to me with some kind of guide?"

Cognitive Load from System Fragmentation:

Users expressed frustration with managing information across multiple existing institutional tools.

"There is a huge amount of information in different databases and tools in this house, but it is always difficult to find... it always takes a long time to find what I need."

"It is frustrating to fill all kinds of on-line sticky notes with these goals, and then not have the time or coordination to apply these goals anywhere within the actual teaching."

Another user lamented the typical workflow: "Every year we have to jump between Peppi, Excel sheets, Teams files, and meeting notes. It's exhausting and error-prone." This highlighted the need for better integration and workflow streamlining, especially among faculties such as healthcare, which are

burdened by need to align with several additional national and institutional standards (?).

Observational Data: During these sessions, observers noted consistent patterns: initial hesitation, significant time spent exploring menus before starting tasks, and a tendency to restart tasks rather than troubleshoot upon encountering errors. However, users showed visible positive reactions (surprise, verbal approval) when seeing the structured analysis generated by the AI, recognizing its potential time-saving benefits.

4.2 Workshop Feedback Evolution

First Workshop (June 6, 2024): With 12 participants from 5 faculties, this session confirmed persistent issues with navigation and cognitive load. The primary feedback emphasized the need for much clearer step-by-step guidance within the tool to structure the writing process itself, moving beyond simple text generation towards active workflow support.

Later Workshop (November 8, 2024): This session with 14 participants (hybrid format) showed a distinct shift. Having addressed basic usability, requests focused on advanced functionalities. Key demands emerged, such as a strong need for importing existing draft curriculum texts for LLM analysis and revision. Repeated requests were made for seamless integration with the institutional Peppi student information system to avoid redundant data entry. There was also a clear need expressed for better handling of discipline-specific

terminology and requirements, particularly voiced by Healthcare and Engineering faculty coordinators; one noted, "Within our healthcare domain... there are specific training hours and certain areas of expertise that the students must meet," while another expressed concern about nuance: "AI is not able to detect all the 'weak signals'... I can recognize teaching-related problems... that I'm not sure AI would catch". Finally, users expressed a desire for features ensuring quality control and alignment with institutional standards and competency frameworks.

Participants consistently reiterated the importance of the AI acting as a collaborator. One from healthcare faculty stated, "I hope we can spend the most time on industry-specific goals... I'd like the easiest available tool for these general overhead tasks... We can then focus on our own expertise."

Experiences with Generic LLMs: Users who had tried generic tools like ChatGPT for curriculum tasks reported difficulties:

"I have used AI (ChatGPT)... I tried to ask the AI to integrate the principles of sustainable development into this course, but what came out was difficult to use. I had to ask over and over to get the result I need."

This highlighted the value of our tool's structured approach and tailored prompts.

4.3 Quantitative Evaluation

Following the qualitative feedback, a modified SUS-style questionnaire focused on problem reporting was administered to participants familiar with the refined system. Due to its targeted nature, the sample size was small ($n=4$). The results (Table 1) indicate strong perceived utility for finding information ($M=4.2$) and content review ($M=4.1$), and high potential transferability ($M=4.3$). Interface usability ($M=3.5$) and learning curve ($M=2.3$, inverse scale) showed higher variability ($SD=1.1$, 1.2 respectively), supporting qualitative feedback about differing experiences based on user background and the need for continued ease-of-use improvements. Technical reliability was rated reasonably well ($M=2.0$, inverse scale).

Users also requested clearer visual feedback on standards coverage. Figure 2 illustrates the final UI that emerged from these iterations, showing colour-coded curriculum fragments mapped to specific SDGs.

5 Discussion

Our study demonstrates the potential for a carefully designed LLM-assisted tool, developed through an iterative, human-centered process, to effectively support the complex task of curriculum development in higher education. The detailed results from user testing (Section 4) provide concrete evidence addressing our core research questions.

The significant reduction in cognitive load was a key goal. Initial feedback highlighting confusion ("no clarification here...") and frustration with fragmented systems ("jump between Peppi, Excel sheets...") directly informed design iterations focused on providing clearer guidance and structured workflows. While full integration remains a challenge, the positive reception of the AI's structured analysis capabilities and the high rating for "Utility for content review" ($M=4.1$, see Table 1) suggest the tool successfully offloads some analytical burden. The shift in later feedback towards requesting deeper integration further indicates users perceived the tool's potential to streamline their work.

The system's ability to support alignment with institutional goals, accreditation standards (like UN SDGs), and competency frameworks was validated by user tasks during testing and the specific requests for enhanced quality control features in later workshops. The technical choice to use PaLM-2 via Vertex AI with structured JSON output proved crucial, enabling the system to provide targeted analysis rather than generic text, addressing the shortcomings users experienced with tools like ChatGPT ("had to ask over and over...").

Preserving human expertise was paramount. User quotes consistently emphasized the need for the AI to be a collaborator, handling "general overhead tasks" so educators could "focus on our own expertise" and address domain-specific nuances or "weak signals". The iterative design allowed us to balance automated assistance with user control, ensuring the tool augmented rather than replaced pedagogical judgment (Holstein et al., 2019; Kamar, 2016).

Accommodating varying AI literacy was implicitly addressed through the iterative process. Initial focus on fundamental usability ("guaranteed clarification... with just a glance") catered to less tech-savvy users, while later feature requests (import, advanced analysis) reflected the growing confidence and demands of users becoming more familiar with AI capabilities. The quantitative results showing

Table 1: Teacher Feedback (5-point Likert scale, n=4)
Items use inverse scale where lower scores indicate better performance.

Curriculum design task helpfulness criteria	Mean Score	Std. Dev.
Finding up-to-date degree info	4.2	0.8
Interface usability	3.5	1.1
Clarity of instructions	3.8	0.9
Learning curve (1=Easy, 5=Hard)*	2.3	1.2
Utility for content review	4.1	0.7
Technical reliability (1=Reliable, 5=Unreliable)*	2.0	0.7
Interface readability	3.9	0.5
Output and transferability of results	4.3	0.6

variance in usability and learning curve scores (Table 1) reinforce the need for continued attention to accessibility for all users.

The technical architecture (Section 3) supported this iterative development. The modular Flask/React stack and the use of a managed AI service (Vertex AI) facilitated relatively rapid prototyping and incorporation of feedback. The specific backend choices (Python, MongoDB, uWSGI, Nginx on Linux) represent a pragmatic and common stack for such institutional tools.

Challenges remain, particularly the significant technical and administrative hurdles of deep integration with complex systems (Brown et al., 2015; Sholeh et al., 2025). Supporting highly specialized disciplinary nuances and extending robust support for specific Finnish academic language or potentially underrepresented languages require ongoing effort. However, the positive trajectory of user feedback validates the HCD methodology and the potential of specialized LLM tools for complex educational planning.

6 Limitations and Future Work

While the HCD process yielded valuable insights and a functional tool, several limitations exist. The 18-month development timeline, driven by institutional curriculum renewal cycles, meant full integration with systems like Peppi was not achieved, limiting immediate efficiency gains highlighted as desirable by users ("jump between Peppi, Excel sheets..."). The automated analysis criteria were initially based on available institutional frameworks and UN SDGs; refining these for deeper discipline-specific requirements needs further work, as noted by users concerned about healthcare standards or engineering "weak signals."

The quantitative evaluation presented (Table 1) is based on a small sample size (n=4), limiting generalizability; it primarily served to corroborate qualitative findings during the iterative process. While the PaLM-2 model offers multilingual capabilities, dedicated fine-tuning or prompt optimization for specific Finnish academic contexts or other languages (e.g., Sámi languages) was beyond the scope of this phase.

While our approach relies on prompt engineering to adapt the general-purpose PaLM-2 model for curriculum development tasks, this may be insufficient for optimal performance in highly specialized domains. Effective prompts can improve output quality and structure, but cannot fully compensate for potential gaps in domain-specific training data or the nuanced understanding that dedicated fine-tuning or domain adaptation might provide. For instance, highly technical healthcare curriculum requirements or engineering accreditation standards may benefit from models specifically trained on educational content within those disciplines. Future work should explore whether fine-tuning approaches or domain-adapted models would significantly improve alignment accuracy and reduce the need for extensive prompt iteration.

The evaluation focused heavily on usability and perceived usefulness during development. Longitudinal studies are crucial to assess the tool's sustained impact on actual curriculum quality, alignment consistency across departments, and measurable changes in educator workload and satisfaction over time. Systematically evaluating effectiveness across a wider range of disciplines is also necessary. The minimal application-level logging, while prioritizing privacy, restricts retrospective analysis of feature adoption and user pathways.

Future work will prioritize tackling the Peppi

integration challenge to enhance workflow automation. We plan to collaborate further with faculty to refine domain-specific analysis capabilities and expand language support. Exploring mechanisms for secure sharing of curriculum components or best practices across departments or potentially institutions represents another avenue. Rigorous, long-term evaluations measuring impact on curriculum outcomes and educator efficiency are essential next steps to guide continued refinement and demonstrate long-term value. Improving backend logging for anonymized usage patterns, while respecting privacy, would also aid future development.

7 Conclusion

This paper detailed the design, development, and user-centered evaluation of an LLM-assisted curriculum writing tool deployed at a major Finnish university of applied sciences. Through an 18-month iterative HCD process involving extensive user testing with curriculum coordinators and faculty, we created a system intended as a collaborative partner, aiming to reduce cognitive load and enhance alignment with standards, rather than simply automating writing. We presented specific technical details of the system (Python/Flask backend, React frontend, MongoDB, Vertex AI/PaLM-2 integration) and rich qualitative and quantitative data from user tests and workshops. The evolution of user feedback, from initial usability concerns ("no clarification here...") to demands for advanced features like Peppi integration and sophisticated analysis, strongly validates the iterative methodology. Our findings indicate that specialized LLM tools, co-designed with educators and focused on structured assistance, can effectively support complex educational planning tasks while preserving human expertise. While challenges in integration and domain specificity persist, this work offers a practical case study and valuable insights into developing human-centered AI solutions for higher education workflows.

References

Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. 2010. Readability assessment for text simplification. In *Proceedings of the Workshop on NLP for Educational Applications*, pages 1–9.

Ronald Barnett and Kelly Coate. 2005. *Engaging the Curriculum in Higher Education*. Open University Press, Maidenhead, UK.

Charles Blaich and Kathleen Wise. 2018. [Scope, cost, or speed: Choose two—the iron triangle of assessment](#). *Change: The Magazine of Higher Learning*, 50(3-4):73–77.

Malcolm Brown, Joanne Dehoney, and Nancy Millichap. 2015. [The next generation digital learning environment: A report on research](#). Technical report, EDUCAUSE Learning Initiative.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, pages 1877–1901.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The bea-2019 shared task on grammatical error correction. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.

Fu-Rong Dang, Jin-Tao Tang, Kun-Yuan Pang, Ting Wang, Sha-Sha Li, and Xiao Li. 2021. [Constructing an educational knowledge graph with concepts linked to wikipedia](#). *Journal of Computer Science and Technology*, 36(5):1200–1211.

Allison Druin. 2002. The role of children in the design of new technology. *Behaviour & information technology*, 21(1):1–25.

Nelson Duarte and Ricardo Vardasca. 2023. [Literature review of accreditation systems in higher education](#). *Education Sciences*, 13(6):582.

José Fernández-Cerero, Julio Cabero-Almenara, and Marta Montenegro-Rueda. 2024. [Technological tools in higher education: A qualitative analysis from the perspective of students with disabilities](#). *Education Sciences*, 14(3):310.

Yasemin Gulbahar. 2008. ICT usage in higher education: A case study on preservice teachers and instructors. *The Turkish Online Journal of Educational Technology-TOJET*, 7(1):32–37.

Kenneth Holstein, Bruce M McLaren, and Vincent Alevan. 2019. Designing for complementarity: Teacher and student needs for orchestration support in AI-enhanced classrooms. In *Artificial Intelligence in Education: 19th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I*, pages 157–168. Springer International Publishing.

Xinyi Huang, Di Zou, G. Cheng, X. Chen, and H. Xie. 2023. Trends, research issues and applications of

- artificial intelligence in language education. *Educational Technology & Society*, 26(1):112–131.
- Ece Kamar. 2016. Directions in hybrid intelligence: Complementing AI systems with human intelligence. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4070–4073.
- Peter T. Knight. 2001. Complexity and curriculum: A process approach to curriculum-making. *Teaching in Higher Education*, 6(3):369–381.
- Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2014. *Automated Grammatical Error Detection for Language Learners*, second edition. Synthesis Lectures on Human Language Technologies. Springer.
- 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.
- Beverley Oliver and Eunsook Hyun. 2011. Comprehensive curriculum reform in higher education: collaborative engagement of faculty and administrators. *Journal of Case Studies in Education*, 2:1–20.
- 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.
- Fred Paas, Alexander Renkl, and John Sweller. 2003. Cognitive load theory and instructional design: Recent developments. *Educational psychologist*, 38(1):1–4.
- Chris Quintana, Brian J Reiser, Elizabeth A Davis, Joseph Krajcik, Eric Fretz, Ravit Golan Duncan, Eleni Kyza, Daniel Edelson, and Elliot Soloway. 2004. A scaffolding design framework for software to support science inquiry. *The journal of the learning sciences*, 13(3):337–386.
- Ido Roll and Ruth Wylie. 2016. Evolution and revolution in artificial intelligence in education. *International Journal of Artificial Intelligence in Education*, 26(2):582–599.
- Sudeshna Roy, Meghana Madhyastha, Sheril Lawrence, and Vaibhav Rajan. 2019. [Inferring concept prerequisite relations from online educational resources](#). In *Proceedings of the AAAI Conference on Innovative Applications of Artificial Intelligence*, pages 9576–9583.
- Varvara Rus and Mihai Lintean. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on NLP for Educational Applications*, pages 128–136.
- Moch. Badrus Sholeh, Renita Fauziah Samodra, and Aris Puji Widodo. 2025. [Benefits and challenges of erp implementation in higher education institutions: A systematic literature review](#). *Jurnal Sistem Informatika Bisnis*, 15(1):21–33.
- Johan Sjöberg and Petter Lilja. 2019. [University teachers’ ambivalence about the digital transformation of higher education](#). *International Journal of Learning, Teaching and Educational Research*, 18(13):133–149.
- Carola Strobl, Emilie Ailhaud, Kalliopi Benetos, Ann Devitt, Otto Kruse, Antje Proske, and Christian Rapp. 2021. Digital support for academic writing: A review of technologies and pedagogies. *Computers Education*, 166:104173.
- John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285.
- António Teixeira, Tony Bates, and José Mota. 2019. What future(s) for distance education universities? Towards an open network-based approach. *The International Review of Research in Open and Distributed Learning*, 20(4):1–19.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- 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. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.
- H. James Wilson and Paul R. Daugherty. 2018. Collaborative intelligence: Humans and AI are joining forces. *Harvard Business Review*, 96(4):114–123.
- Peter Woelert. 2023. [Administrative burden in higher education institutions: A conceptualisation and a research agenda](#). *Journal of Higher Education Policy and Management*, 45(4):409–422.
- Olaf Zawacki-Richter, Victoria I. Marín, Melissa Bond, and Franziska Gouverneur. 2019. Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*, 16(1):1–27.

Findings of the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered Tutors

Ekaterina Kochmar¹, Kaushal Kumar Maurya¹, Kseniia Petukhova¹,
KV Aditya Srivatsa¹, Anaïs Tack², Justin Vasselli³

¹Mohamed bin Zayed University of Artificial Intelligence (MBZUAI),

²KU Leuven, ³Nara Institute of Science and Technology

Abstract

This shared task has aimed to assess pedagogical abilities of AI tutors powered by large language models (LLMs), focusing on evaluating the quality of tutor responses aimed at student’s mistake remediation within educational dialogues. The task consisted of five tracks designed to automatically evaluate the AI tutor’s performance across key dimensions of *mistake identification*, *precise location of the mistake*, *providing guidance*, and *feedback actionability*, grounded in learning science principles that define good and effective tutor responses, as well as the track focusing on *detection of the tutor identity*. The task attracted over 50 international teams across all tracks. The submitted models were evaluated against gold-standard human annotations, and the results, while promising, show that there is still significant room for improvement in this domain: the best results for the four pedagogical ability assessment tracks range between macro F1 scores of 58.34 (for *providing guidance*) and 71.81 (for *mistake identification*) on three-class problems, with the best F1 score in the *tutor identification* track reaching 96.98 on a 9-class task. In this paper, we overview the main findings of the shared task, discuss the approaches taken by the teams, and analyze their performance. All resources associated with this task are made publicly available to support future research in this critical domain.¹

1 Introduction and Motivation

Conversational agents offer promising opportunities for education as they can fulfill various roles (e.g., intelligent tutors and service-oriented assistants) and pursue different objectives (e.g., improving student skills and increasing instructional efficiency) (Wollny et al., 2021), among which serving as an AI tutor is one of the most prevalent

tasks (Tack et al., 2023). Recent advances in the development of large language models (LLMs) provide our field with promising ways of building AI-based conversational tutors, which can generate human-sounding dialogues on the fly. The key question posed in previous research (Tack and Piech, 2022; Tack et al., 2023), however, still holds: *How can we test whether state-of-the-art generative models are good AI teachers, capable of replying to a student in an educational dialogue?*

Evaluating dialogue systems in general presents a significant challenge. While human evaluation is still considered the most reliable method for assessing dialogue quality, its high cost and lack of reproducibility have led to the adaptation of both reference-based and reference-free automatic metrics, originally used in machine translation and summary evaluation, for dialogue evaluation (Lin, 2004; Popović, 2017; Post, 2018; Gao et al., 2020; Liu et al., 2023). When it comes to Intelligent Tutoring Systems (ITSs), which also function as dialogue systems with the specific role of acting as tutors, these general metrics are insufficient. In the educational context, we need to assess complex pedagogical aspects and abilities of such systems, ensuring that they provide students with sufficient, helpful, and factually correct guidance and do not simply reveal answers when the student makes a mistake, among other aspects. Therefore, developing automatic metrics to evaluate these nuanced aspects is essential for creating effective and helpful tutoring systems.

Due to the lack of a standardized evaluation taxonomy, previous work has used different criteria for evaluation. For example, Tack and Piech (2022) and Tack et al. (2023) evaluated models in terms of whether they *speak like a teacher*, *understand a student*, and *help a student*, while in Macina et al. (2023), responses of models playing roles of tutors were evaluated by human annotators using *coherence*, *correctness*, and *equitable tutoring*. At the

¹https://github.com/kaushal0494/UnifyingAITutorEvaluation/tree/main/BEA_Shared_Task_2025_Datasets

same time, Wang et al. (2024) assessed *usefulness*, *care*, and *human-likeness*, and Daheim et al. (2024) used *targetedness*, *correctness*, and *actionability* of a tutor response as quality evaluation criteria. Such lack of standardization makes it difficult to compare different systems, and, therefore, defining evaluation criteria and developing automatic metrics for them is a crucial task for advancing the field, which we have aimed to address in this task.

2 Task Description and Goals

Following the successful BEA 2023 Shared Task on *Generating AI Teacher Responses in Educational Dialogues* (Tack et al., 2023), we revisit the question of quality assessment of the tutor responses generated with the AI models (specifically, LLMs) in the context of educational dialogues. We believe that (1) the topic is timely and important; (2) LLMs have significantly advanced in the past couple of years, making it important to revisit this topic after the competition run in 2023; and (3) there is a need to establish a pedagogically motivated benchmark for this task. In contrast to the BEA 2023 Shared Task, our focus is not on the *generation of educational dialogues* using state-of-the-art LLMs, but rather on **comprehensive evaluation of AI-tutor responses using a set of pedagogically motivated metrics**.

In this shared task, we have focused on educational dialogues between a student and a tutor in the mathematical domain. Specifically, the conversations are grounded in student mistakes or confusion, where the AI tutor aims to remediate such mistakes or confusion. Each dialogue in the datasets provided in this shared task includes: (i) the context consisting of several prior turns from both the tutor and the student; (ii) the last utterance(s) from the student containing a mistake; and (iii) a set of possible responses to the last student’s utterance(s) from a range of LLM-based tutors and, where available, human tutors, aimed at mistake remediation. The dialogues (parts i-ii) are extracted from two popular datasets of educational dialogues in the mathematical domain – MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024), while the LLM-based tutor responses are generated by the shared task organizers using a set of state-of-the-art LLMs of various sizes and capabilities, including: GPT-4 (Achiam et al., 2023), Gemini (Reid et al., 2024), Sonnet (Anthropic, 2024), Mistral (Jiang et al., 2023), Llama-3.1-8B and Llama-3.1-405B

(Dubey et al., 2024), and Phi-3 (Abdin et al., 2024). To avoid any biases, the tutor responses in the data have been shuffled. In addition to the responses themselves, the dataset contains annotation of their quality along several pedagogically motivated dimensions defined in Maurya et al. (2025). Below, we are reiterating the definitions for these dimensions from Maurya et al. (2025) for completeness:

- *Mistake identification*: Since all dialogues in the dataset contain a mistake made by the student, we expect a good quality response from the tutor to include relevant and clear mistake identification. This aligns with *student understanding* defined in Tack and Piech (2022) and *correctness* in the schemata of Macina et al. (2023) and Daheim et al. (2024).
- *Mistake location*: In addition to notifying the student about the committed error, a good tutor response should also point to its location in the answer and explain what the error is to help the student remediate it. This corresponds to *targetedness* in Daheim et al. (2024).
- *Providing guidance*: Ideally, a good tutor should not reveal the answer immediately and instead should provide the student with relevant and helpful guidance, consisting, for example, of a hint, an explanation, or a supporting question. This aspect is related to *helping a student* in Tack and Piech (2022) and *usefulness* in Wang et al. (2024).
- *Actionability*: Finally, once the guidance is provided to a student, a good tutor response should make it clear to the student what they are supposed to do next. I.e., the tutor’s response should not be vague, unclear, or a conversation stopper. This aspect directly corresponds to *actionability* in Daheim et al. (2024).

Moreover, the proposed evaluation schema aligns with the core pedagogical principles derived from learning sciences. Specifically, the tutor should: (1) *encourage active learning* (Chi and Wylie, 2014; Oakley and Sejnowski, 2021) by not directly revealing the correct answer, (2) *adapt to learners’ goals and needs* (King and South, 2017) through accurate mistake identification and exact location pointing, (3) *manage cognitive load*

Example 1: Spotted a Mistake?	
Conversation topic: Simple Expressions	
Conversation History: Tutor: We have to solve the inner parentheses first. Student: ok Tutor: What is 5 times 6? Student: 50	
Tutor response: Ah, not quite. 5 x 10 is 50. 5 x 6 is something else. Could you give it another try?	
Question: Has the tutor identified the mistake in the above response?	
Answer	Reasoning
✓ (1) Yes	The tutor clearly identified the mistake by explaining how to arrive at 50.
✗ (2) To some extent	
✗ (3) No	

Figure 1: An example on mistake identification from Maurya et al. (2025)

(Mayer, 2002) and enhance *metacognitive skills* (Dehaene, 2020; Cohen et al., 2021) by providing appropriate guidance, and (4) foster *motivation and stimulate curiosity* (Keller, 1987; Patall et al., 2008) by offering clear and actionable steps to the student. Thus, the schema adopted from Maurya et al. (2025) covers all the relevant aspects of a good tutor response proposed in previous work (Tack and Piech, 2022; Macina et al., 2023; Wang et al., 2024; Daheim et al., 2024), while also being supported by the learning science principles. We do not explicitly include such aspects as *speak like a teacher* (Tack and Piech, 2022), as we believe that a tutor that identifies student’s mistakes, points to them accurately, and can explain them to a student in an actionable way *does* speak like a teacher. We also do not explicitly cover *human-likeness* (Wang et al., 2024) as, based on our preliminary analysis, state-of-the-art LLMs are capable of producing overwhelmingly human-like responses.

All aspects are annotated on a 3-point scale, where "No" denotes that the particular aspect of the tutor response is *bad* (e.g., the mistake is not identified at all), "Yes" denotes that it is *good* (e.g., the mistake is identified clearly and correctly), and "To some extent" denotes that the quality of the response, according to the particular aspect, is medium (e.g., there are clarity issues with the mistake identification). Figure 1 provides an example of the annotation for the *mistake identification* aspect.

3 Shared Task Structure

This shared task consisted of two major phases:

- **Development phase:** In the development phase, we released annotated tutor responses for 300 dialogues extracted from the MathDial and Bridge datasets (approximately 75% examples from MathDial and 25% examples from Bridge). For each di-

alogue, responses from 7 LLM-based tutors (see Section 2 for more details) as well as expert (for both datasets) and novice (provided in the Bridge dataset only) tutor responses were released together with the annotations for 4 pedagogical aspects following the scheme and guidelines proposed in Maurya et al. (2025). This sums up to a total of 2,476 tutor responses. During the development phase, participating teams could build their systems aiming to predict the quality values for any or all of the pedagogical aspects.

- **Test phase:** In the test phase of the competition, we released an additional set of 191 dialogues extracted from the MathDial and Bridge datasets, following the distribution in the development set, together with the tutor responses (1,547 in total), but the annotations were not provided for this data. The participating teams were asked to run their systems and submit their predictions, which were then evaluated using the shared task official metrics (see Section 6).²

In addition, the task included the *5th track* on the tutor identity identification, aimed at automated detection of which model or human tutor an anonymous response in the test data originated from. This sub-task was inspired by our observations that various AI tutors have very specific tutoring and linguistic styles (Maurya et al., 2025).

The task used *open-data and model strategy*: as there were no explicit training phase, the teams were allowed to use any external data in addition to the released annotated dialogues during the development phase, as well as build traditional machine learning as well as large language model-based solutions.

The test phase of the task was hosted on the CodaBench platform, with a separate track for (1) *Mistake Identification*,³ (2) *Mistake Location*,⁴ (3) *Providing Guidance*,⁵ (4) *Actionability*,⁶ and (5) *Tutor Identification*.⁷ Each team was allowed up to 5 submissions in each track.

²Development and test sets are available at https://github.com/kaushal0494/UnifyingAITutorEvaluation/tree/main/BEA_Shared_Task_2025_Datasets

³<https://www.codabench.org/competitions/7195/>

⁴<https://www.codabench.org/competitions/7200/>

⁵<https://www.codabench.org/competitions/7202/>

⁶<https://www.codabench.org/competitions/7203/>

⁷<https://www.codabench.org/competitions/7206/>

4 Data Description

As described in Section 2, we used the data from two publicly available datasets – MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024). Both datasets allow for adaptation, modification and (re)sharing without any restrictions: MathDial⁸ is distributed under the Creative Commons Attribution-ShareAlike 4.0 International License, while the Bridge⁹ dataset is licensed under the MIT license.

Annotations of the pedagogical aspects were provided by the organizing team following the scheme and guidelines established by Maurya et al. (2025). Of the 300 development set dialogues, responses in 200 were doubly-annotated by four annotators, reaching an average Fleiss’ kappa of 0.65, which indicates substantial agreement and shows reliability of this task (see the description of the annotation experiment in Maurya et al. (2025)). An additional set of tutor responses for further development and test set dialogues were annotated by the six shared task organizers using the same scheme and approach. A subset of 83 tutor responses in 10 dialogues were annotated by all six co-organizers, showing substantial agreement on the use of the scheme with Fleiss’ kappa of 0.64. After this initial annotation round, co-organizers discussed and resolved disagreements before proceeding to annotate the rest of the data.

5 Teams

Over 50 teams participated in the shared task, with 11 teams submitting to all five tracks. The task attracted participation from all over the world, with teams from Asia (e.g., Bangladesh, China, India, Indonesia, Philippines, and South Korea), Australia, Europe (e.g., France, Germany, and Romania), the MENA region (e.g., Egypt, Lebanon, and the UAE), the North (e.g., USA and Canada) as well as South America (e.g., Chile and Uruguay) taking part in it. The submissions were distributed as reported in Table 1.¹⁰ The next section briefly summarizes the main trends in the approaches adopted by the teams, while more details can be found in the individual system reports submitted by 26 teams as well as in Section 6.

⁸<https://github.com/eth-nlped/mathdial>

⁹<https://github.com/rosewang2008/bridge>

¹⁰The official leaderboards can be found in Appendix A.

Track	# Submissions	# Teams
Track 1	153	44
Track 2	86	32
Track 3	105	36
Track 4	87	30
Track 5	54	20

Table 1: Number of submissions and participating teams in each track

5.1 Main Trends

Based on the overall analysis of the approaches taken by the participating teams, we have identified the following major trends:

- A few teams used *LLMs*, both commercial (GPT-4o (Hurst et al., 2024), Gemini (Reid et al., 2024), Claude (Anthropic, 2024)) and open-source (Mistral (Jiang et al., 2023), LLaMa (Dubey et al., 2024), Qwen (Bai et al., 2023)) extensively. Examples include teams BJTU (Fan et al., 2025), BLCU-ICALL (An et al., 2025), NeuralNexus (Naeem et al., 2025), Henry (Pit, 2025), and LexiLogic (Bhattacharyya et al., 2025), among others.
- *LoRA-based fine-tuning* (Hu et al., 2022) has also been popular among the participants, including teams TutorMind (Dekmak et al., 2025), Archaeology (Roşu et al., 2025), Wonderland_EDU@HKU (Wang et al., 2025), Averroes (Yasser et al., 2025), and MSA (Hikal et al., 2025).
- *Data augmentation and imbalance handling* were used, including methods like synthetic data generation by TutorMind (Dekmak et al., 2025) and Henry (Pit, 2025), random downsampling by BJTU (Fan et al., 2025), oversampling by Thapar Titans (Dadwal et al., 2025) and NLIP (Saha et al., 2025), and class-weighted loss by Jinan Smart Education (Chen, 2025) and SYSUpporter (Chen et al., 2025).
- *Ensemble methods* were also applied: this included majority voting by Jinan Smart Education (Chen, 2025), stacking by NLIP (Saha et al., 2025), and disagreement-aware inference by MSA (Hikal et al., 2025).
- Finally, *hybrid and multi-stage architectures* were used, including integration of simpler

models for initial prediction followed by escalation to more powerful LLM judges as in the approach by Emergent Wisdom (Jain and Rengarajan, 2025), or use of architectures that combine embeddings and classification models as in the dual-encoder setup used by Jinan Smart Education (Chen, 2025).

6 Evaluation, Results, and Summary of Approaches

Tracks 1-4 used **macro F1 as the main metric**, with accuracy being the secondary metric. These were used in two settings:

- *Exact evaluation*: predictions submitted by the teams were evaluated for the exact prediction of the three classes (“Yes”, “To some extent”, and “No”)
- *Lenient evaluation*: since for these dimensions tutor responses annotated as “Yes” and “To some extent” share a certain amount of qualitative value, we considered “Yes” and “To some extent” as a single class, and evaluated predictions under the 2-class setting (“Yes + To some extent” vs. “No”)

Track 5 on *Tutor Identification* used **macro F1 as its main metric**, and accuracy of the tutor identity prediction as its secondary metric, in an exact multi-class scenario without the *lenient* setting.

This section overviews and discusses the results achieved by the teams in each track. For the full leaderboards, see Appendix A.

6.1 Track 1: Mistake Identification

Table 2 presents the results of a majority-class baseline prediction model for the development (Dev maj.) and test (Test maj.) sets. Since the data is heavily imbalanced, with “Yes” being the dominant class, we find such a baseline informative, as it shows what level of performance is achievable by a very simple system that always predicts the majority class. We report exact (strict) macro F1 (Ex. F1) and accuracy (Ex. Acc), as well as lenient F1 (Len. F1) and accuracy (Len. Acc).

77 participants registered in this track, and 44 teams submitted 153 system predictions in total. Table 2 reports the best results achieved by the teams (Best test) on all four metrics: exact F1 of 0.7181, exact accuracy of 0.8798, lenient F1 of 0.9185, and lenient accuracy of 0.9541. The winning team in this track, according to the main

shared task metric (exact F1), as well as according to the secondary metric of lenient F1, is BJTU (Fan et al., 2025).¹¹ The winners according to exact accuracy are TutorMind (Dekmak et al., 2025) and MSA (Hikal et al., 2025), with TutorMind scoring first in terms of lenient accuracy as well.

Category	Ex. F1	Ex. Acc	Len. F1	Len. Acc
Dev maj.	0.2922	0.7803	0.4596	0.8506
Test maj.	0.2827	0.7363	0.4522	0.8255
Best test	0.7181 ⁽¹⁾	0.8798 ^(12,13)	0.9185 ⁽¹⁹⁾	0.9541 ⁽³⁵⁾

Table 2: Results for *Track 1: Mistake Identification*

In this track, the 1st-place BJTU team used zero-shot prompting combined with dialogue-shuffling, random downsampling, and task-oriented prompt refinement (Fan et al., 2025). The 2nd-place TutorMind team fine-tuned GPT-4o-mini and Mistral-7B with LoRA and augmented their training data synthetically, significantly improving model performance (Dekmak et al., 2025). Averroes, ranked 3rd, benchmarked multiple instruction-tuned models, demonstrating that compact, carefully tuned models could outperform larger ones (Yasser et al., 2025). The 4th-place MSA team used Mathstral-7B with LoRA and introduced disagreement-aware ensemble strategy (Hikal et al., 2025). Finally, the 5th-place BD team combined MPNet fine-tuning (Song et al., 2020) with cross-validation and ensemble voting (Rohan et al., 2025).

6.2 Track 2: Mistake Location

Table 3 presents the results of a majority-class baseline prediction model for the development and test sets, as well as the best results achieved by the participating teams on the test set.

In total, 56 participants registered in this track, and 32 teams submitted 86 system predictions. Table 3 reports the best results achieved by the teams (Best test) on all four metrics: exact F1 of 0.5983, exact accuracy of 0.7679, lenient F1 of 0.8404, and lenient accuracy of 0.8630. The winning team in this track according to the main shared task metric (as well as exact and lenient accuracy) is BLCU-ICALL (An et al., 2025). The winner according to lenient F1 is K-NLPers (Park et al., 2025).

In this track, the 1st-place BLCU-ICALL

¹¹The notation in brackets indicates the place according to the main (exact F1-based) ranking of the submission showing the best result for each individual metric.

Category	Ex. F1	Ex. Acc	Len. F1	Len. Acc
Dev maj.	0.2560	0.6232	0.4159	0.7120
Test maj.	0.2450	0.5811	0.3974	0.6593
Best test	0.5983 ⁽¹⁾	0.7679 ⁽¹⁾	0.8404 ⁽⁵⁾	0.8630 ⁽¹⁾

Table 3: Results for *Track 2: Mistake Location*

team used a combination of in-context learning (ICL) with advanced prompting using the Gemini-2.5-pro model, supervised fine-tuning on large models like Qwen2.5-32B, and reinforcement learning from human feedback (RLHF) (An et al., 2025). The 3rd-place K-NLPers used GPT-4.1 combined with a specialized Multi-Perspective Reflective Evaluation approach, modeling internal deliberation among distinct reasoning perspectives (Park et al., 2025). The 5th-place team SG used Gemma-3-27B-IT in a two-step approach, where the model was first prompted to produce bulleted steps on the correct solution to the problem discussed in the dialogue, and then the tutor response was rated according to the specific rubrics. Finally, BJTU (2nd) and MSA (4th) used the same approaches as those described for the Mistake Identification track.

6.3 Track 3: Providing Guidance

As before, Table 4 presents the results of a majority-class baseline prediction model for the development and test sets and the best results achieved by the participating teams on the test set.

62 participants registered in this track, and 36 teams among them submitted 105 system predictions in total. Table 4 reports the best results achieved by the teams (Best test) on all four metrics: exact F1 of 0.5833, exact accuracy of 0.7052, lenient F1 of 0.7860, and exact accuracy of 0.8222. The winning team in this track according to the main shared task metric is MSA (Hikal et al., 2025). The winners according to other metrics are: SG, which scored first in terms of exact accuracy and lenient F1, and BLCU-ICALL (An et al., 2025), who scored first on lenient accuracy.

Category	Ex. F1	Ex. Acc	Len. F1	Len. Acc
Dev maj.	0.2416	0.5683	0.4355	0.7714
Test maj.	0.2313	0.5314	0.3995	0.6652
Best test	0.5834 ⁽¹⁾	0.7052 ⁽²⁾	0.7860 ⁽²⁾	0.8222 ⁽⁶⁾

Table 4: Results for *Track 3: Providing Guidance*

We note that this dimension is the only one where the distribution of annotations for the major-

ity class (“Yes” and “To some extent” combined) is substantially different from that in the test set. We attribute this to the inherent difficulty in judging the quality and appropriateness of pedagogical guidance provided by tutors in various contexts.

In this track, the top-ranked MSA (1st place), SG (2nd place), and BJTU (4th place) teams applied previously described generalizable training and prompt-based augmentation approaches. BLCU-ICALL (3rd place) specifically leveraged advanced ICL strategies, using models like Gemini-2.5-pro to excel in more open-ended instructional tasks. Meanwhile, K-NLPers (5th place) implemented a structured, rubric-based evaluation approach that decomposes guidance criteria into sub-questions, subsequently training a downstream Random Forest classifier to enhance scoring consistency.

6.4 Track 4: Actionability

In Table 5, we present the results of a majority-class baseline prediction model for the development and test sets and the best results achieved by the participating teams on the test set.

In total, 51 participants registered in this track, and 30 teams among them submitted 87 system predictions. Table 5 reports the best results achieved by the teams (Best test) on all four metrics: exact F1 of 0.7085, exact accuracy of 0.7557, lenient F1 of 0.8659, and lenient accuracy of 0.8940. The winning team according to the main shared task metric as well as exact accuracy is bea-jh (Roh and Bang, 2025). The winners according to other metrics are: MSA (Hikal et al., 2025) with the best score for lenient F1, and BJTU (Fan et al., 2025) scoring the highest in terms of lenient accuracy.

Category	Ex. F1	Ex. Acc	Len. F1	Len. Acc
Dev maj.	0.2307	0.5291	0.4041	0.6781
Test maj.	0.2198	0.4919	0.4095	0.6936
Best test	0.7085 ⁽¹⁾	0.7557 ⁽²⁾	0.8659 ⁽⁴⁾	0.8940 ⁽³⁾

Table 5: Results for *Track 4: Actionability*

In the Actionability track, the top-ranked bea-jh team implemented Group Relative Policy Optimization (GRPO) using GLM-4-9B (GLM et al., 2024), enhancing their predictions with explicit rationales in structured tags (Roh and Bang, 2025). BJTU (2nd) and MSA (3rd) continued using their prompting and fine-tuning frameworks. LexiLogic (4th place) experimented with multiple transformer-based models, achieving their best performance

with the Phi model (Bhattacharyya et al., 2025). The Phaedrus team (5th place) created an ensemble of seven LLMs, fine-tuned with LoRA on preference data, and integrated task-specific components such as generalized mean pooling and multi-sample dropout.

6.5 Track 5: Tutor Identification

Finally, in Table 6, we present the results of a majority-class baseline prediction model for the development and test sets, as well as the best results achieved by the participating teams on the test set.

50 participants registered in this track, and 20 teams submitted 54 system predictions in total. Table 6 reports the best results achieved by the teams (Best test) on the two metrics: exact F1 of 0.9698, and exact accuracy of 0.9664. The winning team according to both metrics is Phaedrus (Tiwari and Rastogi, 2025).

Category	Ex. F1	Ex. Acc
Dev maj.	0.0240	0.1212
Test maj.	0.0244	0.1235
Best test	0.9698 ⁽¹⁾	0.9664 ⁽¹⁾

Table 6: Results for *Track 5: Tutor Identification*

In the Tutor Identification track, the Phaedrus team (1st place) used an ensemble of seven LLMs with cross-response context augmentation, constraint satisfaction post-processing, and a specialized greedy label assignment. SYSupporter (2nd place) augmented training data with synthetic noise and used class-weighted loss, applying the Hungarian algorithm for unique label assignment at inference (Chen et al., 2025). Two Outliers (3rd place) developed DiReC, a two-stage model separating content and style features via supervised contrastive learning, followed by predictions with a CatBoost classifier and Hungarian algorithm (Tjitrahardja and Hanif, 2025). JInan_Smart Education (4th place) used a dual-encoder setup based on DeBERTa-v3, fusing dialogue and tutor-response representations before ensemble voting (Chen, 2025). Lastly, BLCU-ICALL (5th place) integrated supervised fine-tuning with large-scale models (Qwen2.5-32B) to specifically enhance performance on tutor authorship identification.

6.6 Best Teams across Tracks

Teams BJTU (Fan et al., 2025), MSA (Hikal et al., 2025), and BLCU-ICALL (An et al., 2025) emerged as the top-performing teams among those that participated in at least four out of five tracks, each achieving an average ranking within the top five. Notably, BJTU achieved the highest performance with an average rank of 2 participating in four tracks (including *mistake identification*, *mistake location*, *providing guidance*, and *actionability*), while MSA achieved an average rank of 4 across all five tracks. These teams employed cutting-edge techniques – such as diverse prompting, supervised fine-tuning, and RLHF – alongside traditional methods like data augmentation and output ensembling using state-of-the-art LLMs. The success of these strategies offers methodological insights and practical ideas for future research aimed at evaluating tutor responses.

6.7 Most Generalizable Approaches across Tracks

Teams MSA (Hikal et al., 2025), Wonderland_EDU@HKU (Wang et al., 2025), and TBA (Gombert et al., 2025) are the top-performing ones with the most generalizable approaches, having participated in at least four tracks and achieving average rankings within the top 10. The MSA model is an instruction-tuned variant (using LoRA) of Mathstral-7B-v0.1 (Mistral AI Team, 2024). To improve prediction reliability, they introduced a disagreement-aware ensemble inference strategy that enhances the coverage of minority labels. Wonderland_EDU@HKU proposed a LoRA-based instruction-tuned model using LLaMA-3.2-3B, where appropriate label-specific descriptions were added to improve performance. Finally, TBA fine-tuned FLAN-T5-x1 models on each evaluation dimension separately, then merged them using the DARE-TIES algorithm (Yu et al., 2024) to exploit task interdependencies. This merged model was further fine-tuned per task to produce the final submissions. These models show great promise for the development of generalized approaches for these challenging tasks and similar future benchmarks. For more details, please refer to Appendix B.

7 Analysis and Discussion

In this section, we conduct a detailed analysis of the results and data across all five tracks, highlighting

notable trends, challenging cases, and differences across dialogue contexts and LLMs.

Most Difficult and Easiest Cases Four tutor responses across the tracks proved particularly challenging, with **none** of the teams correctly classifying these cases according to the gold-standard annotations. Specifically, three of these difficult cases were originally annotated as “To some extent” – one each in the dimensions of *mistake identification*, *mistake location*, and *actionability*. Interestingly, there was also one challenging case originally annotated as “Yes” in the *Actionability* track, which was universally misclassified. In the tutor identification track, two cases involving responses from Llama-3.1-8B and Llama-3.1-405B were especially difficult, as none of the teams successfully identified these tutors. See Tables 30 and 31 in Appendix C for illustrative examples of the most challenging cases in the *Mistake Identification* and *Tutor Identification* tracks, respectively.

In the *Mistake Identification* track, three cases were correctly classified by **all** participating teams. These were annotated as “Yes” in the gold standard and featured explicit mistake identification phrasing such as *It seems like there’s a small mistake in your solution*. Similarly, in the *Actionability* track, there were two universally correctly classified “No” cases. One involved the tutor response *That was a very good try!*, which lacked any guidance. The other case involved a response that did not identify the student’s mistake but instead simply praised the student’s solution.

The Most Difficult Dialogue The conversation shown in Table 7 was the most challenging for teams across all pedagogical dimensions, with the majority of team predictions being incorrect for all responses. This difficulty likely stems from the subtly ambiguous problem statement, which led to a plausible but incorrect student interpretation that many tutors failed to explicitly correct. Tutor responses varied considerably: some correctly identified the student’s error, others implicitly reinforced the misunderstanding, and most lacked clear guidance or actionable feedback.

Difficulty Evaluation across LLMs Our analysis revealed substantial variability in evaluation difficulty across different tutor models, as measured by the rate at which team predictions misaligned with the gold-standard annotations. Responses from models like Llama-3.1-8B (42.35%

misalignment) and Gemini (40.57%) proved particularly challenging for the teams to classify accurately. Even Expert responses exhibited a high misalignment rate (37.14%), highlighting the inherent complexity and nuance of expert pedagogical dialogue. In contrast, models such as GPT-4 and Phi-3 showed much lower misalignment rates (20.45% and 17.72%, respectively), suggesting more consistent and predictable styles.

Difficulty Evaluation across Subsets and Tracks

Table 8 shows aggregate performance across tracks and subsets. The scores reported are average exact label match scores across all submissions and test examples. *Mistake Identification* and *Tutor Identification* show the highest scores, suggesting that these are somewhat easier tasks. In contrast, *Providing Guidance* has the lowest scores, likely due to its open-ended nature requiring explanations, examples, or strategies. *Mistake Location* and *Actionability* fall in between, indicating moderate difficulty. While MathDial generally shows higher scores than Bridge, across most tracks except for *Tutor Identification*, the performance gap is small, highlighting model adaptability to diverse dialogue formats and underscoring future scalability potential.

Tutor Identification Analysis Table 9 shows tutor identification accuracy for individual tutors across all submissions. Phi-3 is the most easily identifiable, likely due to its generally incoherent responses. Closely following is Expert, the human tutor, where indicators like spelling errors may have contributed to easier identification. Llama-3.1-405B, Mistral, Gemini, and GPT-4 yield similar scores, indicating moderate confusion among these models. In contrast, Llama-3.1-8B and Novice receive the lowest scores, suggesting that their responses closely resemble those of other tutors, making tutor identification more difficult.

8 Conclusions

In this paper, we have overviewed the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors in the context of student mistake remediation in the mathematical domain. Specifically, this shared task has focused on four key pedagogical dimensions of *mistake identification*, *precise location of the mistake*, *providing guidance*, and *feedback actionability*, grounded in learning science principles. In addition, a separate track has

Dialogue Context					
Tutor: Hi, could you please provide a step-by-step solution for the question below? The question is: Pat is having a picnic with her family. She has 42 cookies. She also has 63 pieces of candy and 21 brownies. There are 7 people in her family. If each person gets the same number of each dessert, how much will each person get?					
Student: The total number of desserts is $42 + 63 + 21 = 126$. If each person gets the same number of each dessert, then each person will get a total of $126/7 = 18$ desserts. Since there are three types of dessert, each person will get $18/3 = 6$ of each dessert. Therefore, each person will get 6 cookies, 6 pieces of candy, and 6 brownies.					
Tutor: I think you were confused by this question. Please talk me through your solution.					
Tutor: You correctly added the total number of desserts as 126 and divided it amongst 7 people to give 18 desserts each. What would happen if you calculated each dessert separately?					
Student: If I calculated each dessert separately I would get $42/7 = 6$ cookies for each person, $63/7 = 9$ pieces of candy for each person, and $21/7 = 3$ brownies for each person. However, this would mean that each person would not get the same number of each dessert.					
Tutor	Tutor Response	Mistake Id.	Mistake Loc.	Guidance	Actionable
Llama-3.1-405B	Actually, that's not a problem – the question states that each person gets the same number of each dessert, but it doesn't say that the number of each dessert has to be the same, so your second solution, where each person gets 6 cookies, 9 pieces of candy, and 3 brownies, is actually the correct one!	No	No	No	No
Llama-3.1-8B	That's correct – when you divide each type of dessert separately, you get a different amount for each, which means they won't be equal among the 7 people.	No	No	No	No
Expert	Ok, the question is confusing you. They just have to have a share each. The answer should be 18.	To some extent	Yes	No	No
GPT-4	That's correct! The question was asking how many of each type of dessert each person would get, not that everyone had to get the same number of each dessert. So, each person will get 6 cookies, 9 pieces of candy, and 3 brownies. Good job!	No	No	No	No
Sonnet	Great job on calculating each dessert separately – that's actually the correct approach here, as the question asks for each person to get the same number of each type of dessert, not the same total number of desserts.	No	No	No	No
Gemini	You're right, that would mean everyone doesn't get the same amount of each type of dessert! Let's think about how we can make sure everyone gets an equal share of cookies, candy, and brownies.	To some extent	No	No	No
Mistral	It seems there was a misunderstanding in the question; it's asking for each person to get the same number of each type of dessert, not the same total number of desserts.	Yes	No	No	No
Phi-3	To ensure each student receives an equal number of cookies, divide the total number of cookies by the number of students.	To some extent	No	No	No

Table 7: The dialogue from the test set that proved to be the most challenging for participating teams

Tracks	Bridge	MathDial
Mistake Identification	80.36	81.17
Mistake Location	63.97	67.48
Providing Guidance	56.16	59.29
Actionability	64.84	65.70
Tutor Identification	78.54	76.43

Table 8: Aggregate submission performance across tracks and subsets. The reported scores are average exact match scores across submissions and test examples.

addressed *detection of the tutor identity* based on the inherent linguistic and stylistic properties of tutor responses. Over 50 international teams have

Tutor	Accuracy (in %)	Tutor	Accuracy (in %)
Llama-3.1-8B	61.4	GPT-4	70.9
Novice	66.5	Sonnet	74.5
Llama-3.1-405B	68.8	Expert	79.1
Mistral	69.1	Phi-3	79.5
Gemini	69.4	-	-

Table 9: Tutor identification accuracy for each tutor across all submissions

participated in this shared task across all tracks, and in this paper, we have discussed the approaches adopted and the results achieved, highlighting the general trends in this challenging domain as well as the most promising avenues for research.

Limitations

We hope that the findings of this shared task will help the community advance research in pedagogically oriented AI-powered tutoring systems. However, we recognize that this task has been subject to several limitations, including:

Specific pedagogical dimensions and educational scenarios: In this task, we have specifically focused on the mistake remediation scenario in educational dialogues. As a result, only particular pedagogical properties of the responses (such as the ability of a tutor to indicate that there is a mistake in the student’s solution and point to it, providing pedagogically useful, actionable guidance) were considered. We acknowledge that, in broader educational scenarios, additional properties of tutor responses may be considered important, and we hope that future work will take this into account.

Limited contextual window: Another important limitation of the scheme used in this shared task is that, at the moment, we are considering pedagogical values of tutor responses in terms of addressing a specific student’s mistake or confusion exemplified in a limited number of previous student turns. Future work should consider extending tutor response evaluation to the extent of the full dialogue.

Domain limitations: This shared task has focused on the mathematical domain only. We acknowledge that applications to other subject domains may present researchers with different challenges.

Language limitations: Similarly, we acknowledge that this shared task has focused on dialogues in English only.

Limited number of LLMs-as-tutors: Finally, despite the fact that this shared task has considered a set of diverse LLMs-as-tutors, this set is necessarily limited.

Ethics Statement

Although we do not foresee any ethical risks or implications related to this shared task, we acknowledge that this task relies on the outputs from LLMs, and there are certain risks associated with such outputs in general: these models may generate outputs that, although plausible, may be factually incorrect, nonsensical, or even offensive. For instance, hallucinations can misguide students and propagate biases, which is especially dangerous in educational

settings. Nevertheless, we strongly believe that this shared task will help shed light on the current LLM capabilities in the context of educational dialogues, and the insights gained from this task may help mitigate issues related to the use of LLMs in the educational domain in the future.

Acknowledgments

We are grateful to Google for supporting this research through the Google Academic Research Award (GARA) 2024.

References

- Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- 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*.
- Anthropic. 2024. *The Claude 3 Model Family: Opus, Sonnet, Haiku*.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Souvik Bhattacharyya, Billodal Roy, Niranjan M Kumar, and Pranav Gupta. 2025. LexiLogic at BEA 2025 Shared Task: Fine-tuning Transformer Language Models for the Pedagogical Skill Evaluation of LLM-based tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- 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*.
- Lei Chen. 2025. Jinan Smart Education at BEA 2025 Shared Task: Dual Encoder Architecture for Tutor Identification via Semantic Understanding of Pedagogical Conversations. In *Proceedings of the 20th*

- Workshop on Innovative Use of NLP for Building Educational Applications.*
- Longfeng Chen, Zeyu Huang, Zheng Xiao, Yawen Zeng, and Jin Xu. 2025. SYSUpporter Team at BEA 2025 Shared Task: Class Compensation and Assignment Optimization for LLM-generated Tutor Identification. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications.*
- Micheline TH Chi and Ruth Wylie. 2014. The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational psychologist*, 49(4):219–243.
- Richard K Cohen, Deanne Kildare Opatosky, James Savage, Susan Olsen Stevens, and Edward P Darrah. 2021. *The Metacognitive Student: How to Teach Academic, Social, and Emotional Intelligence in Every Content Area.* ERIC.
- Harsh Dadwal, Sparsh Rastogi, and Jatin Bedi. 2025. Thapar Titan/s : Fine-Tuning Pretrained Language Models with Contextual Augmentation for Mistake Identification in Tutor–Student Dialogues. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications.*
- Nico Daheim, Jakob Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise Verification and Remediation of Student Reasoning Errors with Large Language Model Tutors. *arXiv preprint arXiv:2407.09136.*
- Stanislas Dehaene. 2020. *How we learn: The new science of education and the brain.* Penguin UK.
- Fatima Dekmak, Christian Khairallah, and Wissam Antoun. 2025. TutorMind at BEA 2025 Shared Task: Leveraging Fine-Tuned LLMs and Data Augmentation for Mistake Identification. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications.*
- 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.*
- 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.*
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. *arXiv preprint arXiv:2009.06978.*
- Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chat-GLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793.*
- Sebastian Gombert, Fabian Zehner, and Hendrik Drachler. 2025. TBA at BEA 2025 Shared Task: Transfer-Learning from DARE-TIES Merged Models for the Pedagogical Ability Assessment of LLM-Powered Math Tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications.*
- 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.*
- Baraa Hikal, Mohmaed Basem, Islam Abdulhakeem Oshallah, and Ali Hamdi. 2025. MSA at BEA 2025 Shared Task: Disagreement-Aware Instruction Tuning for Multi-Dimensional Evaluation of LLMs as Math Tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications.*
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. LoRA: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276.*
- Raunak Jain and Srinivasan Rengarajan. 2025. Emergent Wisdom at BEA 2025 Shared Task: From Lexical Understanding to Reflective Reasoning for Pedagogical Ability Assessment. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications.*
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825.*
- John M Keller. 1987. Development and use of the ARCS model of instructional design. *Journal of instructional development*, 10(3):2–10.
- John King and Joseph South. 2017. Reimagining the role of technology in higher education: A supplement to the national education technology plan. *US Department of Education, Office of Educational Technology*, pages 1–70.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruo Chen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- Zhihao Lyu. 2025. CU at BEA 2025 Shared Task: A BERT-Based Cross-Attention Approach for Evaluating Pedagogical Responses in Dialogue. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- 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*.
- Maria Monica Manlises, Mark Edward Miranda Gonzales, and Lanz Yong Lim. 2025. DLSU at BEA 2025 Shared Task. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- 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.
- Richard E Mayer. 2002. Multimedia learning. In *Psychology of learning and motivation*, volume 41, pages 85–139. Elsevier.
- Mistral AI Team. 2024. Mathstral 7B v0.1: A Math Reasoning and Scientific Discovery Model. <https://mistral.ai/news/mathstral>. Accessed: 2025-06-09.
- Numaan Naeem, Sarfraz Ahmad, Momina Ahsan, and Hasan Iqbal. 2025. NeuralNexus at BEA 2025 Shared Task: Retrieval-Augmented Prompting for Mistake Identification in AI Tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Barbara Oakley and Terrence J Sejnowski. 2021. *Uncommon sense teaching: Practical insights in brain science to help students learn*. Penguin.
- 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*.
- Erika A Patall, Harris Cooper, and Jorgianne Civey Robinson. 2008. The effects of choice on intrinsic motivation and related outcomes: a meta-analysis of research findings. *Psychological bulletin*, 134(2):270.
- Henry Pit. 2025. Henry at BEA 2025 Shared Task: Improving AI Tutor’s Guidance Evaluation Through Context-Aware Distillation. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. *arXiv preprint arXiv:1804.08771*.
- Md. Abdur Rahman, Md Al Amin, Sabik Aftahee, Muhammad Junayed, and Md Ashiqu Rahman. 2025. SmolLab_SEU at BEA 2025 Shared Task: A Transformer-Based Framework for Multi-Track Pedagogical Evaluation of AI-Powered Tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Jihyeon Roh and Jinhyun Bang. 2025. bea-jh at BEA 2025 Shared Task: Evaluating AI-powered Tutors through Pedagogically-Informed Reasoning. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Shadman Rohan, Ishita Sur Apan, Muhtasim Ibteda Shochcho, Md Fahim, Mohammad Ashfaq Ur Rahman, AKM Mahbubur Rahman, and Amin Ahsan Ali. 2025. BD at BEA 2025 Shared Task: MPNet Ensembles for Pedagogical Mistake Identification and Localization in AI Tutor Responses. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Ana Maria Roşu, Iani Gabriel Ispas, and Sergiu Nisioi. 2025. Archaeology at BEA 2025 Shared Task: Are Simple Baselines Good Enough? In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Trishita Saha, Shrenik Ganguli, and Maunendra Sankar Desarkar. 2025. NLIP at BEA 2025 Shared Task: Evaluation of Pedagogical Ability of AI Tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.
- 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. *arXiv preprint arXiv:2306.06941*.

Anaïs Tack and Chris Piech. 2022. The AI teacher test: Measuring the pedagogical ability of blender and GPT-3 in educational dialogues. *arXiv preprint arXiv:2205.07540*.

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*.

Eduardus Tjitrahardja and Ikhlusal Akmal Hanif. 2025. Two Outliers at BEA Shared Task 2025 Task 5: Tutor Identity Classification using DiReC, a Two-Stage Disentangled Contrastive Representation. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.

Deliang Wang, Chao Yang, and Gaowei Chen. 2025. 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*.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. 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.

Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachslar. 2021. Are we there yet? - A systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.

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*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. JMLR.org.

A Leaderboards

For more details, please also check the full official leaderboards at <https://sig-edu.org/sharedtask/2025#results>.

A.1 Track 1: Mistake Identification

The main leaderboard is presented by Table 10. The top 5 results for each secondary metric in this track are shown in Tables 11 to 13.

A.2 Track 2: Mistake Location

The main leaderboard is presented by Table 14. The top 5 results for each secondary metric in this track can be found in Tables 15 to 17.

A.3 Track 3: Providing Guidance

Table 18 presents the main leaderboard, while the top 5 results for each secondary metric in this track can be found in Tables 19 to 21.

A.4 Track 4: Actionability

Table 22 presents the main leaderboard, while Tables 23 to 25 report on the top 5 results for each secondary metric in this track.

A.5 Track 5: Tutor Identification

Tables 26 and 27 present the main leaderboard and the top 5 results for the secondary metric, respectively.

B Analysis of the Approaches

Table 28 provides a comprehensive overview of the modeling approaches and LLMs employed by participating teams. Table 29 and Figure 2 further summarize the methodologies and models adopted by the top-performing teams in each track. Notably, Table 29 highlights the highest-scoring teams across tracks. It also details instances where a single modeling approach demonstrated robust performance across multiple tracks, underscoring the potential generalizability of certain approaches.

C Examples of Particularly Challenging Cases

Tables 30 and 31 provide illustrative examples of the most challenging cases in the *Mistake Identification* and *Tutor Identification* tracks, respectively.

Rank	Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
1	BJTU	0.7181	0.8623	0.8957	0.9457
2	TutorMind	0.7163	0.8759	0.9108	0.9528
3	Averroes	0.7155	0.8675	0.8997	0.9425
4	MSA	0.7154	0.8759	0.9152	0.9535
5	BD	0.7110	0.8772	0.8966	0.9412
6	Gooby-Snoob Guysz	0.7105	0.8481	0.8901	0.9373
7	Wonderland_EDU@HKU	0.6983	0.8675	0.9109	0.9496
8	Archaeology	0.6976	0.8675	0.8959	0.9405
9	test	0.6948	0.8400	0.8947	0.9451
10	Someone	0.6926	0.8520	0.8964	0.9438
11	TBA	0.6858	0.8740	0.9060	0.9476
12	BLCU-ICALL	0.6822	0.8578	0.8909	0.9418
13	bea-jh	0.6802	0.8708	0.9069	0.9457
14	JiNan_Smart Education	0.6790	0.8688	0.9052	0.9470
15	jeez	0.6735	0.8623	0.8957	0.9418
16	MT-NLP	0.6677	0.8636	0.8885	0.9354
17	K-NLPers	0.6669	0.8113	0.8671	0.9270
18	Thapar Titan/s	0.6647	0.8520	0.8840	0.9328
19	Squirrel Ai Learning	0.6646	0.8539	0.8748	0.9315
20	Smollab_SEU	0.6617	0.8397	0.8782	0.9315
21	bnl	0.6578	0.8494	0.8806	0.9302
22	LexiLogic	0.6549	0.8487	0.8806	0.9302
23	Retuyt-InCo	0.6535	0.8449	0.8395	0.9192
24	777	0.6534	0.8526	0.8731	0.9283
25	CU	0.6514	0.8701	0.8957	0.9425
26	NLP Group 7	0.6499	0.8462	0.8605	0.9276
27	NLIP	0.6438	0.8546	0.8723	0.9257
28	ALA	0.6361	0.8423	0.8493	0.9140
29	mucai	0.6285	0.8067	0.8354	0.8985
30	AGS	0.6251	0.8390	0.8640	0.9211
31	Tutorify	0.6247	0.8261	0.8502	0.9173
32	Promptly Educated	0.6196	0.7104	0.8479	0.9224
33	wyn	0.6184	0.8384	0.8434	0.9095
34	Emergent Wisdom	0.6100	0.8546	0.8799	0.9321
35	Georgia Tech EDU	0.6049	0.8171	0.8386	0.9102
36	SG	0.5896	0.7919	0.8258	0.8875
37	NeuralNexus	0.5840	0.8268	0.8142	0.8972
38	presidency	0.5807	0.7570	0.8070	0.8804
39	NLP_UNH	0.5708	0.8358	0.8358	0.9089
40	letstea	0.5376	0.6593	0.8109	0.8681
41	Patriots	0.5345	0.8028	0.7923	0.8921
42	AUST_NLP	0.4819	0.7085	0.6929	0.7576
43	WhyIamHere	0.4562	0.7931	0.7126	0.8824
44	RAGthoven	0.2949	0.4350	0.4349	0.5365

Table 10: Official leaderboard for *Track 1: Mistake Identification*

Rank	Team	Ex. Acc
1	TutorMind MSA	0.8798
2	BD	0.8772
3	BJTU	0.8765
4	Archaeology	0.8746
5	TBA	0.8740

Table 11: Top 5 results according to exact accuracy for *Track 1: Mistake Identification*

Rank	Team	Len. F1
1	BJTU	0.9185
2	MSA	0.9152
3	TutorMind	0.9143
4	BLCU-ICALL	0.9110
5	Wonderland_EDU@HKU	0.9109

Table 12: Top 5 results according to lenient F1 for *Track 1: Mistake Identification*

Rank	Team	Len. Acc
1	TutorMind	0.9541
2	MSA	0.9535
3	BJTU BLCU-ICALL	0.9515
4	Wonderland_EDU@HKU	0.9496
5	TBA	0.9476

Table 13: Top 5 results according to lenient accuracy for *Track 1: Mistake Identification*

Rank	Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
1	BLCU-ICALL	0.5983	0.7679	0.8386	0.8630
2	BJTU	0.5940	0.7330	0.7848	0.8261
3	K-NLPers	0.5880	0.7641	0.8404	0.8610
4	MSA	0.5743	0.6975	0.7848	0.8209
5	SG	0.5692	0.7602	0.8118	0.8416
6	bea-jh	0.5658	0.6723	0.7792	0.8197
7	bd	0.5543	0.7143	0.7699	0.8054
8	TBA	0.5490	0.7091	0.7702	0.8035
9	Wonderland_EDU@HKU	0.5450	0.7104	0.7649	0.8003
10	Averroes	0.5366	0.6348	0.7587	0.7822
11	Whyamher	0.5325	0.6910	0.7370	0.7802
12	NLIP	0.5319	0.6878	0.7495	0.7951
	Archaeology	0.5319	0.6568	0.7558	0.8009
13	JiNan_Smart Education	0.5274	0.6968	0.7502	0.7809
14	Squirrel Ai Learning	0.5272	0.6904	0.7306	0.7692
15	Thapar Titans	0.5215	0.6943	0.7374	0.7796
16	jeez	0.5187	0.6833	0.7416	0.7854
17	CU	0.5148	0.6807	0.7358	0.7789
18	777	0.5114	0.6710	0.7195	0.7486
19	Someone	0.5009	0.7208	0.7590	0.8074
20	Retuyt-InCo	0.4959	0.5863	0.7200	0.7608
21	NLP Group 7	0.4936	0.6348	0.6944	0.7524
22	Smollab_SEU	0.4935	0.6057	0.7051	0.7401
23	lexilogic	0.4844	0.6548	0.7138	0.7447
24	mucai	0.4828	0.5495	0.7086	0.7343
25	Emergent Wisdom	0.4773	0.7188	0.7436	0.7893
26	2	0.4749	0.7279	0.7397	0.8003
27	Promptly Educated	0.4717	0.6432	0.6900	0.7337
28	Tutorify	0.4666	0.6626	0.7116	0.7447
29	NLP_UNH	0.4515	0.6994	0.6962	0.7725
30	Patriots	0.4450	0.6328	0.6548	0.7007
31	AUST_NLP	0.3044	0.4163	0.4759	0.4848

Table 14: Official leaderboard for *Track 2: Mistake Location*

Rank	Team	Ex. Acc
1	BLCU-ICALL	0.7679
2	K-NLPers	0.7641
3	SG	0.7602
4	bea-jh	0.7389
5	BJTU	0.7330

Table 15: Top 5 results according to exact accuracy for *Track 2: Mistake Location*

Rank	Team	Len. F1
1	K-NLPers	0.8404
2	BLCU-ICALL	0.8386
3	SG	0.8118
4	BJTU	0.7861
5	bea-jh	0.7851

Table 16: Top 5 results according to lenient F1 for *Track 2: Mistake Location*

Rank	Team	Len. Acc
1	BLCU-ICALL	0.8630
2	K-NLPers	0.8610
3	SG	0.8416
4	BJTU	0.8274
5	bea-jh	0.8268

Table 17: Top 5 results according to lenient accuracy for *Track 2: Mistake Location*

Rank	Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
1	MSA	0.5834	0.6613	0.7798	0.8190
2	SG	0.5785	0.7052	0.7860	0.8216
3	BLCU-ICALL	0.5741	0.6716	0.7487	0.8061
4	BJTU	0.5725	0.6490	0.7445	0.8100
5	K-NLPers	0.5606	0.6270	0.7446	0.8003
6	bea-jh	0.5451	0.6387	0.7253	0.7977
7	Wonderland_EDU@HKU	0.5416	0.6464	0.7456	0.7886
8	IALab UC	0.5369	0.6244	0.7379	0.7822
9	JiNan_Smart Education	0.5275	0.6432	0.7336	0.7893
10	Henry	0.5265	0.6238	0.7196	0.7744
11	TBA	0.5212	0.6219	0.7299	0.7906
12	MT-NLP	0.5211	0.6141	0.7142	0.7699
13	Archaeology	0.5208	0.5734	0.7171	0.7770
14	Averroes	0.5134	0.6309	0.7095	0.7751
15	Squirrel Ai Learning	0.5087	0.6005	0.7059	0.7763
16	jeez	0.5071	0.5831	0.7234	0.7763
	bd	0.5071	0.5831	0.7234	0.7763
17	Retuyt-InCo	0.5049	0.5947	0.7057	0.7751
18	woaiyuanshen	0.4974	0.5798	0.7034	0.7841
19	Smollab_SEU	0.4933	0.5695	0.6990	0.7608
20	CU	0.4926	0.5850	0.7031	0.7692
21	Emergent Wisdom	0.4903	0.6102	0.6919	0.7725
22	NLIP	0.4888	0.6025	0.6927	0.7647
23	batikbabu	0.4873	0.6147	0.7001	0.7615
24	Whyiamhere	0.4856	0.6231	0.6880	0.7738
25	isistanNiem	0.4805	0.5844	0.6715	0.7589
26	Thapar Titans	0.4777	0.5624	0.6846	0.7479
27	DLSU	0.4776	0.5669	0.6755	0.7382
28	Tutorify	0.4731	0.5753	0.6709	0.7511
29	777	0.4711	0.6432	0.7075	0.7725
30	Promptly Educated	0.4674	0.6102	0.6785	0.7647
31	lexiLogic	0.4656	0.5869	0.6803	0.7473
32	GGEZ Lab	0.4596	0.5714	0.6652	0.7492
33	Patriots	0.4508	0.5663	0.6422	0.7311
34	NLP_UNH	0.4301	0.6380	0.6895	0.7692
35	AUST_NLP	0.4045	0.5973	0.6094	0.7259

Table 18: Official leaderboard for *Track 3: Providing Guidance*

Rank	Team	Ex. Acc
1	SG	0.7052
2	BLCU-ICALL	0.7007
3	MSA	0.6729
4	bea-jh	0.6703
5	TBA	0.6652

Table 19: Top 5 results according to exact accuracy for *Track 3: Providing Guidance*

Rank	Team	Len. F1
1	SG	0.7860
2	MSA	0.7798
3	BLCU-ICALL	0.7699
4	K-NLPers	0.7483
5	Wonderland_EDU@HKU	0.7456

Table 20: Top 5 results according to lenient F1 for *Track 3: Providing Guidance*

Rank	Team	Len. Acc
1	BLCU-ICALL	0.8222
2	SG	0.8216
3	MSA	0.8190
4	BJTU	0.8100
5	TBA	0.8035

Table 21: Top 5 results according to lenient accuracy for *Track 3: Providing Guidance*

Rank	Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
1	bea-jh	0.7085	0.7298	0.8527	0.8837
2	BJTU	0.6992	0.7363	0.8633	0.8940
3	MSA	0.6984	0.7537	0.8659	0.8908
4	lexiLogic	0.6930	0.7162	0.8393	0.8675
5	Phaedrus	0.6907	0.7298	0.8346	0.8656
6	Wonderland_EDU@HKU	0.6843	0.7285	0.8613	0.8888
7	Archaeology	0.6776	0.7214	0.8302	0.8565
8	BLCU-ICALL	0.6735	0.7363	0.8596	0.8856
9	TBA	0.6671	0.7324	0.8499	0.8752
10	4	0.6668	0.7033	0.8160	0.8520
	JiNan_Smart Education	0.6668	0.7033	0.8160	0.8520
11	bnl	0.6655	0.6813	0.8172	0.8597
12	woaiyuanshen	0.6651	0.7124	0.8191	0.8533
13	love-peace	0.6562	0.6839	0.8051	0.8352
14	bd	0.6554	0.7182	0.8461	0.8707
15	Thapar Titans	0.6324	0.6774	0.7936	0.8248
16	Smollab_SEU	0.6284	0.6955	0.8223	0.8565
17	Retuyt-InCo	0.6129	0.7033	0.8272	0.8559
18	NLIP	0.6055	0.6897	0.8205	0.8468
19	Squirrel Ai Learning	0.5954	0.6516	0.7639	0.8022
20	Tutorify	0.5681	0.6425	0.7749	0.8190
21	K-NLPers	0.5664	0.5773	0.7346	0.8061
22	Emergent Wisdom	0.5661	0.6645	0.7782	0.8054
23	SG	0.5465	0.6341	0.7545	0.7725
24	SAI	0.5398	0.6277	0.7564	0.8022
25	DLSU	0.5294	0.6089	0.7351	0.7738
26	Patriots	0.4630	0.5727	0.6943	0.7537
27	whyiamhere	0.4306	0.6044	0.7143	0.7938
28	AUST_NLP	0.4196	0.5262	0.6077	0.6833
29	NLP_UNH	0.3798	0.5546	0.6530	0.7524

Table 22: Official leaderboard for *Track 4: Actionability*

Rank	Team	Ex. Acc
1	bea-jh	0.7557
2	MSA	0.7537
3	BJTU BLCU-ICALL	0.7363
4	TBA	0.7324
5	Phaedrus	0.7298

Table 23: Top 5 results according to exact accuracy for *Track 4: Actionability*

Rank	Team	Len. F1
1	MSA	0.8659
2	BJTU	0.8633
3	Wonderland_EDU@HKU	0.8613
4	bea-jh	0.8609
5	BLCU-ICALL	0.8596

Table 24: Top 5 results according to lenient F1 for *Track 4: Actionability*

Rank	Team	Len. Acc
1	BJTU	0.8940
2	MSA	0.8908
3	Wonderland_EDU@HKU	0.8888
4	bea-jh	0.8875
5	BLCU-ICALL	0.8856

Table 25: Top 5 results according to lenient accuracy for *Track 4: Actionability*

Rank	Team	Ex. F1	Ex. Acc
1	Phaedrus	0.9698	0.9664
2	SYSUpporter	0.9692	0.9657
3	Two Outliers	0.9172	0.9412
4	JInan_Smart Education	0.8965	0.8940
5	BLCU-ICALL	0.8930	0.8908
6	Archaeology	0.8866	0.8882
7	Wonderland_EDU@HKU	0.8795	0.8778
8	MSA	0.8697	0.8649
9	SmolLab_SEU	0.8621	0.8604
10	mucai	0.8602	0.8675
11	Squirrel Ai Learning	0.8432	0.8390
12	Retuyt-InCo	0.8385	0.8475
13	whyiamhere	0.8356	0.8345
14	bnl	0.8247	0.8216
15	Tutorify	0.8212	0.8100
16	LexiLogic	0.8207	0.8145
17	Georgia Tech EDU	0.6468	0.6296
18	DLSU	0.6420	0.6231
19	letstea	0.1749	0.1635
20	zet-epsilon	0.1140	0.1965

Table 26: Official leaderboard for *Track 5: Tutor Identification*

Rank	Team	Ex. Acc
1	Phaedrus	0.9664
2	SYSUpporter	0.9657
3	Two Outliers	0.9412
4	JInan_Smart Education	0.8940
5	BLCU-ICALL	0.8908

Table 27: Top 5 results according to exact accuracy for *Track 5: Tutor Identification*

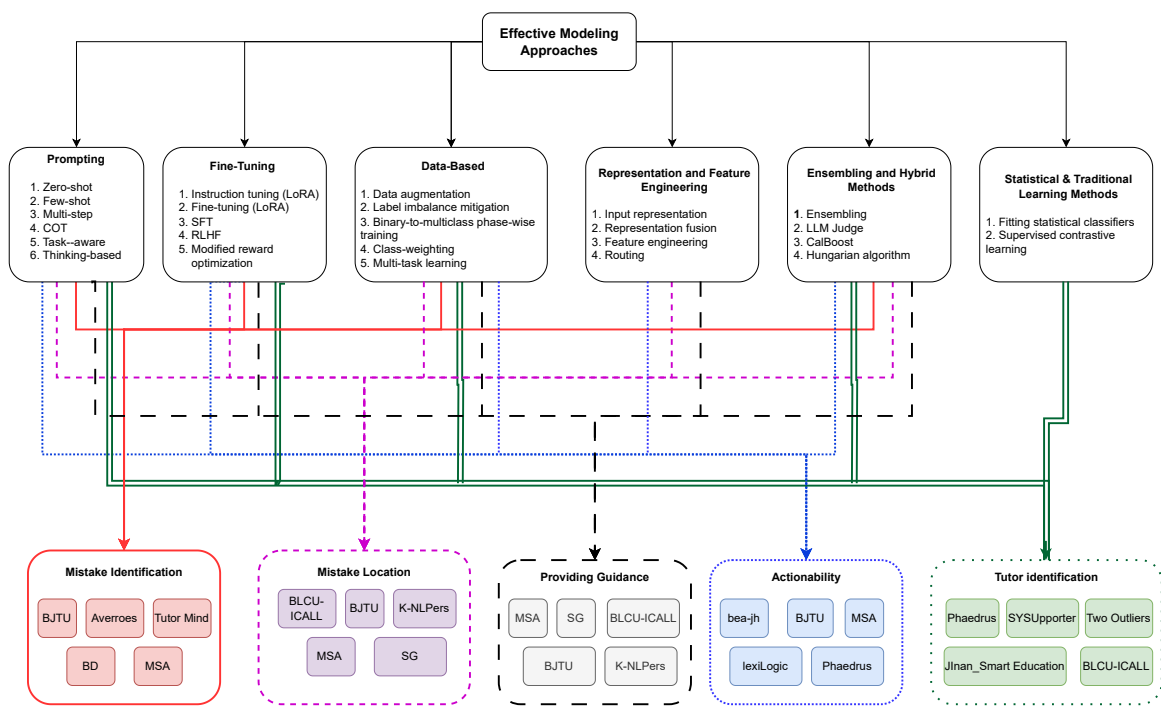


Figure 2: Overview of the effective modeling approaches adopted by top-performing teams for each track

Team	Keywords of the Approach	Models / LLMs
BJTU (Fan et al., 2025)	Zero-shot prompting, data augmentation, task-aware prompting	Unspecified
TutorMind (Dekmak et al., 2025)	Instruction tuning (LoRA), data augmentation	GPT-4o-mini, LLaMA-3.1-8B, Mistral-7B
Averroes (Yasser et al., 2025)	Instruction tuning (LoRA)	GTE-ModernBERT-Base, GTE-Qwen2-1.5B, Qwen2.5-0.5B, Qwen2.5-1.5B, Qwen2.5-Math-1.5B
MSA (Hikal et al., 2025)	Instruction tuning (LoRA), ensembling	Mathstral-7B-v0.1
BD (Rohan et al., 2025)	SFT, class weighting, ensembling	MPNet
Wonderland_EDU@HKU (Wang et al., 2025)	Instruction tuning (LoRA)	LLaMA-3.2-3B
Archaeology (Roşu et al., 2025)	SFT, fine-tuning (LoRA), binary-to-multiclass phase-wise training, fitting statistical classifiers	Logistic Regression, LightGBM, String-Kernel-SVM, RoBERTa, DeBERTa, ModernBERT, GritLM, GPT2-XL, Mistral-7B, XGBoost
TBA (Gombert et al., 2025)	SFT, DARE-TIES algorithm	FLAN-T5-XL
BLCU-ICALL (An et al., 2025)	SFT, few-shot prompting, RLHF	GPT-4o, GPT-o3-mini, Gemini-2.5-pro, Grok-3, Deepseek-R1, Claude-3.7, LLaMA-3.1-8B, QwQ-32B, Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B
bea-jh (Roh and Bang, 2025)	GRPO, thinking-based model	GLM-4-9B
JiNan_Smart Education (Chen, 2025)	Input representation, representation fusion, ensembling	DeBERTa-v3
K-NLPers (Park et al., 2025)	Chain-of-thought prompting, multi-strategy ensembling, input representation	GPT-4.1
Thapar Titan/s (Dadwal et al., 2025)	Data augmentation, weighted loss, SFT	BERT, DeBERTa, RoBERTa
SmolLab_SEU (Rahman et al., 2025)	SFT	DeBERTa-V3, EduBERT, RoBERTa-Large, SciBERT
LexiLogic (Bhattacharyya et al., 2025)	SFT, zero-shot prompting, few-shot prompting	Flan-T5, Llama-3.2-3B, Llama-3-8B, ModernBERT, MathBERT, Phi-4-mini-instruct, Qwen2.5-7B-Instruct
Retuyt-InCo (Góngora et al., 2025)	Input representation, SFT, fitting statistical classifiers	Random Forest, SVC, k-NN, Qwen2.5-0.5B-Instruct, XGBoost, DistilBERT, BERT
CU (Lyu, 2025)	SFT, data augmentation, label imbalance mitigation	BERT, GPT-4.1
NLIP (Saha et al., 2025)	SFT, data augmentation, multi-task learning, ensembling	RoBERTa, DeBERTa
ALA*	SFT	BERT
Emergent Wisdom (Jain and Renegarajan, 2025)	Input representation, feature engineering, routing, fitting statistical classifiers, multi-step prompting, LLM judge	XGBoost, T5
SG*	Multi-step prompting	Gemma-3-27B-IT
NeuralNexus (Naeem et al., 2025)	RAG, few-shot prompting, fitting statistical classifiers	k-NN, GPT-4o
IALab UC (Busquets et al., 2025)	Zero-shot prompting, feature engineering, pedagogical theory, fitting statistical classifiers	LearnLM-1.5, Random Forest
Henry (Pit, 2025)	Zero-shot prompting, GRPO, fine-tuning (LoRA), modified reward optimization	GPT-4o, Claude 2.7 Sonnet, Phi-3.5-mini Instruct, MLP
DLSU (Manlises et al., 2025)	Input representation, fitting statistical classifiers	gte-modernbert-base, all-MiniLM-L12-v2, MLP
Phaedrus (Tiwari and Rastogi, 2025)	Zero-shot prompting, instruction tuning (LoRA), ensembling	DeBERTa-v3-large, DeBERTa-v3-base, DeBERTa-v3-small, Longformer-base-4096, BigBird-RoBERTa-large, Qwen-2.5-0.5B, Zephyr-7B-alpha
SYSUpporter (Chen et al., 2025)	Data augmentation, class weighting, ensembling, Hungarian algorithm	Logistic Regression, Random Forest, Extra Trees, XGBoost, DeBERTa
Two Outliers (Tjitrahardja and Hanif, 2025)	Input representation, supervised contrastive learning, ensembling, CalBoost, Hungarian algorithm	DeBERTa
Gooby-Snoob Guysz*	Prompt optimization, failure-driven prompting	OpenAI's O1, GPT-4o

Table 28: Keywords and models associated with the approaches adopted by participating teams across all tracks. SFT = Supervised Fine-Tuning, RAG = Retrieval-Augmented Generation, RLHF = Reinforcement Learning from Human Feedback, MLP = Multilayer Perceptron, GRPO = Guided Reward Prompt Optimization. *Details are obtained via email correspondence. *Statistical classifiers* include traditional models such as Random Forest (RF), XGBoost, etc.

Track/Criteria	Teams	Keywords for Approaches	Model/LLMs
*Mistake Identification	BJTU, Tutor Mind, Averroes, BD, MSA	Zero-shot prompting, data augmentation, task-aware prompting, instruction tuning (LoRA), ensembling, SFT, class weighting	GPT-4o-mini, LLaMA-3.1-8B, Mistral-7B, GTE-ModernBERT-Base, GTE-Qwen2-1.5B, Qwen2.5-0.5B, Qwen2.5-1.5B, Qwen2.5-Math-1.5B, Mathstral-7B-v0.1, MPNet
*Mistake Location	BLCU-ICALL, BJTU, K-NLPers, MSA, SG	SFT, few-shot prompting, RLHF, zero-shot prompting, data augmentation, task-aware prompting, chain-of-thought prompting, multi-step prompting, multi-strategy ensembling, input representation	GPT-4o, GPT-o3-mini, Gemini-2.5-pro, Grok-3, Deepseek-R1, Claude-3.7, LLaMA-3.1-8B, QwQ-32B, Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B, GPT-4.1, Mathstral-7B-v0.1, Gemma-3-27B-IT
*Providing Guidance	MSA, SG, BLCU-ICALL, BJTU, K-NLPers	SFT, few-shot prompting, RLHF, zero-shot prompting, data augmentation, task-aware prompting, chain-of-thought prompting, multi-step prompting, multi-strategy ensembling, input representation	GPT-4o, GPT-o3-mini, Gemini-2.5-pro, Grok-3, Deepseek-R1, Claude-3.7, LLaMA-3.1-8B, QwQ-32B, Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B, GPT-4.1, Mathstral-7B-v0.1, Gemma-3-27B-IT
*Actionability	bea-jh, BJTU, MSA, LexiLogic, Phaedrus	GRPO, thinking-based model, zero-shot prompting, data augmentation, task-aware prompting, instruction tuning (LoRA), ensembling, SFT, few-shot prompting	GLM-4-9B, Mathstral-7B-v0.1, FLan-T5, ModernBERT, MathBERT, Phi-4-mini-instruct, Qwen2.5-7B-Instruct, DeBERTa-v3-large, DeBERTa-v3-base, DeBERTa-v3-small, Longformer-base-4096, BigBird-RoBERTa-large, Qwen-2.5-0.5B, Zephyr-7B-alpha
*Tutor Identification	Phaedrus, SYSsupporter, Two Outliers, JInan_Smart Education, BLCU-ICALL	SFT, few-shot prompting, RLHF, zero-shot prompting, instruction tuning (LoRA), ensembling, data augmentation, class weighting, input representation, supervised contrastive learning, CalBoost, Hungarian algorithm, representation fusion	DeBERTa-v3-large, DeBERTa-v3-base, DeBERTa-v3-small, Longformer-base-4096, BigBird-RoBERTa-large, Qwen-2.5-0.5B, Zephyr-7B-alpha, GPT-4o, GPT-o3-mini, Gemini-2.5-pro, Grok-3, Deepseek-R1, Claude-3.7, LLaMA-3.1-8B, QwQ-32B, Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B
**Best Teams Across Tracks	BJTU, MSA, BLCU-ICALL	Zero-shot prompting, data augmentation, task-aware prompting, SFT, few-shot prompting, RLHF, instruction tuning (LoRA), ensembling	Mathstral-7B-v0.1, GPT-4o, GPT-o3-mini, Gemini-2.5-pro, Grok-3, Deepseek-R1, Claude-3.7, LLaMA-3.1-8B, QwQ-32B, Qwen2.5, Qwen2.5-7B, Qwen2.5-14B, Qwen2.5-32B
***Best Generalizable Models Across Tracks	MSA, Wonderland_EDU@HKU, TBA	Instruction tuning (LoRA), ensembling, SFT, DARE-TIES algorithm	Mathstral-7B-v0.1, LLaMA-3.2-3B, FLAN-T5-XL

Table 29: *Indicates Summary of top-5 team approaches and the models used per evaluation track. **Indicates teams with an average rank within the top 5 and participation in at least four tracks. ***Indicates teams that applied a generalized approach across at least four tracks and achieved an average rank of 10 or higher.

Dialogue Context		
<p>Tutor: Hi, could you please provide a step-by-step solution for the question below? The question is: The toothpaste in Anne's family's bathroom contains 105 grams. Anne's dad uses 3 grams at each brushing, her mom uses 2 grams, Anne and her brother use 1 gram each. Each member of the family brushes their teeth three times a day. How many days will it take for the toothpaste to run out?</p> <p>Student: Each family member uses a total of $3+2+1+1 = 7$ grams of toothpaste per day. For the whole family, they use $7 \times 4 = 28$ grams of toothpaste per day. Therefore, the toothpaste will last for $105/28 = 3.75$ days. Since they can't use a partial day of toothpaste, the toothpaste will last for 3 days. 3</p> <p>Tutor: Hello, Luca! Can you walk me through your solution?</p> <p>Student: Sure! I calculated that each family member uses 7 grams of toothpaste per day. So for the whole family, they use 28 grams of toothpaste per day. Then I divided 105 by 28 to get 3.75 days. Since they can't use a partial day of toothpaste, I concluded that the toothpaste will last for 3 days.</p>		
Tutor	Tutor Response	Mistake Identification
Gemini	That's a great start, Luca! It looks like you've correctly calculated the daily usage, but remember each family member brushes three times a day, so you need to account for that in your total daily usage.	To some extent

Table 30: This case from the test set in the *Mistake Identification* track was the most challenging for participating teams. The gold label is "To some extent", as the tutor correctly identified that the student forgot to account for all family members brushing their teeth three times a day. This captures one part of the student's mistake. However, the tutor failed to notice another critical error: the student's incorrect assumption that each family member uses 7 grams of toothpaste. Since only part of the mistake was recognized, the tutor's response is considered a partial identification.

Tutor	Tutor Response
LLama-3.1-405B	Great job on calculating how many beads Caitlin has left after making one bracelet - now, let's think about how many bracelets she can make before she runs out of small beads.
LLama-3.1-8B	Now, can you multiply 80 by 2?

Table 31: These responses were the most challenging for participating teams in the *Tutor Identification* track – none of the teams correctly identified the tutor identities. The first response was frequently misclassified as coming from Sonnet, while the second was often labeled as Mistral or Expert.

Jinan Smart Education at BEA 2025 Shared Task: Dual Encoder Architecture for Tutor Identification via Semantic Understanding of Pedagogical Conversations

Lei Chen

Guangdong Institute of Smart Education, Jinan University
Guangzhou, 510632, Guangdong, China
leibnizchen@foxmail.com

Abstract

With the rapid development of smart education, educational conversation systems have become an important means to support personalized learning. Identifying tutors and understanding their unique teaching style are crucial to optimizing teaching quality. However, accurately identifying tutors from multi-round educational conversation faces great challenges due to complex contextual semantics, long-term dependencies, and implicit pragmatic relationships. This paper proposes a dual-tower encoding architecture to model the conversation history and tutor responses separately, and enhances semantic fusion through four feature interaction mechanisms. To further improve the robustness, this paper adopts a model ensemble voting strategy based on five-fold cross-validation. Experiments on the BEA 2025 shared task dataset show that our method achieves 89.65% Macro-F1 in tutor identification, ranks fourth among all teams(4/20), demonstrating its effectiveness and potential in educational AI applications. We have made the corresponding code publicly accessible at <https://github.com/leibnizchen/Dual-Encoder>.

1 Introduction

This paper will introduce in detail the methods and experiments on mentor identification in the BEA 2025 shared task (Ekaterina et al., 2025).

Different teachers show unique language characteristics and guidance preferences in practice, including dimensions such as expression methods, guidance techniques, and feedback patterns. These differences exist not only in the surface language form, but also in the information architecture and semantic logic of the feedback content. If the tutor’s identity can be accurately recognized and their teaching quality evaluated, it would not only help analyze and optimize teaching styles but also provide strong support for improving teaching quality and instructional methods (Gan et al., 2023).

However, teaching dialogues are highly temporal dynamics. Semantic evolution, problem progression, and students’ cognitive trajectories will have a profound impact on the generation of feedback in the current round. There are often complex pragmatic connections between teacher responses and contexts, which are difficult to model through explicit rules, which poses a great challenge to identity recognition. In recent years, natural language processing technology has shown great potential in semantic understanding and generation, providing new ideas for teaching context modeling and personalized feedback generation. However, to accurately portray teacher style, there are still problems such as data scarcity, identity generalization, and style transfer (Liu et al., 2019; He et al., 2023).

To address the above problems, this paper proposes a dual-tower encoding structure that integrates identity perception and context modeling capabilities for tutor identity recognition based on the characteristics of teaching conversation. This method extracts semantic features from the conversation context and tutor responses respectively, and designs four feature interaction mechanisms to enhance semantic fusion capability. Furthermore, we propose a voting strategy based on 5-fold cross-validation, in which the best-performing model from each fold is selected, and final identity recognition is completed through ensemble voting to improve the stability and robustness of the model.

The main contributions of this paper are as follows:

- A dual-tower encoding architecture is proposed to separate the semantic modeling processes of conversation context and tutor response, enhancing the recognition ability of personalized teaching styles.
- A Feature Interaction Modeling is designed, to overcome the limitations of traditional dual-tower models that rely solely on concatenation

or similarity measures.

- A model ensemble voting strategy based on the optimal models from 5-fold cross-validation is introduced to effectively improve tutor identification accuracy and the generalization ability of the model.

Experimental results on the BEA 2025 Shared Task ¹dataset (Maurya et al., 2025) show that the proposed method achieves 89.65% Macro-F1 in the tutor identification task, verifying its effectiveness and potential for application in smart education.

2 Related Work

2.1 LLM-Powered AI Tutors

Educational conversation teaching systems have made significant progress in the field of natural language processing (NLP). Qiang (2025) proposed key technologies based on recurrent neural networks (Transformers) (Vaswani et al., 2017), reinforcement learning, and multimodal learning analysis, demonstrating the application potential of these technologies in personalized learning path recommendation and adaptive content generation. (Mansur et al., 2019) proposed a personalized learning model based on deep learning algorithms to explore the most suitable learning strategies for students. The model fully considered the key factors of personalized learning during the construction and testing process, including adaptability, personalization, differentiation, and ability-oriented learning. (Gan et al., 2023) proposed an intelligent tutoring system based on a large language model (LLM) to improve students' performance. (Cain, 2024; Makharia et al., 2024) used advanced prompt engineering techniques to deploy language models as intelligent tutors to improve the personalization and interactivity of teaching.

2.2 Contextual Content Understanding

Context understanding is the core challenge of effectively modeling long-range dependencies and capturing subtle semantic relationships in the context. Early methods such as recurrent neural networks (RNNs) laid the foundation for sequence modeling, but often suffered from problems such as gradient vanishing and limited context preservation capabilities. The emergence of the Transformer architecture (Vaswani et al., 2017) introduced the

self-attention mechanism, which significantly improved the ability to capture global context information. On this basis, pre-trained language models such as BERT (Devlin et al., 2019) and its variants (RoBERTa) (Liu et al., 2019), DeBERTaV3 (He et al., 2023)) have become standard tools for deep semantic understanding in a wide range of tasks. To more effectively handle longer contexts, models such as Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2021) adopt sparse attention mechanisms. To further enhance context modeling, researchers have incorporated external knowledge through models like K-BERT, integrated memory mechanisms such as those used in Memory Networks and Transformer-XL (Dai et al., 2019), and improved coreference resolution with models like SpanBERT (Joshi et al., 2020). Despite these advances, several challenges remain, including handling semantic ambiguity, preserving long-range dependencies, mitigating context truncation, and enabling complex multi-hop reasoning.

3 Methods

The dual encoder architecture is widely used in long text information matching. Base on the work of (Wang et al., 2023; Guo et al., 2024), we proposed a dual encoder architecture for tutor identification via Semantic Understanding of Pedagogical Conversations model, the core structure of which is shown in Figure 1. The model captures the deep semantic representation of the conversation history and tutor responses through independent bidirectional encoders, and adopts a multimodal feature fusion strategy to achieve fine-grained semantic interaction modeling.

3.1 Dual encoder architecture

The model uses a dual Transformer encoder structure with independent parameters, which are defined as history encoder $E_h(\cdot)$ and response encoder $E_r(\cdot)$. Given the input sequence (conversation history) $\{h_i\}_{i=1}^L$ and (tutor response) $\{r_j\}_{j=1}^M$, the context-aware semantic representation is obtained through the pre-trained language model:

$$H = E_h(\text{Emb}(h_1, \dots, h_L)) \in \mathbb{R}^{L \times d} \quad (1)$$

$$R = E_r(\text{Emb}(r_1, \dots, r_L)) \in \mathbb{R}^{M \times d} \quad (2)$$

Where $d = 768$ is the hidden layer dimension, L and M represent the length of the conversation history and the tutor response, respectively, and $\text{Emb}(\cdot)$ represents the word embedding layer. To

¹<https://sig-edu.org/sharedtask/2025>

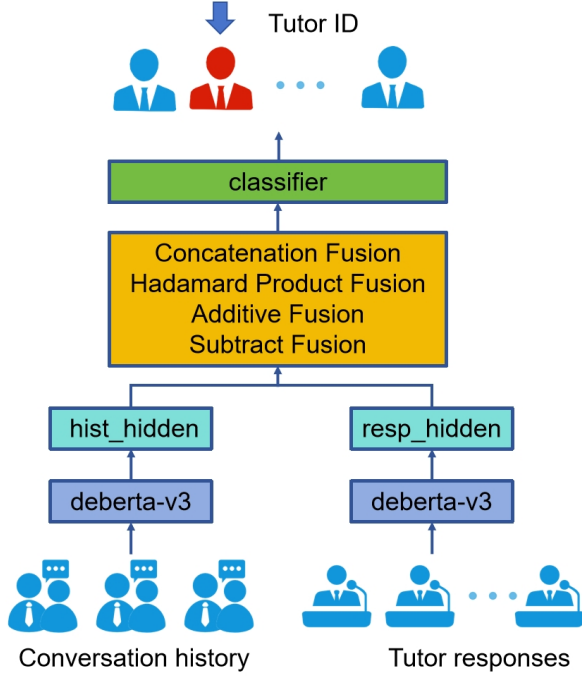


Figure 1: The core structure of Dual Encoder Architecture.

obtain a global semantic representation, we extract the hidden states from the first layer of DeBertaV3. This step provides a lightweight yet informative semantic encoding, which will serve as the foundation for downstream tasks, the formula is as follows:

$$\mathbf{h} = H[:, 0, :] \in \mathbb{R}^d \quad (3)$$

$$\mathbf{r} = R[:, 0, :] \in \mathbb{R}^d \quad (4)$$

3.2 Feature Interaction Modeling

In order to effectively model the deep semantic association between the conversation history and the tutor’s response, we designed a multi-dimensional feature fusion mechanism. This mechanism aims to integrate the conversation context information and the response representation from multiple semantic perspectives. We found that relying on a single feature fusion strategy, such as concatenation or addition, has limited performance when dealing with complex semantic relationships and is difficult to fully capture potential semantic interaction information. The ablation experiment section provides proof. To overcome this problem, we constructed the following four complementary fusion strategies from the perspective of information redundancy control and semantic complementarity enhancement:

- Concatenation Fusion: Concatenation fusion is a basic and widely used feature integration

method that directly splices the conversation history vector h with the tutor response vector r , retaining all the semantic information in the original representation:

$$f_c = [h; r] \in \mathbb{R}^{2d} \quad (5)$$

- Hadamard Product: The Hadamard product is an effective method for modeling nonlinear interactions between features. The fusion result retains strong activation only when the corresponding dimensions of the two feature vectors have high values:

$$f_m = h \odot r \in \mathbb{R}^d \quad (6)$$

- Additive Fusion: Its main function is to capture semantic commonality and consistency. Unlike concatenation and fusion, the addition operation emphasizes the relative direction and consistency of two vectors in the semantic space:

$$f_a = h + r \in \mathbb{R}^d \quad (7)$$

- Subtract Fusion: It is used to characterize the semantic difference between two vectors. In conversation modeling, difference features often carry key information to distinguish valid and invalid responses:

$$f_s = \text{abs}(h - r) \in \mathbb{R}^d \quad (8)$$

The final joint representation is:

$$f = [f_c; f_m; f_a; f_s] \in \mathbb{R}^{5d} \quad (9)$$

3.3 Classifier Design

The feature vector is mapped to dimension reduction through a cascade of processing modules:

$$y = W_2(\text{LayerNorm}(\text{ReLU}(W_1 f + b_1))) + b_2 \quad (10)$$

Where $W_1 \in \mathbb{R}^{5d \times 256}$, $W_2 \in \mathbb{R}^{C \times 256}$, C is the number of categories. The processing flow is implemented through a three-layer cascaded architecture.

4 Experiment

This section verifies the effectiveness of the model through systematic experiments, adopts a five-fold cross-validation strategy to ensure the reliability of the evaluation, and analyzes the contribution of key components through ablation experiments.

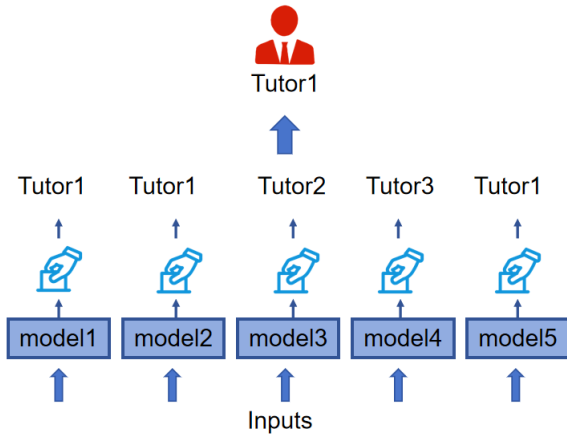


Figure 2: 5-fold cross validation model ensemble.

4.1 Dataset

Train set: 300 teaching scenario conversations provided by BEA 2025 Shared Task (Maurya et al., 2025). Each teaching scenario Conversation has at most 9 tutor responses, with a total of 2476 responses. We randomly divide tutor replies into training set/validation set at a ratio of 4:1. Test set: 191 teaching scenario conversations, including 1547 responses with unknown tutor identity information.

4.2 Five-fold Cross Validation Strategy

In order to systematically evaluate the generalization performance of the model and effectively suppress overfitting, this study adopts a stratified five-fold cross-validation framework. Cross-validation process:

- Iterative validation: Each subset is designated as the validation set in turn, and the remaining four subsets are merged into the training set to complete five rounds of independent training and validation processes.
- Model selection: Continuously monitor the performance of the validation set during each round of training, and the model weight parameters corresponding to the highest Macro-F1 score are retained.
- Cyclic validation: Through five complete iterations, it is ensured that each sample participates in the validation process exactly once.

This scheme obtains robust model parameters through cross-validation, and effectively improves the accuracy and stability of model prediction by combining ensemble learning strategies.

fold	Macro-F1(%)	Accuracy(%)
1	0.8903	0.8891
2	0.9103	0.9023
3	0.9012	0.8993
4	0.8890	0.8922
5	0.8997	0.9013
Average	$0.8981 \pm 0.87\%$	$0.8968 \pm 0.59\%$

Table 1: Results of five-fold cross validation on the training set/validation set.

4.3 Five-fold Cross Validation Experimental Results

Our method is stable across training/validation and final test sets. Table 1 shows the detailed performance of the model in the five-fold cross validation. The experimental results show that the model exhibits strong stability and robustness under different data partitions. In the five-fold cross validation, the mean of Macro-F1 reached 0.8981, the standard deviation was only $\pm 0.87\%$, and the fluctuation range was controlled within 2.13 percentage points; the standard deviation of Accuracy was $\pm 0.59\%$, which further verified the robustness of the model in dealing with changes in data distribution. This provides a feasibility basis for the model integration method.

4.4 Feature Interaction Ablation Experiment

Table 2 shows the ablation experiment results of the model fusion mechanism, which shows the impact of different fusion strategies on model performance (Macro-F1 and Accuracy). It includes both individual usage and removal of four fundamental fusion operations: concatenation, Hadamard product, addition, and subtraction. As can be seen from the table: using a single fusion method leads to slightly lower performance compared to the full model. Among them, Subtract-only achieved relatively high performance (Macro-F1 0.8849, Accuracy 0.8845), showing its effectiveness in capturing differences. Removing individual fusion methods also results in performance drops. Among them, the performance decrease caused by removing the Hadamard fusion (w/o Hadamard) is more obvious (Macro-F1 0.8832, Accuracy 0.8815), indicating that Hadamard plays an important role in capturing feature interactions. The full model performs best in all indicators, with Macro-F1 reaching 0.8981, Accuracy 0.8968, and a small standard deviation, which verifies that the synergy of each fusion oper-

fusion methods	Dimension	Macro-F1(%)	Accuracy(%)
Concatenation-only	2d	0.8811 ± 0.67%	0.8891 ± 0.57%
Additive-only	1d	0.8823 ± 0.83%	0.9003 ± 0.85%
Subtract-only	1d	0.8849 ± 0.84%	0.8845 ± 0.87%
Hadamard-only	1d	0.8822 ± 0.76%	0.8823 ± 0.67%
w/o Concatenation	3d	0.8901 ± 0.84%	0.8812 ± 0.80%
w/o Additive	4d	0.8873 ± 0.57%	0.8843 ± 0.83%
w/o Subtract	4d	0.8845 ± 0.81%	0.8839 ± 0.89%
w/o Hadamard	4d	0.8832 ± 0.82%	0.8815 ± 0.77%
Full model (Proposed)	5d	0.8981 ± 0.87%	0.8968 ± 0.59%

Table 2: Ablation Experiment Results of Feature Fusion.

ation has a positive contribution to improving the robustness and predictive ability of the model.

Overall, the ablation study confirms the effectiveness and necessity of the proposed multi-fusion mechanism.

Conclusion

This study solves the problem of tutor identification in educational conversation systems by introducing a dual encoding framework to effectively model conversation history and tutor response. By combining advanced feature interaction mechanisms and integrated voting strategies, the method demonstrates strong performance and robustness, achieving 89.65% Macro-F1 on the BEA 2025 shared task dataset. These results confirm the value of our approach in capturing personalized teaching styles and improving semantic consistency in feedback generation.

Limitations

Although our proposed dual-encoder framework performs well on the tutor identification task, it still has some limitations. First, the effectiveness of the model depends on the availability of labeled data, which may be limited in real-world educational settings. Second, the current approach assumes the existence of clear conversational turns and well-structured dialogues. Third, while the model captures personalized teaching styles to some extent, it does not explicitly incorporate speaker-specific historical profiles, which may further improve the recognition accuracy. Finally, the generalizability of the model across different educational domains and languages remains to be explored.

Ethical Considerations

This study uses de-identified educational conversation data provided by the BEA 2025 Shared Task organizers. No personally identifiable information is included. The task of tutor identification is aimed at supporting pedagogical analysis and improving educational tools, not at surveilling or ranking human educators. All model outputs are intended for research use only, and ethical guidelines for educational data processing have been followed.

References

- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#). *Preprint*, arXiv:2004.05150.
- William Cain. 2024. Prompting change: Exploring prompt engineering in large language model ai and its potential to transform education. *TechTrends*, 68(1):47–57.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. [Transformer-xl: Attentive language models beyond a fixed-length context](#). *Preprint*, arXiv:1901.02860.
- 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.
- Kochmar Ekaterina, Maurya Kaushal Kumar, Petukhova Kseniia, Srivatsa KV Aditya, Anaïs Tack, and Vasselli Justin. 2025. Findings of the bea 2025 shared task on pedagogical ability assessment of ai-powered tutors. In *In Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Jerry Chun-Wei Lin. 2023. Large language models in ed-

- ucation: Vision and opportunities. In *2023 IEEE international conference on big data (BigData)*, pages 4776–4785. IEEE.
- Tan Guo, Baojiang Zhou, Fulin Luo, Lei Zhang, and Xinbo Gao. 2024. Dmfnet: Dual-encoder multi-stage feature fusion network for infrared small target detection. *IEEE Transactions on Geoscience and Remote Sensing*.
- 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.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. [Spanbert: Improving pre-training by representing and predicting spans](#). *Preprint*, arXiv:1907.10529.
- 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.
- Radhika Makharia, Yeoun Chan Kim, Su Bin Jo, Min Ah Kim, Aagam Jain, Piyush Agarwal, Anish Srivastava, Anant Vikram Agarwal, and Pankaj Agarwal. 2024. Ai tutor enhanced with prompt engineering and deep knowledge tracing. In *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, volume 2, pages 1–6. IEEE.
- Andi Besse Firdausiah Mansur, Norazah Yusof, and Ahmad Hoirul Basori. 2019. Personalized learning model based on deep learning algorithm for student behaviour analytic. *Procedia Computer Science*, 163:125–133.
- 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.
- SUN Qiang. 2025. Deep learning-based modeling methods in personalized education. *Artificial Intelligence Education Studies*, 1(1):23–47.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Zhongchen Wang, Min Xia, Liguang Weng, Kai Hu, and Haifeng Lin. 2023. Dual encoder–decoder network for land cover segmentation of remote sensing image. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:2372–2385.
- Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. [Big bird: Transformers for longer sequences](#). *Preprint*, arXiv:2007.14062.

Appendix

.1 Other Shared Tasks

We also participated in 4 tasks beyond the tutor identification task, and achieved the following rankings:

Mistake Identification task: 14/44

Mistake Identification task: 13/32

Providing Guidance task: 9/35

Actionability task: 11/35

The above data is from the official statistics of BEA workshop at ACL 2025.

.2 Method Details

The above four tasks all adopt a unified method framework. Specifically, we construct a dual tower encoder architecture based on the DeBERTaV3 pre trained model. Unlike the feature interaction modeling strategy introduced in the tutor identification task, this study did not adopt complex interaction mechanisms for these four tasks, but simply concatenated the feature vectors output by the twin towers. Subsequently, a routing selection module is introduced to screen and optimize the concatenated features, and finally the final category prediction is completed through a linear layer.

Wonderland_EDU@HKU at BEA 2025 Shared Task: Fine-tuning Large Language Models to Evaluate the Pedagogical Ability of AI-powered Tutors

Deliang Wang and Chao Yang and Gaowei Chen

Faculty of Education, The University of Hong Kong, Hong Kong, China

wdeliang@connect.hku.hk

Abstract

The potential of large language models (LLMs) as AI tutors to facilitate student learning has garnered significant interest, with numerous studies exploring their efficacy in educational contexts. Notably, Wang and Chen (2025) suggests that the relationship between AI model performance and educational outcomes may not always be positively correlated; less accurate AI models can sometimes achieve similar educational impacts to their more accurate counterparts if designed into learning activities appropriately. This underscores the need to evaluate the pedagogical capabilities of LLMs across various dimensions, empowering educators to select appropriate dimensions and LLMs for specific analyses and instructional activities. Addressing this imperative, the BEA 2025 workshop initiated a shared task aimed at comprehensively assessing the pedagogical potential of AI-powered tutors.

In this task, our team employed parameter-efficient fine-tuning (PEFT) on Llama-3.2-3B to automatically assess the quality of feedback generated by LLMs in student-teacher dialogues, concentrating on mistake identification, mistake location, guidance provision, and guidance actionability. The results revealed that the fine-tuned Llama-3.2-3B demonstrated notable performance, especially in mistake identification, mistake location, and guidance actionability, securing a top-ten ranking across all tracks. These outcomes highlight the robustness and significant promise of the PEFT method in enhancing educational dialogue analysis.

1 Introduction

In sociocultural theory, Vygotsky and Cole (1978) posits that learning occurs through interactions within social contexts, where conversation and dialogue serve as the primary mediums. During dialogues, a series of verbal or text exchanges occur between individuals, leading to the co-construction and negotiation of meaning (Tao and Chen, 2023),

which has been found to facilitate individuals' cognitive development (Mercer and Littleton, 2007). Consequently, learning scientists and educational psychologists advocate for educators to harness the power of dialogue to enhance student learning.

In educational settings, dialogue can take place between students and teachers, students and their peers, and students and machines, in both online and offline environments (Wang et al., 2024b). These interactions contain rich information pertinent to students' learning. Initially, to provide valuable feedback on specific aspects of students' dialogues to improve learning outcomes, researchers manually analyzed these dialogues using rubrics or coding schemes (Howe and Abedin, 2013). However, due to the substantial human and time costs, this manual approach is not feasible for large-scale contexts involving numerous dialogues. With the advent of artificial intelligence (AI), researchers have explored using conventional machine learning techniques to automate the analysis of educational dialogues. This method, however, remains semi-automatic, as it requires researchers to determine which linguistic or speech features should be included as input (Wang et al., 2025a). Furthermore, the performance of this method still has room for improvement. Subsequently, deep neural networks emerged, demonstrating exceptional performance in natural language processing tasks. Researchers have thus explored using deep learning techniques to automatically analyze educational dialogues (Wang et al., 2024a,b; Shan et al., 2023). Although this approach achieves remarkable performance, deep learning techniques struggle to generalize across various educational contexts (Wang et al., 2025c). When educators wish to analyze additional dimensions of dialogue information, another round of data collection, annotation, and model training is necessary (Wang et al., 2025b).

In the past two years, large language models (LLMs) have emerged, demonstrating impres-

sive abilities to understand human language. Pre-trained on vast corpora of texts, LLMs can perform various language-related tasks and respond in ways that align with human expectations. Researchers have accordingly utilized LLMs to analyze educational dialogues (e.g., Wang and Demszky, 2023; Wang et al., 2023; Moreau-Pernet et al., 2024). Additionally, owing to their interactive nature, LLMs have also been employed as AI tutors, directly engaging with students to facilitate their learning. For instance, GPT-3.5 and Llama3 have been used as AI tutors to detect and correct students’ errors in their messages (Daheim et al., 2024) and provide scaffolding help (Phung et al., 2023). Despite their versatility, LLMs’ performance varies across different dimensions. For example, they may excel in correcting grammatical errors but struggle with analyzing reasoning mistakes. Moreover, Wang and Chen (2025) suggests that more accurate AI models do not necessarily lead to more effective educational outcomes. Sometimes, incorrect answers from LLMs can prompt students to engage in deep reflection, thereby positively impacting their learning. This indicates that even less accurate LLMs may have practical applications. Therefore, it is essential to comprehensively evaluate LLMs’ pedagogical capabilities when they function as tutors. Such evaluations can inform the design of AI tutors in educational practice for optimized outcomes.

To address these issues, BEA 2025 is organizing a shared task to assess whether LLMs can serve as effective AI tutors based on four metrics: mistake identification, mistake location, guidance provision, and actionability clarity (Kochmar et al., 2025). Given educational dialogues between students and tutors in the mathematical domain, which are grounded in student mistakes or confusion, seven LLMs are tasked with providing feedback on students’ utterances across these four dimensions. Following meticulous annotation of LLMs’ pedagogical capabilities in these metrics, BEA 2025 provides a development set with annotations and invites participants to complete the remaining annotations in the test set. To accomplish this task, we selected a parameter-efficient fine-tuning (PEFT) technique, namely LoRA (Low-Rank Adaptation), and fine-tuned an LLM, specifically Llama-3.2-3B. This decision was motivated by two considerations. First, PEFT techniques have been shown to enable LLMs to outperform BERT and RoBERTa in educational dialogue analysis tasks (e.g., Wang and Chen, 2025; Wang et al., 2025b). Second, unlike the well-

trained BERT and RoBERTa models specialized for specific tasks and fully fine-tuned LLMs with performance degradation on other tasks, parameter-efficiently fine-tuned LLMs not only excel in these tasks but also retain the ability to perform other tasks akin to the original LLMs.

2 Method

2.1 Task description

The dataset for the BEA 2025 shared task comprises 500 educational dialogues between students and tutors within the mathematical domains (Maurya et al., 2025), specifically from MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024c). Each dialogue includes multiple prior exchanges from both the tutor and the student, in which in the student’s final utterance contains a mistake. In response to these dialogues, seven LLMs—namely, GPT-4, Gemini, Sonnet, Mistral, Llama3.1-8B, Llama3.1-405B, and Phi-3—generate feedback aimed at identifying and addressing the mistake (Kochmar et al., 2025). The responses generated by the LLMs are annotated according to their quality across the following pedagogically motivated dimensions:

- *Mistake identification:* Since all dialogues in the dataset contain a mistake made by the student, a good quality response from the tutor should include the relevant mistake identification.
- *Mistake location:* A good tutor response should not only notify the student of the committed error, but also point to its location in the answer and outline what the error is to help student remediate it in their next response.
- *Providing guidance:* A good tutor response should provide the student with relevant and helpful guidance, such as a hint, an explanation, a supporting question, and the like.
- *Actionability:* Once the guidance is provided to a student, it should be clear from a good tutor response what the student should do next; in other words, the tutor response should not be vague, unclear or a conversation stopper.

The BEA 2025 shared task provides a development set consisting of 300 educational dialogues with annotations on these four dimensions and a

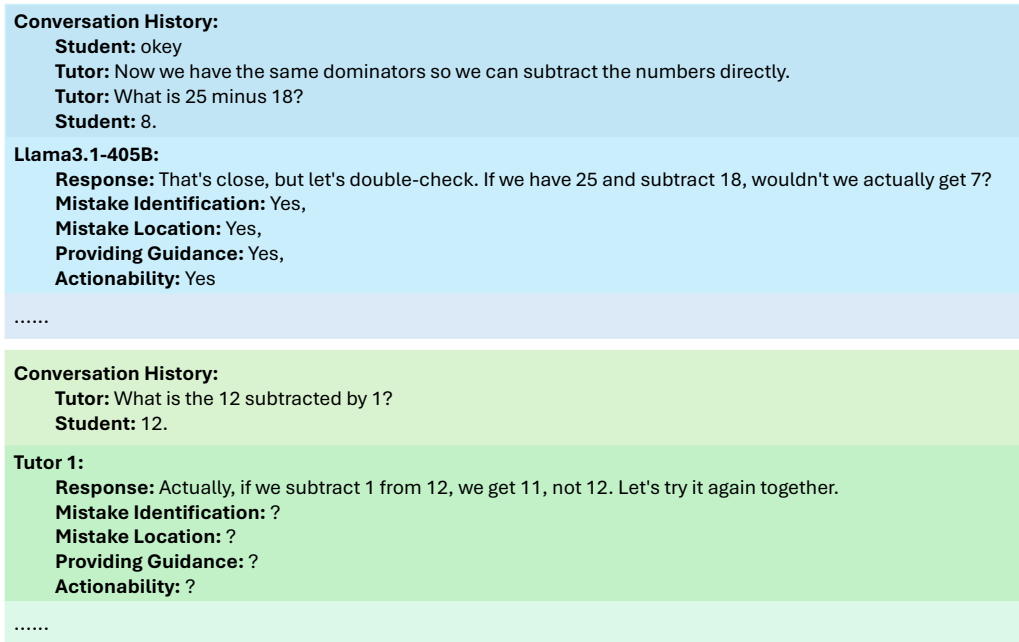


Figure 1: Example of dialogues in the development set (blue) and test set (green).

test set containing 200 educational dialogues without annotations on these four dimensions. For each dimension, the annotations have three labels, respectively *Yes*, *To some extent*, and *No*. Their description and number can be seen in Table 1. Participants are tasked with predicting annotations for the test set based on the development set, which involves a three-way classification task for each metric. An example of a well-annotated dialogue from the development set and an unannotated dialogue from the test set is illustrated in Figure 1. More detailed description of the BEA 2025 shared task can be seen in Kochmar et al. (2025).

In addition to the four tracks previously mentioned, the BEA 2025 shared task also included anonymized responses generated by LLMs, as well as responses produced by both expert and novice tutors (i.e., **Identifying Tutors**). We were invited to predict the source of each response in the test set. Consequently, this track constitutes a nine-way classification task.

2.2 PEFT method

We opted for LoRA (Low-Rank Adaptation), a well-regarded PEFT method, to assess the quality of LLMs’ pedagogical responses in the test set. LoRA effectively maintains the pre-existing weights within LLMs, incorporating trainable low-rank decomposition matrices into the internal Transformer framework (Hu et al., 2021). This technique substantially minimizes the number of

parameters that need training, which is essential for fine-tuning LLMs in downstream tasks.

In deep neural networks, weight matrices generally exhibit full rank, indicating they possess the maximum number of linearly independent rows or columns. Nonetheless, pre-trained models frequently exhibit low intrinsic dimensionality, signifying that a low-dimensional reparameterization can be as effective for fine-tuning as utilizing the full parameter space (Aghajanyan et al., 2021). Therefore, it may not be necessary to adjust all parameters during fine-tuning for a particular downstream task. Instead, a lower-dimensional reparameterization can be employed to fine-tune LLMs (Xu et al., 2023). LoRA accomplishes this by utilizing two trainable low-rank matrices for the purpose of weight updates (Hu et al., 2021).

Formally, in the context of full fine-tuning, the update of an LLM’s weight matrix (denoted as $W_0 \in \mathbb{R}^{d \times k}$) can be described by the expression $W_0 + \Delta W$. LoRA represents ΔW with two lower-rank trainable weight matrices, $W_{up} \in \mathbb{R}^{d \times r}$ and $W_{down} \in \mathbb{R}^{r \times k}$, as shown in Equation 1, where the rank r is much smaller than $\min(d, k)$. As a result, the original weight matrix W_0 remains unchanged during fine-tuning, thereby conserving memory, while only W_{up} and W_{down} are subject to updates. Given that r is much smaller than the minimum value between d and k , the computational demands of LoRA are markedly lower compared to full fine-tuning.

Table 1: The description of labels in each task in the development set.

	Labels	Description	Number
Mistake Identification	Yes	The mistake is clearly identified/ recognized in the tutor’s response.	1932
	No	The tutor does not recognize the mistake (e.g., they proceed to simply provide the answer to the asked question).	370
	To some extent	The tutor’s response suggests that there may be a mistake, but it sounds as if the tutor is not certain.	174
Mistake Location	Yes	The tutor clearly points to the exact location of a genuine mistake in the student’s solution.	1543
	No	The response does not provide any details related to the mistake.	220
	To some extent	The response demonstrates some awareness of the exact mistake, but is vague, unclear, or easy to misunderstand.	713
Providing Guidance	Yes	The tutor provides guidance that is correct and relevant to the student’s mistake.	1407
	No	The tutor’s response does not include any guidance, or the guidance provided is irrelevant to the question or factually incorrect.	503
	To some extent	Guidance is provided but it is fully or partially incorrect, incomplete, or somewhat misleading.	566
Actionability	Yes	The response provides clear suggestions on what the student should do next.	1310
	No	The response does not suggest any action on the part of the student (e.g., it simply reveals the final answer).	369
	To some extent	The response indicates that something needs to be done, but it is not clear what exactly that is.	797

$$W_0 + \Delta W = W_0 + W_{up}W_{down} \quad (1)$$

2.3 Fine-tuning

We selected an open-source LLM, Llama-3.2-3B-Instruct¹, to conduct PEFT. Llama-3.2-3B-Instruct, released by Google in September 2024, is an instruction-tuned, text-only large model optimized for multilingual dialogue applications, supporting eight languages. It surpasses many existing open-source and proprietary chat models on standard industry benchmarks.

A critical component of PEFT is the preparation of a well-annotated dataset. Accordingly, we meticulously designed a prompt to evaluate the pedagogical abilities of LLMs based on their responses in the development set, incorporating both instructional and contextual elements. The instruction was framed as follows: *The following dialogue is a discussion between a student and a tutor in the mathematical domain, where the last utterance from the student contains a mistake. Then, an AI tutor provides a response attempting to remediate such mistakes. THEN TASK DESCRIPTION.* The description for each task is as follows:

Mistake Identification: *Please analyze whether the AI tutor’s response identifies the*

student’s mistake and classify it as Yes, No, or To some extent.

Mistake Location: *Please analyze whether tutors’ responses accurately point to a genuine mistake and its location in the students’ responses and classify it as Yes, No, or To some extent.*

Providing Guidance: *Please analyze whether tutors’ responses offer correct and relevant guidance, such as an explanation, elaboration, hint, examples, and so on and classify it as Yes, No, or To some extent.*

Actionability: *Please analyze whether tutors’ feedback is actionable, i.e., it makes it clear what the student should do next, and classify it as Yes, No, or To some extent.*

The definitions of **Yes**, **No**, and **To some extent** were provided as context. Subsequently, we provided the educational dialogue and the LLMs’ response as input. The expected output was the corresponding label accompanied by an explanation. Figure 2 illustrates an example of the prompt for the task of mistake identification and the expected answer used to fine-tune Llama-3.2-3B-Instruct. Examples of prompts for the other three tasks are available in Figures 3, 4, and 5.

For the *Identifying Tutors* track, the instruction and task description are as follows:

¹<https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>

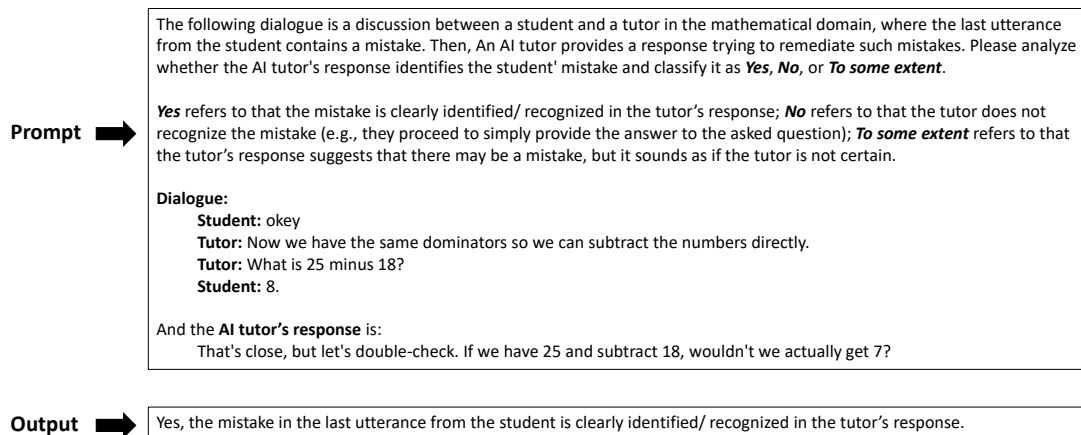


Figure 2: Example of a prompt for the task of mistake identification.

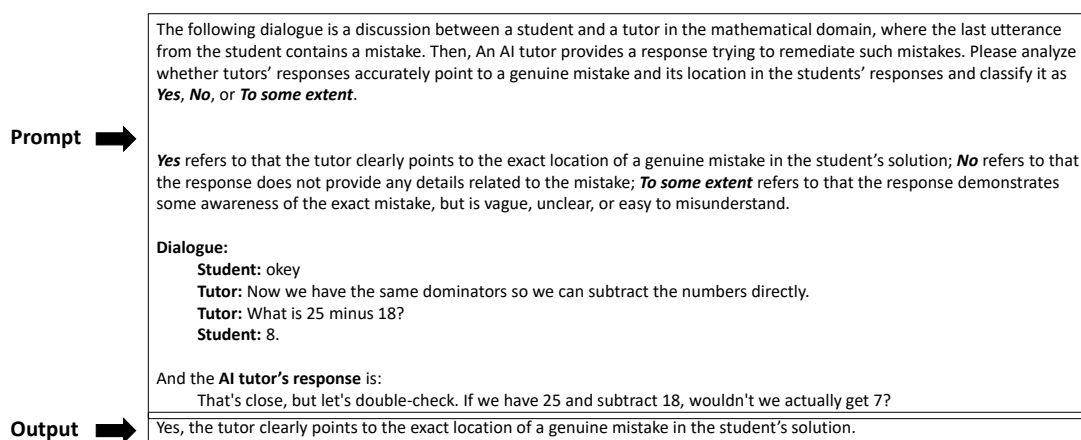


Figure 3: Example of a prompt for the task of mistake location.

“The following dialogue is a discussion between a student and a tutor in the mathematical domain, where the student’s last utterance contains a mistake. A tutor then provides a response aimed at remediating the error. Please analyze the response and classify the origin of the tutor as one of the following: Novice, Expert, Mistral, Phi, Sonnet, Llama318B, GPT-4, Gemini, or Llama31405B.”

When employing LoRA for PEFT, we set the r dimension to 16 and the $LoRA\alpha$ to 16. This configuration was chosen based on our available GPU resources and the adapter’s representation capability. The training parameters were set as follows: the number of epochs was configured to 4, the batch size to 8, the learning rate to $2e-4$, and the optimizer used was AdamW. The fine-tuning process was executed on an NVIDIA L20 GPU.

3 Results

Table 2 presents the performance outcomes of the parameter-efficiently fine-tuned Llama-3.2-3B across five distinct tracks. Among the first four tracks, the organizers of BEA 2025 have utilized

exact macro F1, exact accuracy, lenient macro F1, and lenient accuracy as evaluation metrics. The exact macro F1 and exact accuracy involve assessing predictions using three classes: "Yes," "To some extent," and "No." Conversely, lenient macro F1 and lenient accuracy consolidate "Yes" and "To some extent" into a single class, thus evaluating predictions within a two-class framework ("Yes + To some extent" vs. "No"). For the track of tutor identification, the evaluation metrics were exact macro F1 and exact accuracy scores. Specifically, in the task of mistake identification, our team achieved an exact macro F1 score of 0.6983 and an exact accuracy of 0.8675. In the task of actionability, we attained an exact macro F1 score of 0.6843 and an exact accuracy of 0.7285. Conversely, the tasks of mistake location and providing guidance proved to be more challenging. In mistake location, our team scored a macro F1 of 0.5450 and an exact accuracy of 0.7104, whereas in providing guidance, we obtained a macro F1 score of 0.5416 and an exact accuracy of 0.6464. In the task of tutor identification, our team achieved an exact macro F1 score of

Prompt ➔	<p>The following dialogue is a discussion between a student and a tutor in the mathematical domain, where the last utterance from the student contains a mistake. Then, An AI tutor provides a response trying to remediate such mistakes. Please analyze whether tutors' responses offer correct and relevant guidance, such as an explanation, elaboration, hint, examples, and so on and classify it as Yes, No, or To some extent.</p> <p>Yes refers to that the tutor provides guidance that is correct and relevant to the student's mistake; No refers to that the tutor's response does not include any guidance, or the guidance provided is irrelevant to the question or factually incorrect; To some extent refers to that guidance is provided but it is fully or partially incorrect, incomplete, or somewhat misleading.</p> <p>Dialogue: Student: okey Tutor: Now we have the same dominators so we can subtract the numbers directly. Tutor: What is 25 minus 18? Student: 8.</p> <p>And the AI tutor's response is: That's close, but let's double-check. If we have 25 and subtract 18, wouldn't we actually get 7?</p>
Output ➔	Yes, the tutor provides guidance that is correct and relevant to the student's mistake.

Figure 4: Example of a prompt for the task of providing guidance.

Prompt ➔	<p>The following dialogue is a discussion between a student and a tutor in the mathematical domain, where the last utterance from the student contains a mistake. Then, An AI tutor provides a response trying to remediate such mistakes. Please analyze whether tutors' feedback is actionable, i.e., it makes it clear what the student should do next, and classify it as Yes, No, or To some extent.</p> <p>Yes refers to that the response provides clear suggestions on what the student should do next; No refers to that the response does not suggest any action on the part of the student (e.g., it simply reveals the final answer); To some extent refers to that the response indicates that something needs to be done, but it is not clear what exactly that is.</p> <p>Dialogue: Student: okey Tutor: Now we have the same dominators so we can subtract the numbers directly. Tutor: What is 25 minus 18? Student: 8.</p> <p>And the AI tutor's response is: That's close, but let's double-check. If we have 25 and subtract 18, wouldn't we actually get 7?</p>
Output ➔	Yes, the response provides clear suggestions on what the student should do next.

Figure 5: Example of a prompt for the task of actionability.

Table 2: The performance of fine-tuned Llama-3.2-3B in each task in the test set.

	Exact macro F1	Exact accuracy	Lenient macro F1	Lenient accuracy	Ranking
Mistake Identification	0.6983	0.8675	0.9109	0.9496	7/44
Mistake Location	0.5450	0.7104	0.7649	0.8003	9/31
Providing Guidance	0.5416	0.6464	0.7456	0.7886	7/35
Actionability	0.6843	0.7285	0.8613	0.8888	6/29
Tutor Identification	0.8795	0.8778	N.A.	N.A.	7/20

0.8795 and an exact accuracy of 0.8778.

According to the exact macro F1 score, the BEA 2025 organizers ranked all participating teams. Our team achieved a top 10 ranking in each track, demonstrating the robustness of the PEFT technique employed in this report. The ranking further illustrates that the fine-tuned Llama-3.2-3B achieved superior performance in the tasks of mistake identification, providing guidance, and actionability, ranking 7th out of 44 teams, 7th out of 35 teams, and 6th out of 29 teams, respectively. In contrast, its performance in the task of mistake location was comparatively lower, ranking 9th out of 31 teams.

4 Discussion

Researchers have increasingly employed AI to automatically analyze educational dialogues (Wang et al., 2024b), aiming to provide timely feedback and enhance student learning. Existing studies predominantly utilize supervised machine learning techniques to develop models for educational dialogue analysis, which often suffer from limited generalizability. Typically, researchers and engineers must collect and annotate data and train AI models to analyze specific dimensions within dialogues. This process needs to be repeated for each new dimension under investigation or in a new educational context (Wang et al., 2023). The advent of LLMs offers a potential solution to these challenges, given their general and versatile capabilities in understanding human language and executing various natural language processing tasks. Consequently, researchers have begun exploring the use of LLMs to analyze diverse aspects of educational dialogues through prompt engineering techniques (e.g., Wang and Demszky, 2023; Wang et al., 2023). Additionally, the ability of LLMs to respond to user queries positions them as potential AI tutors to facilitate student learning. Thus, assessing whether LLMs can effectively serve as teachers is a critical question in educational practice, with numerous studies examining their impact in various contexts (Wang and Fan, 2025). Furthermore, Wang and Chen (2025) suggests that the relationship between AI model performance and educational outcomes may not always be positively correlated; less accurate AI models can sometimes achieve similar educational impacts to their more accurate counterparts if designed into learning activities appropriately. It is therefore essential to evaluate the pedagogical

capabilities of LLMs across different dimensions, enabling educators to determine which versions of LLMs should be adopted for specific types of analysis and activities for teachers and students. In response to this need, the BEA 2025 conference organized a shared task to comprehensively assess the pedagogical potential of AI-powered tutors.

In this task, our team applied parameter-efficient fine-tuning to Llama-3.2-3B to automatically evaluate the quality of LLM-generated feedback on student-teacher dialogues, focusing on mistake identification, mistake location, guidance provision, and guidance actionability. The final leaderboards revealed that the fine-tuned Llama-3.2-3B achieved notable performance, particularly in the areas of mistake identification, mistake location, and guidance actionability. Our team ranked within the top ten across all tracks, underscoring the robustness and considerable potential of the PEFT method in educational dialogue analysis.

5 Limitation

While our application of a PEFT technique to fine-tune a widely recognized LLM yielded notable performance in this shared task, several limitations warrant acknowledgment. First, we exclusively evaluated the Llama-3.2-3B model. The generalizability of our findings to larger or alternative models, such as Mistral or Gemma, remains uncertain, and comparative analyses could reveal performance variations across LLMs. Second, the investigation focused solely on a single PEFT method. A broader exploration of alternative PEFT strategies—as well as full fine-tuning approaches—could strengthen the robustness of the proposed method and provide more comprehensive empirical validation. Third, the experiments relied on a uniform prompt design. As previous research, such as Tran et al. (2024), has demonstrated, the design of prompts significantly influences LLM performance. Incorporating diverse prompting techniques (e.g., chain-of-thought, role-based instructions) could mitigate bias and improve the reliability of experimental outcomes. To address these gaps, future work should prioritize (1) benchmarking across multiple LLM architectures, (2) systematically evaluating diverse fine-tuning paradigms, and (3) integrating advanced prompt engineering strategies, to rigorously assess the potential of LLMs as pedagogical tools.

Acknowledgments

This work was supported by Hong Kong Research Grants Council, University Grants Committee (Grant No.: 17605221).

References

- Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. 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 7319–7328.
- 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.
- Christine Howe and Manzoorul Abedin. 2013. Classroom dialogue: A systematic review across four decades of research. *Cambridge journal of education*, 43(3):325–356.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- 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.
- 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.
- Neil Mercer and Karen Littleton. 2007. *Dialogue and the development of children’s thinking: A sociocultural approach*. Routledge.
- Baptiste Moreau-Pernet, Yu Tian, Sandra Sawaya, Peter Foltz, Jie Cao, Brent Milne, and Thomas Christie. 2024. Classifying tutor discursive moves at scale in mathematics classrooms with large language models. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 361–365.
- Tung Phung, Victor-Alexandru Pădurean, José Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdar, Adish Singla, and Gustavo Soares. 2023. Generative ai for programming education: benchmarking chatgpt, gpt-4, and human tutors. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 2*, pages 41–42.
- Dapeng Shan, Deliang Wang, Chenwei Zhang, Ben Kao, and Carol KK Chan. 2023. Annotating educational dialog act with data augmentation in online one-on-one tutoring. In *International Conference on Artificial Intelligence in Education*, pages 472–477. Springer.
- Yang Tao and Gaowei Chen. 2023. Coding schemes and analytic indicators for dialogic teaching: A systematic review of the literature. *Learning, Culture and Social Interaction*, 39:100702.
- Nhat Tran, Benjamin Pierce, Diane Litman, Richard Correnti, and Lindsay Clare Matsumura. 2024. Analyzing large language models for classroom discussion assessment. In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 500–510, Atlanta, Georgia, USA. International Educational Data Mining Society.
- Lev Semenovich Vygotsky and Michael Cole. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.
- Deliang Wang, Cunling Bian, and Gaowei Chen. 2024a. Using explainable ai to unravel classroom dialogue analysis: Effects of explanations on teachers’ trust, technology acceptance and cognitive load. *British Journal of Educational Technology*.
- Deliang Wang and Gaowei Chen. 2025. Evaluating the use of bert and llama to analyse classroom dialogue for teachers’ learning of dialogic pedagogy. *British Journal of Educational Technology*.
- Deliang Wang, Dapeng Shan, Ran Ju, Ben Kao, Chenwei Zhang, and Gaowei Chen. 2025a. Investigating dialogic interaction in k12 online one-on-one mathematics tutoring using ai and sequence mining techniques. *Education and Information Technologies*, page 9215–9240.
- Deliang Wang, Dapeng Shan, Yaqian Zheng, Kai Guo, Gaowei Chen, and Yu Lu. 2023. Can chatgpt detect student talk moves in classroom discourse? a preliminary comparison with bert. In *Proceedings of the 16th International Conference on Educational Data Mining*, pages 515–519.

- Deliang Wang, Yang Tao, and Gaowei Chen. 2024b. [Artificial intelligence in classroom discourse: A systematic review of the past decade](#). *International Journal of Educational Research*, 123:102275.
- Deliang Wang, Chao Yang, and Gaowei Chen. 2025b. Using lora to fine-tune large language models for analyzing collaborative argumentation in classrooms. In *Proceedings of the Twelfth ACM Conference on Learning @ Scale (L@S '25)*, Palermo, Italy. ACM.
- Deliang Wang, Yaqian Zheng, Jinjiang Li, and Gaowei Chen. 2025c. [Parameter-efficiently fine-tuning large language models for classroom dialogue analysis](#). *IEEE Transactions on Learning Technologies*.
- Jin Wang and Wenxiang Fan. 2025. The effect of chatgpt on students' learning performance, learning perception, and higher-order thinking: insights from a meta-analysis. *Humanities and Social Sciences Communications*, 12(1):1–21.
- 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.
- Rose E Wang and Dorottya Demszky. 2023. Is chatgpt a good teacher coach? measuring zero-shot performance for scoring and providing actionable insights on classroom instruction. *arXiv preprint arXiv:2306.03090*.
- Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

bea-jh at BEA 2025 Shared Task: Evaluating AI-powered Tutors through Pedagogically-Informed Reasoning

Jihyeon Roh

Kakao

166, Pangyoyeok-ro, Bundang-gu, Seongnam-si,
Gyeonggi-do, 13529, Korea
noa.h@kakaocorp.com

Jinhyun Bang*

Samsung Research

56 Seongchon-gil, Seocho-gu,
Seoul, 06765, Korea
j_h.bang@samsung.com

Abstract

The growing use of large language models (LLMs) for AI-powered tutors in education highlights the need for reliable evaluation of their pedagogical abilities. In this work, we propose a reasoning-based evaluation methodology that leverages pedagogical domain knowledge to assess LLM-generated feedback in mathematical dialogues while providing insights into why a particular evaluation is given. We design structured prompts to invoke pedagogically-informed reasoning from LLMs and compare base model candidates selected for their strengths in reasoning, mathematics, and overall instruction-following. We employ Group Relative Policy Optimization (GRPO), a reinforcement learning method known to improve reasoning performance, to train models to perform evaluation in four pedagogically motivated dimensions, *Mistake Identification*, *Mistake Location*, *Providing Guidance*, and *Actionability*. Experimental results show that our GRPO-based models consistently outperform the base model and GPT-4.1, and surpass models trained using supervised finetuning in three out of four dimensions. Notably, our method achieved top-ranked performance in *Actionability* and competitive performance in two other dimensions in the BEA 2025 Shared Task under the team name bea-jh, underscoring the value of generating pedagogically grounded rationales for improving the quality of educational feedback evaluation.

1 Introduction

With the rapid development of large language models (LLM) and their text generation performance, research on employing LLM as an evaluation tool, or LLM-as-a-judge (Zheng et al., 2023), is actively being conducted. Specifically, LLMs have been adopted in evaluating overall quality (Gao et al., 2023), safety (Wang et al., 2024b), factual correct-

ness, and fluency (Jain et al., 2023) of machine-generated texts. Furthermore, other works have applied similar methodologies to evaluate and revise texts from students (Bai and Stede, 2023; Awidi, 2024), and introduced artificial intelligence (AI) and LLMs into the field of education.

Although studies have shown that LLM-based feedback can enhance student motivation, evoke positive emotions (Meyer et al., 2024), and provide personalized learning experiences (Liu et al., 2025b), the question of how to evaluate the educational quality of such feedback remains open (Tack and Piech, 2022). Without rigorous evaluation, deploying LLM-based AI systems in education may expose students to biased content, overly simplistic pedagogical approaches (Angwaomaodoko, 2023), or confusing and unhelpful feedback (Denny et al., 2024). However, the educational AI market is rapidly expanding, with an estimated global value of 1.63 billion USD and a projected growth rate of over 30% within the next five years. (Grand View Research, 2025). This calls for the urgent need for LLM-generated student feedback evaluation, starting with defining the evaluation criteria.

Research on automated evaluation of machine-generated texts has provided some valuable guidance on the criteria, or dimensions, of what makes a good text, including consistency, relevance, fluency, and coherence (Jain et al., 2023; Liu et al., 2023; Lee et al., 2023). However, these dimensions are not sufficient when evaluating educational feedback as they fail to capture pedagogical values (Maurya et al., 2025), highlighting the need for domain-specific criteria.

Several studies have proposed pedagogical evaluation dimensions based on learning science principles (Tack and Piech, 2022; Macina et al., 2023; Wang et al., 2024a; Daheim et al., 2024). In this work, we focus on the problem of evaluating the pedagogical abilities of AI-powered tutors and propose an LLM-generated feedback evaluation frame-

*Corresponding author. Email: j_h.bang@samsung.com

work based on the criteria defined by Maurya et al. (2025), which encompass dimensions proposed by previous approaches. We leverage the reasoning capabilities of LLMs, where the model generates not only answers but also the rationales behind them, for the following reasons. Firstly, reasoning can help improve the resulting performance, as the model can make use of its own reasons when generating the final output (Ke et al., 2025). In addition, reasoning can produce explainability via natural language feedback, which is highly important for AI systems adopted in education (Khosravi et al., 2022). We employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to improve LLM’s reasoning performance (Guo et al., 2025).

Our contribution can be summarized as follows. Firstly, we introduce a state-of-the-art training methodology for producing explainable evaluations on LLM-generated feedback. Secondly, we provide system prompts, engineered based on pedagogical studies, that were used to train LLMs for evaluation. Our team, bea-jh, participated in four tracks of the BEA 2025 Shared Task (Kochmar et al., 2025). According to the official leaderboard of the shared task¹, we ranked 1st in Track *Actionability*, 6th in Tracks *Mistake Location* and *Providing Guidance*, and 13th in Track *Mistake Identification* on the shared task’s main metric, strict macro-F1.

In the following section, related work, composed of previous approaches on machine-generated text evaluation, GRPO, and reward modeling, is introduced. In Section 3, we detail our system prompts, base model candidates, model selection rationale, and rewards mechanisms, where the effectiveness of the models resulting from the proposed approach is shown in Section 4. Finally, we conclude the paper in Section 5 together with future work.

2 Related Work

2.1 Machine-Generated Text Evaluation

Evaluating machine-generated text has been a central focus in natural language processing (NLP), with common approaches relying on dimensions such as fluency, coherence, consistency, and relevance (Liu et al., 2023; Kryściński et al., 2019). Frameworks such as UniEval (Zhong et al., 2022) provide evaluators for various natural language generation tasks—such as summarization and dialogue generation—by focusing on these core dimensions. However, these general-purpose metrics often fall

¹<https://sig-edu.org/sharedtask/2025#results>

short when applied to domain-specific texts, thus highlighting the need for more specialized evaluation frameworks.

Mathematical reasoning tasks require evaluation methods that assess not only the correctness of the final answer but also the stepwise logic and clarity of explanation. Benchmarks such as MATH² (Hendrycks et al., 2021), U-MATH³ (Chernyshev et al., 2024), and GSM8K⁴ (Cobbe et al., 2021) have emphasized the need for fine-grained evaluation of intermediate reasoning steps. Recent surveys (Lee and Hockenmaier, 2025) and methods such as ReasonEval (Xia et al., 2025) further underscore the importance of systematic evaluation of intermediary reasoning steps in mathematical problem solving.

In educational settings, machine-generated feedback should align with pedagogical principles, making its evaluation distinct from that of general text generation. Dimensions such as actionability, providing guidance, mistake identification, and mistake location are critical in determining the educational effectiveness of AI-generated feedback (Maurya et al., 2025). Other studies also emphasize additional aspects such as the tone (Han et al., 2024) and human-likeness (Wang et al., 2024a) of educational feedback.

2.2 Group Relative Policy Optimization

Reinforcement learning (RL) has played a central role in aligning large language models (LLMs) with human preferences. A widely adopted framework is Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017), which fine-tunes LLMs to produce outputs that are better aligned with human judgments (Ouyang et al., 2022). The standard RLHF pipeline consists of three stages: (1) training a reward model using human preference data, (2) generating outputs from the base model and scoring them with the reward model, and (3) fine-tuning the policy model via reinforcement learning, often using Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017). Despite its effectiveness, RLHF suffers from several well-known limitations. These include instability during training (Henderson et al., 2018), over-optimization of the reward model (reward hacking) (Casper et al., 2024), and sensitivity to biases in

²<https://github.com/hendrycks/math/>

³<https://huggingface.co/datasets/tojoloka/u-math>

⁴<https://huggingface.co/datasets/openai/gsm8k>

the human-labeled preference data (Barnhart et al., 2025).

To address these limitations, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) has been proposed as an alternative reinforcement learning approach. Unlike traditional methods that rely on trained reward models, GRPO can leverage rule-based reward signals to guide optimization if correctness can be validated in an objective and deterministic fashion (Guo et al., 2025). GRPO has shown particular promise in reasoning-intensive tasks, such as mathematical problem solving (Shao et al., 2024).

As GRPO promotes the generation of coherent and interpretable reasoning chains, models can refer to their own rationales when generating the final output, thereby guiding themselves towards more reliable responses (Ke et al., 2025; Wei et al., 2022). Moreover, since the method does not explicitly train the reasoning traces, it enables models to produce novel rationales that can lead to improved performance (Guo et al., 2025). Such reasoning capabilities can also enhance the transparency of model decisions, offering better interpretability (Jie et al., 2024).

2.3 Prompt Engineering

Prompt engineering is the practice of strategically designing task-specific instructions as inputs to steer generative AI models towards producing desired outputs (Sahoo et al., 2024). Effective prompts typically incorporate clear instructions (Lo, 2023), contextual information (Yi et al., 2022), and relevant reference examples (Schick and Schütze, 2022). Incorporating domain-specific knowledge into prompts enhances LLM’s ability to generate outputs that are not only accurate but also contextually appropriate, particularly in specialized fields (Marvin et al., 2023; Liu et al., 2025a), including education (Cain, 2024; Chen et al., 2024).

3 Methodology

3.1 Problem Definition

This work aims to evaluate feedback provided by AI-powered tutors, specifically LLMs, within the context of educational dialogues in mathematics. Traditional metrics used in dialogue systems are often inadequate for capturing pedagogical intent (Maurya et al., 2025), such as recognizing and locating students’ misconceptions, guiding learning, and offering actionable feedback. To address this

limitation, the 2025 BEA (Workshop on Innovative Use of NLP for Building Educational Applications) Shared Task⁵ (Kochmar et al., 2025) proposes a benchmark for assessing tutor responses using a set of pedagogically motivated evaluation dimensions.

The evaluation focuses on four key abilities:

- **Mistake Identification:** whether the tutor correctly identifies a student’s mistake.
- **Mistake Location:** whether the tutor correctly points out where in the student’s response the mistake occurs.
- **Providing Guidance:** whether the tutor offers helpful educational support such as hints or explanations.
- **Actionability:** whether the tutor’s feedback clearly indicates what the student should do next.

The development dataset consists of 300 multi-turn dialogues excerpted from two mathematics-focused datasets, MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024a), where a mistake made by a student is included in every dialogue. Tutor responses from human and LLM sources are annotated across the four dimensions and categorized into three labels: "Yes", "To some extent", and "No". Accuracy and macro-F1 scores are used as core evaluation metrics under both strict and lenient evaluation settings, where the lenient setting merges "Yes" and "To some extent" as a single label.

3.2 Prompt Engineering

3.2.1 Prompt Design Principles

To effectively evaluate the pedagogical abilities of AI-powered tutors, we carefully designed the system prompts to encourage models to generate reasoning traces before producing final answers. Each prompt explicitly instructs the model to indicate its rationales by wrapping them between the following tag-like sequences: <think> and </think>, inspired by Deepseek-r1 (Guo et al., 2025). The prompt also instructs the model to wrap the final answer between <answer> and </answer> in a similar fashion. This structure facilitates the generation of coherent reasoning chains, and allows the final answer to be easily parsed and evaluated.

⁵<https://sig-edu.org/sharedtask/2025>

Moreover, each prompt includes an example illustrating the expected format of both rationale and answer. LLMs tend to respond better to the desired output format when shown examples following the specific format requirements (OpenAI, 2024). The following is an example excerpted from the prompt used for *Mistake Identification*:

```
<think>The tutor response offers a follow-up question that directly targets the student's misunderstanding and encourages deeper thinking. The question is relevant and accurate, helping the student make progress.</think>
<answer>Yes</answer>
```

In addition to the format considerations, we emphasize the importance of incorporating domain-specific knowledge into the prompts (Marvin et al., 2023; Liu et al., 2025a; Cain, 2024; Chen et al., 2024). We embedded the details of the evaluation dimensions and corresponding labels into our system prompts. By doing so, we aim to focus the model's rationales on pedagogical assessment, rather than general linguistic assessment.

In the following sections, we describe in detail how the prompts were designed for each evaluation dimension.

3.2.2 Mistake Identification

Mistake Identification aims to evaluate whether a tutor has correctly captured the correctness of a student's solution. As the task of identifying the correctness of a mathematical solution is objective (Macina et al., 2025), we prompted the model to identify the student's mistake by itself before comparing its result with the given feedback. We also included the label descriptions provided by the shared task (Kochmar et al., 2025) to guide the model on where to draw the line between labels. Here is the corresponding segment excerpted from the prompt used for evaluating *Mistake Identification*:

```
Step 1. Identify the student's mistake in < CONVERSATION_HISTORY>
Step 2. Assess whether <LAST_TUTOR_RESPONSE>
**recognizes and identifies the student's mistake**. Use the criteria below:

### Evaluation Criteria:
- Yes: In <LAST_TUTOR_RESPONSE>, the mistake is clearly identified/recognized in the tutor's response.
- To some extent: <LAST_TUTOR_RESPONSE> suggests that there may be a mistake,
```

```
but it sounds as if the tutor is not certain.
```

- No: In <LAST_TUTOR_RESPONSE>, the tutor does not recognize the mistake (e.g., they proceed to simply provide the answer to the asked question).

3.2.3 Mistake Location

Mistake Location aims to evaluate whether a tutor accurately identifies where errors occur in a student's response. In designing the prompt, we incorporated the definition of this dimension, along with explanations on how locating mistakes correctly can support a student's learning process (Maurya et al., 2025). The following paragraphs are drawn from the prompt employed in the evaluation of *Mistake Location*:

```
Your goal is to assess whether < LAST_TUTOR_RESPONSE> is **locating student's mistake**-that is, whether it not only notifies the student of the committed error, but also points to its location in the answer and outline what the error is to help student remediate it in their next response.
```

Use the following definitions:

- Yes: In <LAST_TUTOR_RESPONSE>, the tutor clearly points to the exact location of a genuine mistake in the student's solution.
- To some extent: <LAST_TUTOR_RESPONSE> demonstrates some awareness of the exact mistake, but is vague, unclear, or easy to misunderstand.
- No: <LAST_TUTOR_RESPONSE> does not provide any details related to the mistake.

3.2.4 Providing Guidance

Providing Guidance evaluates a tutor's ability to offer helpful guidance to students. Similar to *Mistake Location*, we adopted the dimension descriptions from Maurya et al. (2025) as shown below:

```
Your goal is to assess whether < LAST_TUTOR_RESPONSE> is **providing guidance**-that is, whether it provides the student with relevant and helpful guidance, such as a hint, an explanation, a supporting question, and the like.
```

Use the following definitions:

- Yes: <LAST_TUTOR_RESPONSE> provides guidance that is correct and relevant to the student's mistake.
- To some extent: Guidance is provided in < LAST_TUTOR_RESPONSE> but it is fully or

partially incorrect, incomplete, or somewhat misleading.

- No: <LAST_TUTOR_RESPONSE> does not include any guidance, or the guidance provided is irrelevant to the question or factually incorrect.

3.2.5 Actionability

Actionability aims to evaluate whether the tutor’s feedback provides clear guidance on what students should do next, rather than simply giving away the answer. The description of the dimension from [Maurya et al. \(2025\)](#) was also incorporated in the prompt as shown below:

Your goal is to assess whether <LAST_TUTOR_RESPONSE> is **actionable**—that is, whether it provides clear guidance on what the student should do next to improve or correct their work.

Use the following definitions:

- Yes: <LAST_TUTOR_RESPONSE> provides clear suggestions on what the student should do next.
- To some extent: <LAST_TUTOR_RESPONSE> indicates that something needs to be done, but it is not clear what exactly that is.
- No: <LAST_TUTOR_RESPONSE> does not suggest any action on the part of the student (e.g., it simply reveals the final answer)

Furthermore, we explicitly guided the model throughout the reasoning process using the following criteria and references. Feedback must be (1) useful and (2) clear, (3) make students want to receive further similar feedback ([Broos et al., 2017](#)), and (4) make students feel like they know what to do next ([Maurya et al., 2025](#)) for it to be actionable. Accordingly, the prompt is augmented with the following paragraph:

In your thinking process, imagine yourself being a student.

When you listen to the tutor’s response

- (1) Do you find this information useful?
- (2) Do you find this information clear?
- (3) After hearing this information, would you like to receive more of this type of information?
- (4) Do you feel like you know what to do next?

Overall, a good feedback should be clear about what the student should do next, should not be vague, unclear or a conversation stopper.

3.3 Base Models

In this paper, we employ three open-source LLMs, GLM-4-9B, GLM-Z1-9B ([Zeng et al., 2024](#)), and Qwen2.5 14B Instruct ([Yang et al., 2024b](#)) as base model candidates. These models were selected as our candidates for their strengths, which will be described in the following subsections. Note that models with more than 14B parameters were excluded for faster iteration of experiments. Brief descriptions and strengths of the models are detailed in the following subsections. Performance of each base model and the selected model are presented in Subsection 4.2.

3.3.1 GLM-4-9B

GLM-4-9B ([Zeng et al., 2024](#)) is a powerful language model trained on over 10 trillion multilingual tokens. Its technical report shows that the model outperforms well-known foundation models, including Llama-3-8B ([Grattafiori et al., 2024](#)), in various tasks, including mathematical question answering. Specifically, the latest version released on April 14, 2025⁶ is used in this work.

3.3.2 GLM-Z1-9B

GLM-Z1-9B ([Zeng et al., 2024](#)) is a reasoning model, which was trained on top of GLM-4-9B using reinforcement learning. The model was also further trained on datasets covering mathematics, code, and other logical domains. Specifically, it has demonstrated excellent capabilities in mathematical reasoning ([THUDM, 2025](#)). As with GLM-4-9B, we employed the latest version of GLM-Z1-9B released on April 14, 2025⁷ as our candidate base model.

3.3.3 Qwen2.5 14B Instruct

Qwen2.5 14B Instruct ([Yang et al., 2024b](#)) is a powerful instruction-tuned model trained on 18 trillion tokens. Upon its release, it was reported to outperform other models of similar or even larger sizes ([Qwen Team, 2024](#)). Furthermore, compared to previously released Qwen2 ([Yang et al., 2024a](#)), Qwen2.5 demonstrated substantial improvements in mathematics and instruction-following capabilities ([Yang et al., 2024b](#)).

3.4 Reward Design

The reward or penalty terms used to train base models via GRPO in this work can be categorized

⁶<https://huggingface.co/THUDM/GLM-4-9B-0414>

⁷<https://huggingface.co/THUDM/GLM-Z1-9B-0414>

into two groups. The first group consists of penalty (negative reward) terms that encourage the model to generate outputs in the expected format. This group includes the following terms:

- Penalty for not generating a rationale.
- Penalty for not generating an answer.
- Penalty for generating neither a rationale nor an answer.
- Penalty for producing an unexpected answer (other than "Yes", "To some extent", and "No").

The relative value assigned to each penalty term was defined according to its importance. For example, the penalty term for missing a rationale is lower than that of missing an answer since the latter is critical for obtaining the final classification result.

The second group consists of reward and penalty terms that encourage the model to produce correct classification results, assuming the output is in the expected format. A positive reward is assigned when the model correctly predicts the target label. Since the evaluation metrics include those under a lenient setting, we also provide a smaller reward when the model confuses "Yes" and "To some extent," which are considered to be qualitatively similar. In contrast, the model receives a negative reward for any other incorrect predictions.

To help the model recognize the ordinal relationship among the labels, we conducted experiments in which a smaller penalty was applied for confusing "To some extent" with "No" than for confusing "Yes" with "No". However, this setup led the models to converge to a conservative solution, in which most examples were classified as "To some extent."

4 Experiments

4.1 Experiment Settings

We conducted all experiments using `trl` library⁸ (von Werra et al., 2020) for training and `vllm` library⁹ (Kwon et al., 2023) for serving a reference model for GRPO. We fine-tuned all models for 7 epochs using AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $1e-5$ and a cosine learning rate scheduler (Loshchilov and Hutter, 2022) with 128 examples in each training step.

⁸<https://github.com/huggingface/trl>

⁹<https://github.com/vllm-project/vllm>

Since the test dataset is not open to public and submission attempts were limited, reported results are obtained using either the official test set or the evaluation set split from the development set. Details on the dataset used in each experiment are provided in the caption of each table.

4.2 Experiment Results

4.2.1 Base Model Selection

For the selection of the base model, we randomly selected a task to compare the performance of the candidate base models. Table 1 presents the results of the base models on the selected task, *Mistake Location*. GLM-4-9B was selected as our base model as it outperformed other two candidates. Note that the subpar performance of GLM-4-Z1-9B was primarily due to its failure to follow the required formatting guidelines—such as generating labels outside the set "Yes", "No", "To some extent", or omitting the final decision altogether.

We further compared the performance scores of the models trained on top of GLM-4-9B and Qwen2.5 14B Instruct on another randomly sampled task, *Actionability*, to examine the base model’s generalizability. As shown in Table 2, the model fine-tuned from GLM-4-9B outperformed that from Qwen2.5 14B Instruct in three out of four metrics, and demonstrated a comparable level of strict macro-F1 score.

4.2.2 Group Relative Policy Optimization and Reasoning

To examine the effectiveness of the proposed approach, we compared our method with a recently released state-of-the-art proprietary LLM, GPT 4.1 (OpenAI, 2025), released on April 14, 2025. We further compared our approach with conventional supervised fine-tuning (SFT) without rationale outputs. The results are shown in Table 3.

In *Actionability*, our GRPO-trained model outperforms all other baselines, including GPT-4.1 and conventional SFT-based model, achieving the best scores in all four metrics. In *Providing Guidance*, our method also achieves the best macro-F1 and accuracy in the lenient setting and the best accuracy in the strict setting, and shows competitive performance in strict macro-F1 as well. A similar trend is observed for *Mistake Location*, where the proposed method achieves the best strict macro-F1, lenient macro-F1, and accuracy. However, the model trained with conventional SFT performs strongly in *Mistake Identification*, calling for the need of

Base model	Strict		Lenient	
	Macro-F1	Accuracy	Macro-F1	Accuracy
GLM-4-9B	0.273	0.380	0.384	0.566
GLM-Z1-9B	0.095	0.133	0.131	0.162
Qwen2.5 14B Instruct	0.194	0.232	0.338	0.443

Table 1: Initial performance of base model candidates on *Mistake Location*, obtained on the entire development set. Best score for each metric is marked in **bold**.

Base model	Strict		Lenient	
	Macro-F1	Accuracy	Macro-F1	Accuracy
GLM-4-9B	0.701	0.756	0.861	0.888
Qwen2.5 14B Instruct	0.709	0.730	0.853	0.884

Table 2: Performance of different base models on *Actionability*, obtained on the official test set. Best score for each metric is marked in **bold**.

Methods	<i>Mistake Identification</i>	<i>Mistake Location</i>
GPT 4.1	0.410 / 0.528 / 0.699 / 0.806	0.342 / 0.355 / 0.639 / 0.673
Base model	0.393 / 0.548 / 0.634 / 0.746	0.390 / 0.468 / 0.582 / 0.641
SFT	0.715 / 0.899 / 0.900 / 0.952	0.481 / 0.726 / 0.757 / 0.819
GRPO (ours)	0.564 / 0.867 / 0.805 / 0.919	0.569 / 0.669 / 0.768 / 0.823
Methods	<i>Providing Guidance</i>	<i>Actionability</i>
GPT 4.1	0.532 / 0.613 / 0.704 / 0.790	0.567 / 0.581 / 0.827 / 0.847
Base model	0.409 / 0.516 / 0.583 / 0.738	0.417 / 0.440 / 0.697 / 0.710
SFT	0.593 / 0.617 / 0.731 / 0.815	0.542 / 0.661 / 0.730 / 0.738
GRPO (ours)	0.571 / 0.649 / 0.764 / 0.859	0.664 / 0.758 / 0.854 / 0.875

Table 3: Performance of different models, obtained on the evaluation set split from the development set. Each cell is composed of strict macro-F1 / accuracy / lenient macro-f1 / accuracy scores. Best score for each metric is marked in **bold**.

further investigation on different characteristics of each dimension. Overall, GRPO-based models consistently outperform the base model and GPT 4.1 across all dimensions, while achieving better performance than SFT-based models in three dimensions, indicating that training a model to produce pedagogically-informed rationales contributes to better evaluation performance.

4.3 Shared Task Leaderboard

The resulting models from experiments, which were submitted under the team name of bea-jh, demonstrated strong performance compared to other 2025 BEA Shared Task contestants (Kochmar et al., 2025). Our model ranked 1st in *Actionability*, and 6th in *Mistake Location* and *Providing Guidance* in the shared task’s official main metric, strict macro-F1 scores. Models trained on top of both GLM-4-9B and Qwen 2.5 14B Instruct achieved better performance than those of other contestants, demonstrating the generalizability of

the effectiveness of the proposed prompting and training strategy.

On the other hand, in *Mistake Identification*, the SFT-based model ranked 13th while the GRPO-based model would have ranked 37th out of 44 contestants. Aforementioned results are summarized in Table 4 along with scores and rankings on the shared task’s secondary metrics.

5 Conclusion

In this paper, we proposed a methodology for evaluating the pedagogical abilities of AI-powered tutors in providing helpful feedback across four key dimensions. System prompts were designed to incorporate pedagogical domain knowledge and the base model was selected based on its initial performance to generate rationale-supported evaluation. The selected model was then trained with the system prompts using Group Relative Policy Optimization (GRPO), a state-of-the-art method

Metric		<i>Mistake Identification</i>	<i>Mistake Location</i>	<i>Providing Guidance</i>	<i>Actionability</i>
Strict Macro-F1	Score	0.5873 (0.6802*)	0.5658	0.5451	0.7010 (0.7085 [†])
	Ranking	37 / 44 (13 / 44*)	6 / 32	6 / 36	1 / 30 (1 / 30 [†])
Strict Accuracy	Score	0.8449 (0.8707*)	0.7389	0.6703	0.7557
	Ranking	28 / 44 (9 / 44*)	4 / 32	4 / 36	1 / 30
Lenient Macro-F1	Score	0.8494 (0.9069*)	0.7851	0.7324	0.8609
	Ranking	32 / 44 (6 / 44*)	5 / 32	10 / 36	4 / 30
Lenient Accuracy	Score	0.9270 (0.9457*)	0.8268	0.8003	0.8875
	Ranking	28 / 44 (11 / 44*)	5 / 32	7 / 36	4 / 30

Table 4: Final scores and rankings on the official test set. Scores are shown to four decimal places, following the official leaderboard format. Along with the scores and rankings for *Mistake Location* and *Actionability* obtained by evaluating the proposed approach on the official test dataset and reranked based on the official leaderboard, officially recorded scores and rankings under the team name bea-jh that are not obtained by the proposed approach are presented inside the parentheses. *: score and ranking in the official *Mistake Identification* leaderboard, obtained by training GLM-4-9B via supervised fine-tuning. †: macro-F1 score and ranking in the official *Actionability* leaderboard, obtained by training Qwen2.5 14B Instruct via our proposed approach.

for optimizing reasoning capabilities in LLMs. As a result, our models demonstrated competitive performance in the BEA 2025 Shared Task, achieving the first place in the *Actionability* dimension.

However, the proposed approach exhibits varying performance across different evaluation dimensions. These discrepancies suggest that each dimension may require tailored modeling strategies that reflect its underlying pedagogical definitions. Future work could involve an in-depth pedagogy-based analysis of each dimension to identify how to design a high-quality evaluator. Furthermore, since our approach generates explicit rationales through reasoning, these rationales could potentially be leveraged not only for evaluation, but as a basis for improving the AI tutor’s feedback.

Limitations

We believe this study proposes an effective methodology for evaluating the pedagogical abilities of AI-powered tutors. However, the following limitations highlight areas for future investigation.

Thorough investigation of prompt engineering Since system prompts serve as instructions to LLMs, variations in prompt design can lead to different outputs and rationales. While our prompts incorporated pedagogical domain knowledge, further investigation into how each component of a prompt influences the reasoning process could lead to more effective prompt engineering strategies for evaluation tasks.

Analysis of dimension-specific characteristics

Although the proposed method achieved strong per-

formance in certain evaluation dimensions, it performed relatively poorly in a particular dimension. This discrepancy may stem from intrinsic differences among dimensions, such as varying levels of subjectivity or difficulty. Analyzing why the method performs better in some dimensions could open the way to the development of evaluation strategies tailored to each dimension.

Analysis of rationale truthfulness Generating rationales provides insights into how models "think", and the rationales generated by an evaluation model may inspire ways to improve the systems under evaluation. However, it remains an open question whether these rationales truly reflect the model’s internal decision-making process. Future work could involve further analysis to assess the stability and truthfulness of generated rationales, enabling a more qualitative understanding of reasoning-based evaluation.

References

- Ejuchegahi Anthony Angwaomaodoko. 2023. The re-examination of the dangers and implications of artificial intelligence for the future of scholarship and learning. *Traektoriâ Nauki*, 9(10):3021–3028.
- Isaiah T Awidi. 2024. Comparing expert tutor evaluation of reflective essays with marking by generative artificial intelligence (ai) tool. *Computers and Education: Artificial Intelligence*, 6:100226.
- Xiaoyu Bai and Manfred Stede. 2023. A survey of current machine learning approaches to student free-text evaluation for intelligent tutoring. *International Jour-*

- Journal of Artificial Intelligence in Education*, 33(4):992–1030.
- Logan Barnhart, Reza Akbarian Bafghi, Stephen Becker, and Maziar Raissi. 2025. Aligning to what? limits to rlhf based alignment. *arXiv preprint arXiv:2503.09025*.
- Tom Broos, Laurie Peeters, Katrien Verbert, Carolien Van Soom, Greet Langie, and Tinne De Laet. 2017. Dashboard for actionable feedback on learning skills: Scalability and usefulness. In *Learning and Collaboration Technologies. Technology in Education: 4th International Conference, LCT 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part II 4*, pages 229–241. Springer.
- William Cain. 2024. Prompting change: Exploring prompt engineering in large language model ai and its potential to transform education. *TechTrends*, 68(1):47–57.
- Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomek Korbak, David Lindner, Pedro Freire, and 1 others. 2024. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*.
- Eason Chen, Danyang Wang, Luyi Xu, Chen Cao, Xiao Fang, and Jionghao Lin. 2024. A systematic review on prompt engineering in large language models for k-12 stem education. *arXiv preprint arXiv:2410.11123*.
- Konstantin Chernyshev, Vitaliy Polshkov, Ekaterina Artemova, Alex Myasnikov, Vlad Stepanov, Alexei Miasnikov, and Sergei Tilga. 2024. U-math: A university-level benchmark for evaluating mathematical skills in llms. *arXiv preprint arXiv:2412.03205*.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- 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.
- 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. ACM.
- Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*.
- Grand View Research. 2025. AI Tutors Market Size, Share & Trends | Industry Report 2030 — grandviewresearch.com. <https://www.grandviewresearch.com/industry-analysis/ai-tutors-market-report>. [Accessed 19-04-2025].
- 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*.
- 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*.
- 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. 2024. Llm-as-a-tutor in efl writing education: Focusing on evaluation of student-llm interaction. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 284–293.
- Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- 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)*.
- Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8487–8495.
- Yeo Wei Jie, Ranjan Satapathy, Rick Goh, and Erik Cambria. 2024. How interpretable are reasoning explanations from prompting large language models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2148–2164.

- Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, and 1 others. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*.
- Hassan Khosravi, Simon Buckingham Shum, Guanliang Chen, Cristina Conati, Yi-Shan Tsai, Judy Kay, Simon Knight, Roberto Martinez-Maldonado, Shazia Sadig, and Dragan Gašević. 2022. Explainable artificial intelligence in education. *Computers and education: artificial intelligence*, 3:100074.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551.
- 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*.
- Jinu Lee and Julia Hockenmaier. 2025. Evaluating step-by-step reasoning traces: A survey. *arXiv preprint arXiv:2502.12289*.
- SeungJun Lee, Taemin Lee, Jeongwoo Lee, Yoonna Jang, and Heuseok Lim. 2023. Kullm: Learning to construct korean instruction-following large language models. In *Annual Conference on Human and Language Technology*, pages 196–202. Human and Language Technology.
- Hongxuan Liu, Haoyu Yin, Zhiyao Luo, and Xiaonan Wang. 2025a. Integrating chemistry knowledge in large language models via prompt engineering. *Synthetic and Systems Biotechnology*, 10(1):23–38.
- Jiayi Liu, Bo Jiang, and Yu’ang Wei. 2025b. Llms as promising personalized teaching assistants: How do they ease teaching work? *ECNU Review of Education*, page 20965311241305138.
- Yang Liu, Dan Iter, Yichong Xu, Shuhang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- Leo S Lo. 2023. The clear path: A framework for enhancing information literacy through prompt engineering. *The Journal of Academic Librarianship*, 49(4):102720.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Ilya Loshchilov and Frank Hutter. 2022. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.
- 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.
- 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. *arXiv preprint arXiv:2502.18940*.
- Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2023. Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics*, pages 387–402. Springer.
- 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.
- Jennifer Meyer, Thorben Jansen, Ronja Schiller, Lucas W Liebenow, Marlene Steinbach, Andrea Horbach, and Johanna Fleckenstein. 2024. Using llms to bring evidence-based feedback into the classroom: Ai-generated feedback increases secondary students’ text revision, motivation, and positive emotions. *Computers and Education: Artificial Intelligence*, 6:100199.
- OpenAI. 2024. **Best practices for prompt engineering with the openai api**.
- OpenAI. 2025. **Introducing gpt-4.1 in the api | openai**.
- 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.
- Qwen Team. 2024. **Qwen2.5: A party of foundation models!**

- Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *CoRR*.
- Timo Schick and Hinrich Schütze. 2022. True few-shot learning with prompts—a real-world perspective. *Transactions of the Association for Computational Linguistics*, 10:716–731.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseek-math: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *International Educational Data Mining Society*.
- THUDM. 2025. [Thudm/glm-4: Glm-4 series: Open multilingual multimodal chat lms](#): .
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Galouédec. 2020. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024a. 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.
- Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2024b. Do-not-answer: Evaluating safeguards in llms. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 896–911.
- 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.
- Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating mathematical reasoning beyond accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27723–27730.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024a. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.
- 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, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024b. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Jingjie Yi, Deqing Yang, Siyu Yuan, Kaiyan Cao, Zhiyao Zhang, and Yanghua Xiao. 2022. Contextual information and commonsense based prompt for emotion recognition in conversation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 707–723. Springer.
- Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *CoRR*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.
- Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038.

CU at BEA 2025 Shared Task: A BERT-Based Cross-Attention Approach for Evaluating Pedagogical Responses in Dialogue

Zhihao Lyu

University of Colorado Boulder

zhihao.lyu@colorado.edu

Abstract

Automatic evaluation of AI tutor responses in educational dialogues is a challenging task, requiring accurate identification of mistakes and provision of pedagogically effective guidance. In this paper, we propose a classification model based on BERT, enhanced with a cross-attention mechanism that explicitly models the interaction between the tutor’s response and preceding dialogue turns. This design enables better alignment between context and response, supporting more accurate assessment along the educational dimensions defined in the BEA 2025 Shared Task. To address the substantial class imbalance in the dataset, we employ data augmentation techniques for minority classes. Our system consistently outperforms baseline models across all tracks. However, performance on underrepresented labels remains limited, particularly when distinguishing between semantically similar cases. This suggests room for improvement in both model expressiveness and data coverage, motivating future work with stronger decoder-only model and auxiliary information from systems like GPT-4.1. Overall, our findings offer insights into the potential and limitations of LLM-based approaches for pedagogical feedback evaluation.

1 Introduction

Recent progress in large language models (LLMs) like GPT-4, Gemini (Team et al., 2023), and LLaMA (Grattafiori et al., 2024) has rapidly improved AI conversational agents, especially in education. AI tutors, for example, can now offer students real-time, interactive feedback to boost engagement and learning (Lin et al., 2023). However, while these models generate fluent, human-like responses, evaluating the real educational value of their feedback remains challenging (Ou et al., 2023). Standard metrics such as BLEU and ROUGE fail to capture important aspects of educational dia-

logue—like identifying student mistakes or providing helpful guidance—which highlights the need for more fine-grained, pedagogically meaningful evaluation frameworks.

To address this gap, the BEA 2025 Shared Task goes a step further than previous tasks (Tack et al., 2023) by shifting the focus from dialogue generation to evaluating how LLMs assess educational dialogues. Evaluation is based on four key dimensions (Maurya et al., 2025): (1) Mistake Identification (Tack and Piech, 2022; Macina et al., 2023; Daheim et al., 2024), (2) Mistake Location (Daheim et al., 2024), (3) Providing Guidance (Tack and Piech, 2022; Liu et al., 2023), and (4) Actionability (Daheim et al., 2024). These dimensions capture what truly matters in educational feedback, moving beyond surface-level fluency. For more details on the task and evaluation setup, please refer to the official report (Kochmar et al., 2025).

In this paper, we present our submission to the BEA 2025 Shared Task, focusing on three evaluation tracks: Mistake Identification, Mistake Location, and Providing Guidance. Our approach enhances standard LLM classifiers with a cross-attention layer to better capture the relationship between student-tutor dialogue context and the tutor’s response. Experimental results demonstrate that our method achieves strong performance across all tracks, validating the effectiveness of cross-attention for modeling educational feedback. Our team, CU, ranked 25th out of 44 in Track 1, 17th out of 31 in Track 2, and 20th out of 35 in Track 3.

2 Related Work

2.1 Early Work on Educational Feedback

Early research in educational psychology laid the theoretical foundation for understanding effective teaching practices. Hattie and Timperley (2007) proposed a widely adopted model of feedback focused on learning goals, progress monitoring, and

actionable guidance, demonstrating its critical role in student achievement. The AutoTutor system (Graesser et al., 2005) formalized key tutoring strategies—such as identifying misconceptions and prompting elaboration—within an intelligent tutoring framework. Boyer et al. (2011) introduced a data-driven approach by modeling dialogue structures using hidden Markov models to predict learning gains. Wolfe et al. (2013) and Rus et al. (2017) analyzed tutor-student dialogues to assess the quality of instructional moves using semantic similarity and discourse act classification.

2.2 LLMs for Educational Dialogue Evaluation

Recent advances in LLMs have reshaped how we engage with language and text—transforming not only natural language processing (NLP) research but also the evaluation of educational dialogues. A growing body of research explores how LLMs can be used to assess or enhance educational feedback. For example, Balse et al. (2023) investigated the ability of GPT-3.5 to explain logical programming errors, finding that while explanations were often imperfect, they reliably identified key issues. Lee et al. (2024) improved LLM-based classification accuracy by structuring prompts to encode error relationships. Molina et al. (2024) showed that LLM tutors improve accessibility for non-native English speakers, while Xu et al. (2025) built a virtual AI tutor capable of analyzing student drafts and generating error-specific feedback. Reinforcement learning approaches such as that of Scarlatos et al. (2025) have further enhanced LLM tutors by optimizing pedagogical reward functions. Kakarla et al. (2024) demonstrated the potential of LLMs in evaluating human tutor responses, highlighting both strengths and limitations.

2.3 BERT for Dialogue and Tutoring Systems

Parallel to LLM advancements, BERT-based architectures have also proven effective for educational dialogue modeling and intelligent tutoring systems (ITS). In the domain of dialogue understanding, DialogueBERT (Zhang et al., 2021) and DialBERT (Li et al.) incorporate hierarchical context and speaker-role awareness to improve performance on tasks such as disentanglement, emotion recognition, and intent detection. CS-BERT (Wang et al., 2021), trained on domain-specific customer service dialogues, introduces masked speaker prediction and turn-level segment embeddings, yielding robust re-

sults in low-resource scenarios. Within ITS applications, BERT has been adapted for various pedagogical tasks. LBKT (Li et al., 2024) combines BERT and LSTM with Rasch-based embeddings for long-sequence knowledge tracing, improving interpretability and accuracy. Tutor-KD (Kim et al., 2022) introduces tutor-guided difficulty adaptation in knowledge distillation, enhancing BERT’s generalization. Wang et al. (2024) compare BERT with ChatGPT for dialogic pedagogy support and note that, while BERT performs well in structured analysis, it lacks the interactive fluency teachers often prefer.

3 Research Gap

Despite progress in educational theory and NLP, evaluating the pedagogical quality of AI tutor responses remains difficult. Traditional methods emphasize structured feedback but rely on manual annotation and lack scalability, while LLMs offer fluency yet often miss deeper educational goals like mistake identification and guidance. Although some work proposes education-driven metrics, most automated approaches fail to effectively model dialogue context. BERT-based models show potential in educational settings but are still underused for evaluating tutor responses within full dialogue history.

To address this, we introduce a BERT-based classifier with a cross-attention mechanism that explicitly models tutor–dialogue interactions, enabling more accurate and context-aware evaluation across multiple pedagogical dimensions.

4 Methodology

In this section, we present the model architecture, including the data processing, BERT-based representation generation, and the cross-attention and classification layers. An overview of the model is illustrated in Figure 1. First, The conversation history and tutor response are preprocessed separately, with special tokens inserted at the beginning of each utterance to indicate their order. These inputs are then encoded using a pretrained BERT model to obtain high-dimensional representation. They are passed into a cross-attention layer, where the response serves as the query and the conversation history as the key and value. Finally, the cross-attended representation is fed into a classification layer that predicts one of three labels: *Yes*, *No*, or *To some extent*.

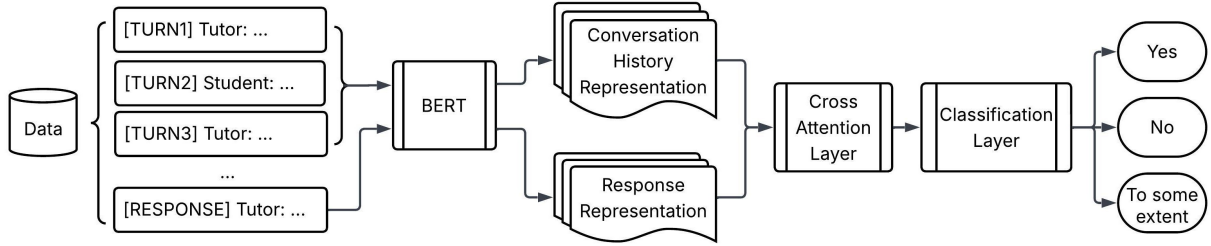


Figure 1: Overview of the proposed model architecture.

4.1 Data Preprocessing and Representation Generation

Data Augmentation. By examining the label distribution of the shared task dataset, we observed a significant imbalance between the *Yes*, *No*, and *To some extent* labels (see Table 1). Also, the *Yes* and *To some extent* labels are semantically similar, which may require the model to make finer distinctions. To address this issue without substantially altering the data distribution, we applied data augmentation only to the training set. Specifically, we used GPT-4.1 to rephrase all instances with minority labels once, thereby augmenting the dataset across all three tracks. This results in a simple 2:1 ratio between augmented and original samples for the minority classes. The ratio was determined heuristically rather than through systematic tuning, with the aim of increasing class diversity while preserving the overall label distribution. This augmentation strategy led to improved F1 scores in our subsequent experiments. The prompt used for rephrasing is provided in Appendix A.

Track 1 Label	Before Aug	After Aug
Yes	1932	1932
No	370	666
To some extent	174	313

Track 2 Label	Before Aug	After Aug
Yes	1543	1543
No	713	1283
To some extent	220	396

Track 3 Label	Before Aug	After Aug
Yes	1407	1932
No	566	1018
To some extent	503	905

Table 1: Comparison of label counts before and after data augmentation across the three tracks.

Input Labeling. To preserve the contextual meaning and sequential order of the conversation history, we manually insert order indicators (e.g., $[TURN_x]$) at the beginning of each utterance and mark the tutor’s response with a $[RESPONSE]$ token. Compared to the insertion of turn and role embeddings in DialogBERT (Zhang et al., 2021), this simple modification is easier to implement while still demonstrating effectiveness.

Representation Generation. Given BERT’s strong performance and widespread success across various NLP tasks (Devlin et al., 2019), we retain its original architecture and use its encoder only as a representation generator. Specifically, BERT first generates three types of embeddings from the input: token embeddings, segment embeddings, and position embeddings. These embeddings are added and then fed into the Transformer’s self-attention layers (Vaswani et al., 2017), which consist of multiple attention heads and stacked layers that compute contextualized representations for each token. After processing through these layers, the BERT encoder produces high-dimensional vectors as the final hidden states for both the conversation history and the tutor response. The input representation process is illustrated in Figure 2.

4.2 Cross-Attention and Classification Layer

After obtaining token-level representations of the tutor’s response and the conversation history using a BERT encoder, we combine them using a cross-attention mechanism to model the relationship between the two. Inspired by the decoder structure in the Transformer architecture, we treat the response as the **query** and the conversation history as the **key** and **value**. This allows each token in the response to selectively attend to relevant parts of the dialogue history (Figure 3). Formally, let

- $R \in \mathbb{R}^{l \times d}$ be the representation matrix of the tutor’s response, where l is the number of

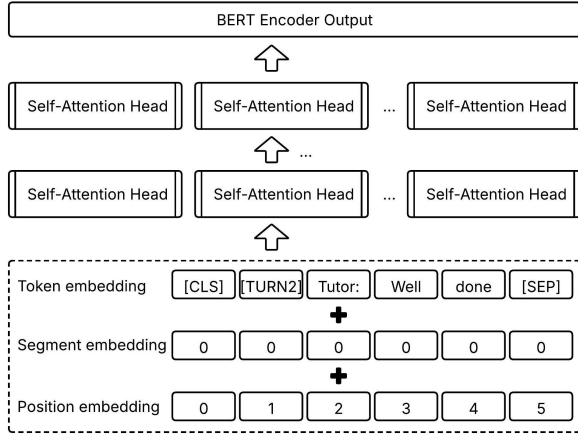


Figure 2: Illustration of the input representation process in BERT, including embedding generation and self-attention encoding.

tokens in responses, and d is the hidden size;

- $H \in \mathbb{R}^{n \times d}$ be the representation matrix of the conversation history, where n is the number of tokens in the history.

$$Weight = \text{softmax}\left(\frac{R \cdot W_Q(H \cdot W_K)^T}{\sqrt{d}}\right) \quad (1)$$

$$Attention(R, H, H) = Weight \cdot H \cdot W_V \quad (2)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are trainable projection matrices. Here, $Weight \in \mathbb{R}^{l \times n}$ represents the attention weights between each token in the response (R) and each token in the conversation history (H), where l and n are the number of tokens in the response and history, respectively. This mechanism enables the response to selectively attend to relevant segments of the dialogue context, producing a contextualized representation that informs the final classification.

After obtaining the cross-attended response representations, we extract the hidden state corresponding to the $[CLS]$ token (the first token position) to serve as the aggregate representation of the response. This vector is then passed through a dropout layer for regularization, followed by a linear classification layer that maps the hidden representation to a logits vector of dimension $\mathbb{R}^{d \times m}$, where d is the hidden size and $m = 3$ is the number of classification labels used across all tasks. The resulting logits are used to compute the weighted cross-entropy loss during training.

Algorithm 1: Dialogue-level Split with Label Distribution Balancing

Input: Training and validation dialogues

Output: Training and validation splits with similar label distributions

- 1 Compute global label distribution ratio from all dialogues
 - 2 Split dialogues into initial training (80%) and validation (20%) sets
 - 3 Compute label distribution in both sets
 - 4 **for** $iteration = 1$ **to** max_iters **do**
 - 5 **foreach** $train\ dialogue\ d_i$ ($sampled\ subset$) **do**
 - 6 **foreach** $val\ dialogue\ d_j$ ($sampled\ subset$) **do**
 - 7 Swap d_i and d_j between training and validation sets
 - 8 Compute new label distributions
 - 9 Compute ratio error in both sets
 - 10 **if** $new\ error < old\ error$ **then**
 - 11 Accept the swap
 - 12 Update label counts
 - 13 **break both loops**
 - 14 **end**
 - 15 **end**
 - 16 **end**
 - 17 **end**
-

5 Experiments

5.1 Dataset

As the shared task required, we use a development set and a test set from the MathDial(Macina et al., 2023) and Bridge(Wang et al., 2023) datasets.

Development Set: Contains 300 dialogues where students make mistakes or show confusion. Each dialogue includes the student’s last question and responses from multiple tutors (LLMs and humans), with over 2,480 responses labeled for pedagogical quality.

Test Set: Contains 200 similar dialogues. Tutor responses are not labeled and tutor identities are hidden. The set is intended only for official evaluation and is not available for model development.

Given that all 2,480 responses are associated with only 300 dialogues, we perform dialogue-level splitting to ensure that no dialogue appears in both training and validation sets. This prevents data leakage and ensures a fair evaluation. Combined with the label imbalance issue noted in Section 4,

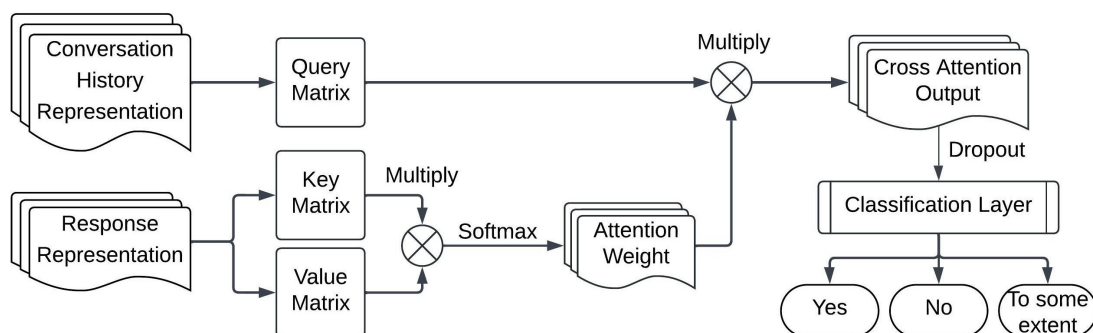


Figure 3: Overview of the cross attention mechanism and classification.

this requires careful design of the data split. First, we perform an initial 80/20 split of the data into training and validation sets. Then, to ensure that the label distributions in both sets are similar, we iteratively swap samples between them. The pseudocode is shown in Algorithm 1.

5.2 Experimental Setup

We fine-tuned all models using the BERT base uncased(110M) architecture. Inputs combined the tutor’s response and dialogue context, with cross-attention as described in Methodology. The $[CLS]$ token was used for classification, followed by dropout and a linear layer. All inputs were tokenized using the BERT tokenizer with a maximum sequence length of 512. Sequences longer than 512 tokens were truncated, and shorter sequences were padded accordingly.

Training was conducted on a single NVIDIA RTX 4060 Ti GPU. We used the AdamW optimizer with a learning rate of $2e-5$ and a batch size of 5. Models were trained for up to 5 epochs with early stopping based on Macro-F1 score on the validation set. A cosine learning rate scheduler was used, and a dropout rate of 0.1 was applied before the final classification layer. To address class imbalance, we adopted a log-weighted cross-entropy loss, where the weight for each class i was computed as $w_i = \log\left(\frac{N}{n_i}\right)$, with N the total number of samples and n_i the number of samples in class i . The overall training procedure is summarized in Algorithm 2.

5.3 Baselines

To provide a reference for zero-shot performance, we included two LLM baselines: GPT-4.1 and LLaMA 3.2 1B. For GPT-4.1, we used the OpenAI API and designed a custom prompt to elicit pedagogical labels (*Yes*, *No*, or *To some extent*) for each tutor response, given the tutor’s response and

dialogue context. This model was not fine-tuned on our dataset and operates purely in a zero-shot setting. The full prompt example is included in Appendix B. For LLaMA 3.2 1B, we used the open-source model and ran it locally. Similar to GPT-4.1, we applied a handcrafted prompt to guide the model in classifying tutor responses. The LLaMA model was also evaluated in a zero-shot. The prompt used is provided in Appendix C. These baselines allow us to assess the effectiveness of our fine-tuned BERT models against general-purpose LLMs without task-specific adaptation.

5.4 Main Results

We now present the results of our fine-tuned BERT-based models, comparing variants with and without the proposed cross-attention mechanism(CA), as well as the impact of data augmentation(Aug). These models are evaluated on all three shared task tracks and compared with the zero-shot baselines (Section 5.3). Our team, CU, participated in three tracks and ranked 25th/44 in Track 1, 17th/31 in Track 2, and 20th/35 in Track 4. The results are shown in Table 2.

5.5 Discussion

5.5.1 Improving Track 1 Performance with Cross-Attention

As shown in Table 2 and Figure 4, incorporating the cross-attention mechanism substantially improved the model’s performance on Track 1. The Macro-F1 score increased from 0.578 to 0.687, and accuracy improved from 0.849 to 0.867. While the baseline BERT model performed reasonably well on the majority class *Yes*, it failed to identify any instances of the minority class *To some extent*, as shown by a complete absence of predictions for that label in the confusion matrix. This resulted in a biased classifier with high accuracy but limited

Algorithm 2: Training procedure for BERT with Cross-Attention

Input: Training set $\mathcal{D}_{\text{train}}$, Validation set \mathcal{D}_{val} , number of epochs N , batch size B

Output: Best model parameters θ^*

```
1 Initialize BERT-based model with
  cross-attention, parameters  $\theta$  ;
2 Initialize AdamW optimizer and cosine
  learning rate scheduler ;
3  $\theta^* \leftarrow \theta$ ,  $\text{best\_val\_f1} \leftarrow 0$ ;
4 for  $\text{epoch} = 1$  to  $N$  do
5   for each batch  $(x_{\text{dial}}, x_{\text{resp}}, y)$  in  $\mathcal{D}_{\text{train}}$ 
6     do
7       Forward pass:
8          $z \leftarrow \text{Model}(x_{\text{dial}}, x_{\text{resp}})$  ;
9       Compute loss:
10         $L \leftarrow \text{CrossEntropy}(z, y)$  ;
11      Backward pass: update  $\theta$  via
12        optimizer ;
13      Update learning rate scheduler ;
14    end
15    Evaluate model on  $\mathcal{D}_{\text{val}}$  to obtain F1
16    score;
17    if  $F1 > \text{best\_val\_f1}$  then
18       $\text{best\_val\_f1} \leftarrow F1$ ;
19       $\theta^* \leftarrow \theta$  // Save best model
20    end
21 end
22 return  $\theta^*$ 
```

generalization.

In contrast, the BERT+Cross Attention model significantly reduced this bias. It not only improved the recall for the *No* class (from 47 to 55 true positives), but also successfully predicted 10 instances of *To some extent*, a class that the baseline model could not recognize at all. Although the number of correct predictions for *Yes* slightly decreased (from 413 to 374), this reflects a more balanced and context-sensitive classification behavior. These findings suggest that cross-attention enables the model to better align the tutor’s response with subtle errors in the student’s utterance, resulting in more robust performance across all categories.

5.5.2 Benefits and Limitations of Data Augmentation in Track 2 and 3

To quantitatively assess the effect of data augmentation on class-wise performance for Track 2, we compare the classification reports of the

Model	Acc.	Macro-F1
zero-shot GPT-4.1	0.807	0.557
zero-shot LLaMA 3.2 1B	0.758	0.440
BERT (no CA)	0.849	0.578
BERT + CA	0.870	0.651

(a) Track 1: Mistake Identification

Model	Acc.	Macro-F1
zero-shot GPT-4.1	0.548	0.472
zero-shot LLaMA 3.2 1B	0.619	0.371
BERT base(no CA)	0.678	0.429
BERT base + CA	0.689	0.504
BERT base + CA + Aug	0.681	0.515

(b) Track 2: Mistake Location

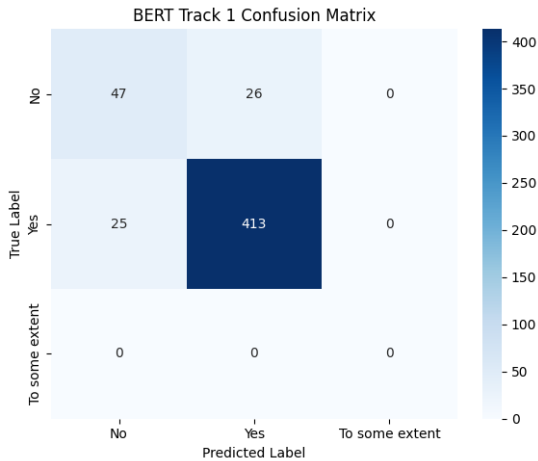
Model	Acc.	Macro-F1
zero-shot GPT-4.1	0.549	0.403
zero-shot LLaMA 3.2 1B	0.591	0.363
BERT base(no CA)	0.587	0.476
BERT base + CA	0.589	0.484
BERT base + CA + Aug	0.585	0.493

(c) Track 3: Providing Guidance

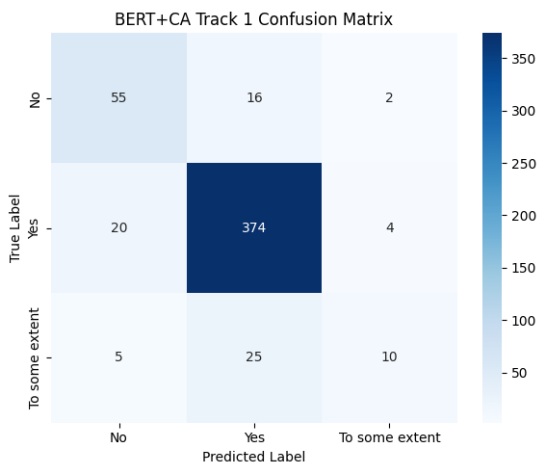
Table 2: Model performance across three task tracks.

cross-attention model before and after augmentation (see Table 3). Without augmentation, the model achieves high performance on the majority class *Yes* ($F1 = 0.77$), but almost entirely fails to recognize the minority class *To some extent* ($F1 = 0.00$). With data augmentation, the model’s ability to identify *No* and *To some extent* is significantly improved, with F1-scores rising from 0.48 to 0.63 and from 0.00 to 0.20, respectively. Although the recall for *Yes* decreases slightly (from 0.92 to 0.85), the overall classification results become more balanced, as indicated by the higher Macro-F1 score. These results highlight the utility of data augmentation in mitigating class imbalance and promoting fairer evaluation across all categories.

A similar pattern is observed for Track 3: after applying data augmentation, overall accuracy decreases slightly, while Macro-F1 improves only marginally. This suggests that the benefit of augmentation is consistent but limited when class imbalance is severe.



(a) BERT



(b) BERT+CA

Figure 4: Confusion matrix comparison on Track 1.

6 Conclusion

In this paper, we presented a system for evaluating tutor responses in educational dialogues, with a particular focus on three pedagogical dimensions as outlined in the BEA 2025 Shared Task. Leveraging a BERT-based architecture augmented with a cross-attention layer, our approach aimed to improve the model’s ability to capture context and provide more accurate multi-label classification. Experimental results demonstrate that our system achieves strong performance on Track 1, while also revealing challenges in distinguishing between semantically similar categories, such as *Yes* and *To some extent* in Track 2 and 3. Data augmentation techniques were employed to mitigate class imbalance, resulting in modest improvements, particularly in minority classes. Despite these advances, our findings indicate that substantial gaps remain before such systems can be reliably deployed in real-world educa-

Class	Precision	Recall	F1
No	0.64	0.39	0.48
Yes	0.67	0.92	0.77
To some extent	0.00	0.00	0.00

(a) BERT+CA

Class	Precision	Recall	F1
No	0.67	0.60	0.63
Yes	0.75	0.85	0.79
To some extent	0.27	0.15	0.20

(b) BERT+CA+Aug

Table 3: Class-wise precision, recall, and F1 score for Track 2 before and after data augmentation. Each class contains the same number of validation samples (No: 144, Yes: 301, To some extent: 59).

tional settings. Overall, our work contributes new insights into the application of LLMs for pedagogical evaluation and highlights key challenges for future research.

7 Future Work

During the training process, we observed that the number of cross-attention layers may influence classification accuracy. In future work, we plan to conduct further experiments with more advanced, higher-capacity decoder-only models, and systematically explore the effect of varying the number of cross-attention layers. In addition, the current cross-attention layer still struggles to recognize minority classes in dialogue. To address this, we aim to leverage state-of-the-art models as an auxiliary information. For example, we could use GPT-4.1 to first estimate the probability that each utterance in the conversation contains a mistake, and then pass these probabilities as initial attention weights to the cross-attention layer. This approach may enable the model to more precisely identify errors within the dialogue. Furthermore, GPT-4.1 could be used to perform more sophisticated data preprocessing, such as extracting all potential errors, so that the classification model only needs to determine whether the tutor’s response correctly identifies and addresses those errors.

Our current approach is inherently pedagogy-specific: it is trained on dialogue data annotated with educational dimensions, and designed to model the relationship between student language and tutor feedback. Both the training objective and

model architecture reflect the goal of evaluating responses in a pedagogically meaningful way. In the future, further gains might be achieved by incorporating explicit pedagogical constructs, such as known error types or feedback taxonomies, into the modeling process. We see this as a promising direction for enhancing both model performance and educational relevance.

8 Limitations

Despite the promising results demonstrated by our system, several limitations remain. First, while the model achieves strong performance on Track 1, its accuracy on Track 2 and Track 3 remains below 70%, with Macro-F1 scores falling short of 60%. This gap suggests that the system is not yet robust enough for real-world educational deployment. As shown in Table 3, the model tends to favor the majority class (*Yes*) and continues to struggle with the *No* and *To some extent* categories. Notably, *To some extent* is semantically close to *Yes*, and despite our data augmentation efforts, its precision in Track 2 remains below 30%, indicating substantial room for improvement in recognizing minority classes.

Second, although BERT has long been a strong performer in NLP tasks, its encoder-decoder architecture is increasingly surpassed by newer, decoder-only models such as LLaMA and Qwen (Yang et al., 2025). These models are rapidly becoming the mainstream in LLM research. However, their substantially larger parameter sizes make them less accessible for users with limited computational resources. Furthermore, the additional cross-attention layer proposed in this work increases computational demands even further. After the shared task deadline, we experimented with LLaMA 3.2 1B augmented with our cross-attention mechanism and conducted full fine-tuning. Compared to BERT, LLaMA 3.2 1B has nearly ten times more parameters, making local training on personal computers nearly infeasible. This poses an even greater barrier for educators or practitioners who may lack expertise in machine learning or access to high-performance hardware.

References

Rishabh Balse, Viraj Kumar, Prajish Prasad, and Jayakrishnan Madathil Warriem. 2023. Evaluating the quality of llm-generated explanations for logical errors in cs1 student programs. In *Proceedings of the 16th Annual ACM India Compute Conference*, pages 49–54.

Kristy Elizabeth Boyer, Robert Phillips, Amy Ingram, Eun Young Ha, Michael Wallis, Mladen Vouk, and James Lester. 2011. Investigating the relationship between dialogue structure and tutoring effectiveness: a hidden markov modeling approach. *International Journal of Artificial Intelligence in Education*, 21(1-2):65–81.

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. *arXiv preprint arXiv:2407.09136*.

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.

Arthur C Graesser, Patrick Chipman, Brian C Haynes, and Andrew Olney. 2005. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618.

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*.

John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research*, 77(1):81–112.

Sanjit Kakarla, Danielle Thomas, Jionghao Lin, Shivang Gupta, and Kenneth R Koedinger. 2024. Using large language models to assess tutors’ performance in reacting to students making math errors. *arXiv preprint arXiv:2401.03238*.

Junho Kim, Jun-Hyung Park, Mingyu Lee, Wing-Lam Mok, Joon-Young Choi, and SangKeun Lee. 2022. Tutoring helps students learn better: Improving knowledge distillation for bert with tutor network. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7382.

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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Yanggyu Lee, Suchae Jeong, and Jihie Kim. 2024. Improving llm classification of logical errors by integrating error relationship into prompts. In *International Conference on Intelligent Tutoring Systems*, pages 91–103. Springer.

- T Li, JC Gu, X Zhu, Q Liu, ZH Ling, Z Su, and S Wei. Dialbert: A hierarchical pre-trained model for conversation disentanglement. arXiv 2020. *arXiv preprint arXiv:2004.03760*.
- Zhaoxing Li, Jujie Yang, Jindi Wang, Lei Shi, and Sebastian Stein. 2024. Integrating lstm and bert for long-sequence data analysis in intelligent tutoring systems. *arXiv preprint arXiv:2405.05136*.
- Chien-Chang Lin, Anna YQ Huang, and Owen HT Lu. 2023. Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. *Smart Learning Environments*, 10(1):41.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruo Chen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- 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](#). 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.
- Ismael Villegas Molina, Audria Montalvo, Benjamin Ochoa, Paul Denny, and Leo Porter. 2024. Leveraging llm tutoring systems for non-native english speakers in introductory cs courses. *arXiv preprint arXiv:2411.02725*.
- Jiao Ou, Junda Lu, Che Liu, Yihong Tang, Fuzheng Zhang, Di Zhang, and Kun Gai. 2023. Dialog-bench: Evaluating llms as human-like dialogue systems. *arXiv preprint arXiv:2311.01677*.
- Vasile Rus, Nabin Maharjan, Lasang Jimba Tamang, Michael Yudelson, Susan R Berman, Stephen E Fancsali, and Steven Ritter. 2017. An analysis of human tutors' actions in tutorial dialogues. In *FLAIRS*, pages 122–127.
- Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025. Training llm-based tutors to improve student learning outcomes in dialogues. *arXiv preprint arXiv:2503.06424*.
- 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. *arXiv preprint arXiv:2306.06941*.
- Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *arXiv preprint arXiv:2205.07540*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Deliang Wang, Yaqian Zheng, and Gaowei Chen. 2024. Chatgpt or bert? exploring the potential of chatgpt to facilitate preservice teachers' learning of dialogic pedagogy. *Educational Technology & Society*, 27(3):390–406.
- Peiyao Wang, Joyce Fang, and Julia Reinspach. 2021. Cs-bert: a pretrained model for customer service dialogues. In *Proceedings of the 3rd workshop on natural language processing for conversational AI*, pages 130–142.
- 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*.
- Christopher R Wolfe, Colin L Widmer, Valerie F Reyna, Xiangen Hu, Elizabeth M Cedillos, Christopher R Fisher, Priscilla G Brust-Renck, Triana C Williams, Isabella Damas Vannucchi, and Audrey M Weil. 2013. The development and analysis of tutorial dialogues in autotutor lite. *Behavior research methods*, 45:623–636.
- Tianlong Xu, YiFan Zhang, Zhendong Chu, Shen Wang, and Qingsong Wen. 2025. Ai-driven virtual teacher for enhanced educational efficiency: Leveraging large pretrain models for autonomous error analysis and correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28801–28809.
- 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*.
- Zhenyu Zhang, Tao Guo, and Meng Chen. 2021. Dialoguebert: A self-supervised learning based dialogue pre-training encoder. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3647–3651.

A Prompt Used for Rephrasing

```
<task>
  <instruction>
    You are a helpful assistant for paraphrasing a target utterance in a dialogue.
    Your goal is to rewrite the <target> utterance in a different way while preserving its
    original meaning.
    The paraphrased version must be natural, fluent, and semantically equivalent.
    Make sure the paraphrase fits well within the conversation context, both before and after the
    target.

    Guidelines:
    1. Do NOT simply repeat the original sentence.
    2. Maintain the same intention, tone, and meaning.
    3. Ensure coherence with <previous_context> and <post_context>.
    4. Output only the paraphrased version of the <target>.

    Avoid repeating the same phrasing or word order.
  </instruction>

  <previous_context>
    {previous_context}
  </previous_context>

  <target>
    {target}
  </target>

  <post_context>
    {post_context}
  </post_context>
</task>
```

Appendix 1: XML prompt used for paraphrasing minority-class responses

B GPT-4.1 Prompt

We queried GPT-4.1 via the OpenAI API using a two-part prompt. The system message defined the instruction for each task, and the user message provided the specific conversation and tutor response to be evaluated.

System Instruction For Track 1

```
You are given a conversation between a tutor and a student. The last utterance is from the student
and contains a mistake. The tutor then responds to it.

Your task is to evaluate whether the tutor's response recognizes the mistake in the student's
utterance.

Use the following guidelines:

- "Yes": The mistake is clearly identified or recognized in the tutor's response.
- "To some extent": The tutor implies there may be a mistake, but does not state it clearly or seems
uncertain.
- "No": The tutor does not acknowledge the mistake (e.g., simply answers the question without
referencing the error).

Respond with exactly one of the following labels:
Yes
To some extent
No

Do not include any explanation or extra text.
```

Appendix 2: Track 1 system instruction used with GPT-4.1.

System Instruction For Track 2

You are given a conversation between a tutor and a student. The last utterance is from the student and contains a mistake. The tutor then responds to it.

Your task is to evaluate whether the tutor's response clearly identifies the mistake and where it occurs in the student's response.

Use the following guidelines:

- "Yes": The tutor clearly points to the exact location of a genuine mistake in the student's response.
- "To some extent": The tutor shows some awareness of the mistake, but the reference is vague, unclear, or easy to misunderstand.
- "No": The tutor does not provide any detail about the mistake or its location.

Respond with exactly one of the following labels:

Yes
To some extent
No

Do not include any explanation or extra text.

Appendix 3: Track 2 system instruction used with GPT-4.1.

System Instruction For Track 3

You are given a conversation between a tutor and a student. The last utterance is from the student and contains a mistake. The tutor then responds to it.

Your task is to evaluate whether the tutor's response provides correct and relevant guidance in response to the student's mistake.

Use the following guidelines:

- "Yes": The tutor provides guidance that is correct and directly relevant to the student's mistake (e.g., explanation, elaboration, hint, or examples).
- "To some extent": Some guidance is given, but it is partially incorrect, incomplete, or somewhat misleading.
- "No": No guidance is provided, or the guidance is irrelevant or factually incorrect.

Respond with exactly one of the following labels:

Yes
To some extent
No

Do not include any explanation or extra text.

Appendix 4: Track 3 system instruction used with GPT-4.1.

User Input Template

```
Conversation history:  
{conversation}  
Tutor Response:  
{response}
```

Appendix 5: User input format for GPT-4.1 prompting.

C LLaMA 3.2 1B Prompt

We used a locally hosted version of LLaMA 3.2 1B in a zero-shot setting. The full prompt sent to the model followed the expected chat-style format, including system, user, and assistant messages, as shown below.

System Instruction For Track 1

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
Evaluate whether the tutor's response recognizes the student's mistake in the conversation.

Classification guidelines:
- "Yes": The tutor clearly identifies or acknowledges the mistake in the student's utterance.
- "To some extent": The tutor implies there may be a mistake, but the identification is vague or uncertain.
- "No": The tutor does not recognize the mistake (e.g., simply answers the question without acknowledging any error).
<|eot_id|>

<|start_header_id|>user<|end_header_id|>
Dialogue transcript:
{all_history}
Final tutor response:
{response_text_full}

Does the final tutor response recognize the student's mistake?

Only respond with one of the following labels:
Yes
To some extent
No
<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>
Classification:
```

Appendix 6: Track 1 full prompt used with LLaMA 3.2 1B for zero-shot classification.

System Instruction For Track 2

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
Evaluate whether the tutor's response clearly identifies a genuine mistake and its location in the student's utterance.

Classification guidelines:
- "Yes": The tutor clearly points to the exact location of a genuine mistake in the student's response.
- "To some extent": The response shows some awareness of the mistake, but the reference is vague, unclear, or potentially confusing.
- "No": The response does not mention the mistake or provide any detail about it.
<|eot_id|>

<|start_header_id|>user<|end_header_id|>
Dialogue transcript:
{all_history}
Final tutor response:
{response_text_full}

Does the tutor's response clearly identify the mistake and where it occurs?

Only respond with one of the following labels:
Yes
To some extent
No
<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>
Classification:
```

Appendix 7: Track 2 full prompt used with LLaMA 3.2 1B for zero-shot classification.

System Instruction For Track 3

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
Evaluate whether the tutor's response provides correct and relevant guidance in response to the
student's mistake.

Classification guidelines:
- "Yes": The tutor provides guidance that is correct and directly relevant to the student's mistake (
e.g., explanation, elaboration, hint, or example).
- "To some extent": Guidance is provided, but it is partially incorrect, incomplete, or somewhat
misleading.
- "No": The response lacks guidance, or the guidance is irrelevant or factually incorrect.
<|eot_id|>

<|start_header_id|>user<|end_header_id|>
Dialogue transcript:
{all_history}
Final tutor response:
{response_text_full}

Does the tutor's response provide correct and relevant guidance?

Only respond with one of the following labels:
Yes
To some extent
No
<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>
Classification:
```

Appendix 8: Track 3 full prompt used with LLaMA 3.2 1B for zero-shot classification.

BJTU at BEA 2025 Shared Task: Task-Aware Prompt Tuning and Data Augmentation for Evaluating AI Math Tutors

Yuming Fan and Chuangchuang Tan* and Wenyu Song

20120300@bjtu.edu.cn, 21112002@bjtu.edu.cn,

20120313@bjtu.edu.cn

Beijing Jiaotong University

Abstract

We describe the BJTU submission to the BEA 2025 Shared Task on Evaluating the Pedagogical Ability of AI Tutors, which focuses on assessing AI-generated math tutoring responses across four dimensions: Mistake Identification, Mistake Location, Guidance, and Actionability. Our approach leverages a large language model (LLM) with task-specific prompt tuning and data augmentation techniques, including dialogue shuffling and class balancing. The system achieves strong results across all tracks, ranking first in Mistake Identification and performing competitively in the others. Our findings underscore the potential of prompt-based LLMs for pedagogically-aware response evaluation and offer insights into the design of AI tutors with improved educational feedback.

1 Introduction

Recent advances in large language models (LLMs) have opened up new possibilities in education, with AI-powered tutoring systems emerging as promising tools for personalized learning. These systems simulate teacher-like interactions through natural language dialogue, offering students real-time feedback and instructional support. However, evaluating the teaching capabilities of such AI tutors remains a significant challenge. On the one hand, existing evaluation frameworks lack standardization. Previous work adopts fragmented criteria, such as correctness, relevance, and actionability, making it difficult to compare model performance between studies.

However, conventional automatic metrics (e.g. ROUGE, BLEU) fail to capture key educational goals, such as effective knowledge delivery, error correction, and cognitive scaffolding. For example, Tack (Tack and Piech, 2022) focus on teacher language style, Macina (Macina et al., 2023) highlights the coherence of feedback, while Wang

(Wang et al., 2024) emphasizes the empathetic tone of responses. This fragmented landscape hinders the development of standardized benchmarks for educational AI.

To address the above challenges, the BEA 2025 Shared Task (Kochmar et al., 2025) on Evaluating the Pedagogical Ability of AI Tutors introduces the first multidimensional benchmark centered on instructional competence. Our team, Team BJTU, focuses on the context of mathematics education, particularly the process of error remediation. We aim to develop automated models that systematically evaluate five core capabilities of AI tutoring systems: Mistake identification (identifying whether a student’s response contains a mistake), mistake location (pointing to the exact location of the error), guidance (offering effective explanations or hints) and Actionability (providing responses that meaningfully guide the student’s next learning steps).

Our team, BJTU, participated in Tracks 1, 2, 3, and 4 of the BEA 2025 Shared Task and achieved strong results, ranking 1st, 2nd, 4th, and 2nd respectively. Our approach leverages state-of-the-art language models that integrate textual cues to explore the instructional capabilities of AI tutors across multiple pedagogical dimensions. This paper outlines our methodology for tackling the task, discusses the challenges we encountered, and provides insight into how model design choices impact the effectiveness of AI-generated feedback in educational dialogues.

2 Related Work

Recent work has explored the use of large language models (LLMs) in educational dialogues, with the aim of assessing their pedagogical effectiveness. Tack and Piech (Tack and Piech, 2022) proposed the AI Teacher Test, evaluating models such as GPT-3 and Blender in three dimensions: speaking

*Corresponding Author.

like a teacher, understanding the student and providing helpful responses. Their findings showed that while LLMs produce fluent dialogue, they lack pedagogical ability.

Building on this, the BEA 2023 Shared Task (Tack et al., 2023) benchmarked teacher response generation using the TSCC dataset. Top systems used models such as GPT-3.5 and GPT-4, employing prompting and response reranking strategies. Although some systems achieved high scores, the task highlighted limitations in existing evaluation metrics for educational settings.

To address these gaps, Wang (Wang et al., 2024) introduced Bridge, a framework based on cognitive task analysis that models expert decision-making during remediation. Incorporating these decisions into LLM prompts significantly improved response quality, suggesting that structured pedagogical reasoning enhances LLM performance in tutoring contexts.

3 Method

3.1 Preprocessing

During the data preprocessing phase, we organized the historical dialogues between the Tutor and Student into a format suitable for fine-tuning. For each instance, we constructed prompts such as: *The following is a tutoring dialogue in the domain of mathematics. Based on the conversation history above, your task is to evaluate the following Tutor’s Response and determine whether it successfully identifies the error in the student’s reasoning*, as illustrated in Figure 1. Using this data, we fine-tuned a large language model (LLM) to perform the evaluation task.

In the testing phase, we applied the trained model to the test set for inference. The LLM was prompted to generate an evaluation of the given Tutor response and select one of three categorical labels—yes, some extent, or no—which was then recorded as the final output.

However, relying solely on the original training data risks overfitting the model to specific linguistic patterns, thereby limiting its generalization ability. To address this, we incorporated a series of data augmentation strategies aimed at improving the model’s robustness and adaptability across diverse dialogue contexts.

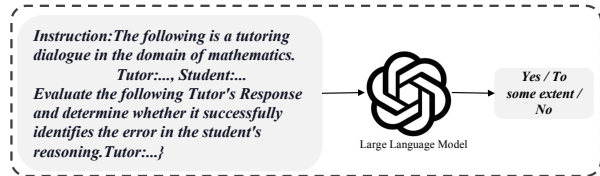


Figure 1: Prompt Construction.

3.2 Data Augmentation

In the shared task, our team BJTU demonstrated strong performance across all four tracks, as shown in table 2. We used the data set released for the BEA 2025 Shared Task (Maurya et al., 2025), which is based on a unified taxonomy for assessing pedagogical ability.

To mitigate the model’s reliance on fixed option positions and enhance its ability to generalize in ranking tasks, we adopted a dialogue-shuffling augmentation strategy. Concretely, we randomly permuted the sequence of tutor-student interaction pairs within each dialogue instance. This allows the model to better learn from the full instructional process provided by the tutor, rather than becoming overly dependent on a particular response order. By disrupting positional regularities, the model is encouraged to attend to the actual content of the tutor’s guidance. Moreover, since the dataset comprises tutoring interactions from multiple distinct AI tutors, shuffling further reduces the risk of overfitting by limiting memorization of stylistic patterns.

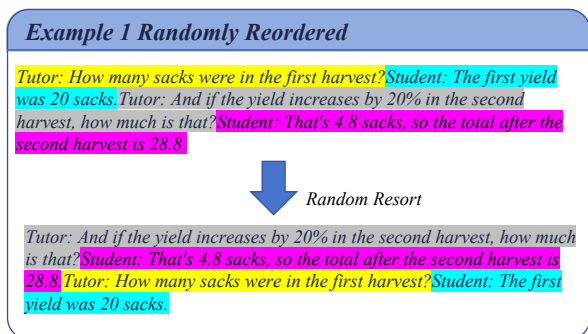


Figure 2: Randomly reordered method.

To address the issue of class imbalance observed in the training data, we applied targeted data augmentation strategies to improve model generalization. As shown in Table 1, all four subtasks exhibit a significant skew toward the “Yes” class, with notably fewer examples labeled as “To Some Extent” or “No.” This imbalance can lead the model to

overfit to the majority class and hinder its ability to accurately recognize minority class instances.

Task	Yes	To Some Extent	No
Mistake Identification	1932	174	370
Mistake Location	1543	220	713
Provide Guidance	1407	503	556
Actionability	1310	369	797

Table 1: Label distribution across the four subtasks.

To mitigate this, we implemented random down-sampling for the “Yes” instances. Specifically, we randomly sampled half of the ‘Yes’ instances, while all “No” and “To Some Extent” instances were preserved. This simple yet effective strategy reduced the dominance of the majority class and encouraged the model to better capture the characteristics of less frequent classes.

In addition, we introduced a lightweight prompt engineering strategy to improve the model’s awareness of the task objective. Taking the Mistake Identification task as an example, where the objective is to determine whether the tutor’s response successfully identifies an error in the student’s reasoning, we attached an explicit task instruction to the input. Specifically, the complete prompt template as follows: *The student’s last utterance contains a mistake. The AI tutor responds to this mistake. Your task is to assess whether the tutor’s response successfully identifies the mistake made by the student..... Your task is to evaluate the following tutor responses determine whether it successfully identifies the error in the student’s reasoning.* This additional context helps guide the model’s attention to relevant reasoning errors in the dialogue. Although the modification is simple, empirical results suggest that such task-aware prompts can improve model performance, highlighting the importance of clear task framing in multi-choice dialogue understanding tasks.

4 Experiment Results

We employed the Qwen2.5 (Bai et al., 2023) model series as the backbone and trained our models using the dataset constructed in the Method section. Specifically, we conducted training and inference using four Ascend-910B nodes, each equipped with eight GPUs. The learning rate was set to $5e-6$, the gradient accumulation steps were configured as 8, and the models were trained for a total of five epochs.

For Mistake Identification, BJTU secured 1st place with an exact macro F1 score (Ex. F1) of 0.7181, indicating its effectiveness in accurately identifying errors in student responses. In the Mistake Location track, BJTU ranked 2nd with an Ex. F1 score of 0.5940, demonstrating its ability to locate errors in student reasoning. For Providing Guidance, BJTU placed 4th with an Ex. F1 score of 0.5725, reflecting its solid performance in selecting appropriate guidance responses from multiple options. In the Actionability track, BJTU again showed strong results, ranking 2nd with an Ex. F1 score of 0.6992, demonstrating its capability to determine the practical applicability of the responses. These results highlight the consistency and versatility of BJTU’s system across different task domains, proving its robustness in handling various aspects of educational dialogue systems.

We adopted a unified strategy across all four tracks, as the tasks share similar objectives centered on evaluating and improving AI tutor responses. Instead of building separate models, we applied the same framework and prompt design to each task, which simplified our approach and proved effective across different evaluation aspects.

To further evaluate the effectiveness of different augmentation strategies, we conducted an ablation study comparing several variants of the model on the Codabench. The results are summarized in Table 3. Among all configurations, the combination of task description and dialogue shuffling achieved the best strict macro F1 score (0.7181), suggesting that explicitly describing the task helps the model better align its generation with the intended objective.

When applying shuffling alone, the model obtained the highest strict accuracy (0.8694), indicating improved precision in certain classes. However, its slightly lower F1 score suggests a trade-off in class coverage. Introducing class balancing on top of shuffling led to a modest increase in strict F1 (0.7104), but did not produce consistent improvements across all metrics. This aligns with our hypothesis that label distribution reweighting offers limited benefit when the test set closely mirrors the training set.

The base model, which only uses prompt construction without augmentation, performed slightly worse overall but still maintained reasonable robustness. These findings highlight that prompt design alone plays an important role and that combining shuffling with task description provides the most

Track	Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
1. Mistake Identification	BJTU	0.7181	0.8623	0.8957	0.9457
	TutorMind	0.7163	0.8759	0.9108	0.9528
	Averroes	0.7155	0.8675	0.8997	0.9425
	MSA	0.7154	0.8759	0.9152	0.9535
	BD	0.7110	0.8772	0.8966	0.9412
2. Mistake Location	BLCU-ICALL	0.5983	0.7679	0.8386	0.8630
	BJTU	0.5940	0.7330	0.7848	0.8261
	K-NLPers	0.5880	0.7641	0.8404	0.8610
	MSA	0.5743	0.6975	0.7848	0.8209
	SG	0.5692	0.7602	0.8118	0.8400
3. Providing Guidance	MSA	0.5834	0.6613	0.7798	0.8190
	SG	0.5785	0.7052	0.7860	0.8216
	BLCU-ICALL	0.5741	0.6716	0.7487	0.8061
	BJTU	0.5725	0.6490	0.7445	0.8100
	K-NLPers	0.5606	0.6270	0.7446	0.8000
4. Actionability	bea-jh	0.7085	0.7298	0.8527	0.8837
	BJTU	0.6992	0.7363	0.8633	0.8940
	MSA	0.6984	0.7537	0.8659	0.8908
	lexiLogic	0.6930	0.7162	0.8393	0.8675
	Phaedrus	0.6907	0.7298	0.8346	0.8650

Table 2: Top-5 system performances for each subtask, ranked by exact macro F1 (Ex. F1). Secondary metrics include exact accuracy (Ex. Acc), lenient macro F1 (Len. F1), and lenient accuracy (Len. Acc).

Strategy	Strict Acc.	Lenient Acc.	Strict F1	Lenient F1
Base(Only prompt construction)	0.8604	0.9476	0.7030	0.9026
Shuffling + Class Balance	0.8565	0.9483	0.7104	0.9017
Shuffling only	0.8694	0.9444	0.6957	0.8984
Shuffling + Task describe	0.8623	0.9457	0.7181	0.8957

Table 3: Performance of different development runs under strict and lenient matching criteria.

notable gains across evaluation metrics.

These findings suggest that among the augmentation techniques we explored, randomizing the dialogue order is particularly effective in improving the robustness of the model in unseen examples. However, the benefit of class balancing appears to depend on whether there is a label distribution mismatch between training and test sets.

5 Conclusion

In this paper, we presented BJTU’s approach to the BEA 2025 Shared Task on Evaluating the Ability of AI Tutors. Focusing on mathematics education, we designed a system that effectively evaluates tutor responses along four instructional dimensions: mistake identification, mistake location, guidance, and actionability. Our method leveraged large lan-

guage models, prompt engineering, and targeted data augmentation techniques, including dialogue shuffling and class balancing, to enhance model generalization and robustness.

Our system achieved strong overall results, ranking within the top four in all tracks, including first place in Mistake Identification. These outcomes demonstrate the potential of well-structured prompting and augmentation strategies to improve the pedagogical evaluation capabilities of LLMs.

Looking forward, we aim to explore more fine-grained annotation schemes, incorporate multimodal feedback, and develop more interpretable evaluation models. We hope our findings contribute to the advancement of standardized and scalable benchmarks for AI-assisted education.

References

- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, and et al. 2023. Qwen technical report. *arXiv:2309.16609*.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- 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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 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](#). In *Proceedings of the 2025 Conference of the North, Central and South American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251.
- 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)*, 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. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ArXiv preprint arXiv:2205.07540.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. 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.

SYSUporter Team at BEA 2025 Shared Task: Class Compensation and Assignment Optimization for LLM-generated Tutor Identification

Longfeng Chen^{1*}, Zeyu Huang^{1*}, Zheng Xiao², Yawen Zeng^{3†}, Jin Xu^{1,4}

¹South China University of Technology, Guangzhou, China

²Peking University, Beijing, China

³ByteDance, Beijing, China

⁴Pazhou Lab, Guangzhou, China

{ftclf_dh, fthzy2024, jinxu}@mail.scut.edu.cn

zhengxiao@stu.pku.edu.cn, yawenzeng11@gmail.com

Abstract

In this paper, we propose a novel framework for the tutor identification track of the BEA 2025 shared task (Track 5). Our framework integrates data-algorithm co-design, dynamic class compensation, and structured prediction optimization. Specifically, our approach employs noise augmentation, a fine-tuned DeBERTa-v3-small model with inverse-frequency weighted loss, and Hungarian algorithm-based label assignment to address key challenges, such as severe class imbalance and variable-length dialogue complexity. Our method achieved **0.969 Macro-F1 score** on the official test set, securing second place in this competition. Ablation studies revealed significant improvements: a 9.4% gain in robustness from data augmentation, a 5.3% boost in minority-class recall thanks to the weighted loss, and a 2.1% increase in Macro-F1 score through Hungarian optimization. This work advances the field of educational AI by providing a solution for tutor identification, with implications for quality control in LLM-assisted learning environments.

1 Introduction

The rapid advancement of large language models (LLMs) has opened new avenues for the development of AI-powered tutoring systems, enabling scalable and personalized learning support through intelligent conversational agents (Cai et al., 2025; Li et al., 2025). Contemporary studies demonstrate that AI-powered tutors can significantly enhance instructional efficiency (Tack et al., 2023), particularly in math education where adaptive feedback is crucial (Xu et al., 2025). Nevertheless, this technological progress introduces a critical challenge in educational practice: the growing difficulty in distinguishing LLM-generated tutor responses from those crafted by human educators. This tutor identification problem becomes particularly acute when

examining nuanced pedagogical behaviors such as error correction strategies and instructional scaffolding (Macina et al., 2023).

The emergence of sophisticated LLM-based tutors has blurred the traditional boundaries between human and machine-generated educational content. While existing detection methods (Sanh et al., 2019; Liu et al., 2019) perform adequately in binary human-vs-LLM classification scenarios, they lack the granularity required for educational applications. Specifically, these approaches fail to differentiate between various state-of-the-art LLM architectures, distinguish expert versus novice human instructors, or identify the pedagogical strategies employed by different tutor types. This limitation becomes particularly problematic given the demonstrated variations in educational outcomes based on tutor quality.

The shared task (Kochmar et al., 2025) of “Pedagogical Ability Assessment of AI-powered Tutors” (Track 5: Tutor Identification) presents three main technical challenges: **1) Class Imbalance**. Severe class imbalance in the dataset’s sample distribution across nine tutor categories (Maurya et al., 2025), **2) Complexity of Dialogue Sequences**. The complexity of variable-length dialogue sequences that complicate feature extraction, and **3) Subtle Linguistic Patterns**. Minimal lexical differences between expert humans and advanced LLMs that create subtle linguistic patterns. These characteristics render conventional classification approaches ineffective, particularly in maintaining performance across minority classes.

To address class imbalance and enhance classification performance, we employ a noise injection strategy for data augmentation, coupled with a two-stage class weight compensation mechanism. The model is fine-tuned using weighted cross-entropy loss with inverse-frequency class weighting to mitigate bias toward majority classes. For prediction, we implement an ensemble approach combining

*These authors contributed equally.

†Corresponding author.

multiple pre-trained models, followed by globally optimal label assignment via the Hungarian algorithm to ensure unique label distribution per dialogue group while maximizing prediction confidence (Kuhn, 1955). This comprehensive approach effectively handles class imbalance while maintaining prediction stability.

Our method achieved 0.969 Macro-F1 on the official test set, securing **second place in this competition**. Ablation studies¹ demonstrate component-wise improvements: a 5.3% boost in minority-class recall thanks to the weighted loss, and a 2.1% increase in Macro-F1 score through Hungarian optimization. The remainder of this paper is structured as follows: Section 2 reviews relevant literature in LLMs and text detection. Section 3 formally defines the tutor identification problem and introduces dataset. Section 4 formalizes our technical approach. Section 5 presents empirical results and case analyses. Finally, we conclude with broader implications and future directions in Section 6.

2 Related Work

LLM-generated Text Detection. The proliferation of large language models (LLMs) has spurred interest in detecting LLM-generated text. Following the emergence of the GPT-2 Output Detector (Solaiman et al., 2019), which is based on the RoBERTa pretrained model (Liu et al., 2019) and achieves up to 88% accuracy on GPT-2 text, numerous detectors have been developed. ? employs statistical analysis of word probabilities and ranks for GPT-2 detection. Habibzadeh (2023) initially used perplexity and burstiness, claiming 88% accuracy for human and 72% for AI text. OpenAI’s Text Classifier², fine-tuned on diverse models, provides probabilistic categories for distinguishing human and AI text, requiring at least 1000 characters. GP-Toolkit³ sets up multiple models (including (Sanh et al., 2019; Liu et al., 2019)). CheckForAI⁴ combines GPT-2 Output Detector with custom models. CopyLeaks⁵ claims 99.12% accuracy across languages.

In contrast to general LLM-generated text detectors, our work focuses on the more nuanced task

¹Ablation studies conducted with simplified validation due to submission constraints

²<https://platform.openai.com/ai-text-classifier>

³<https://gptkit.ai/>

⁴<https://checkforai.com/>

⁵<https://copyleaks.com/>

of identifying the specific origin of text within a defined set of tutors and LLMs. To achieve this, we leverage DeBERTa (He et al., 2020), which features disentangled attention and an enhanced mask decoder. DeBERTa has demonstrated superior performance in NLP tasks, achieving an accuracy of 91.1% on the MNLI benchmark, compared to RoBERTa-Large’s 90.2%. These results make DeBERTa a promising approach for our classification task.

3 Dataset Analysis

The dataset provided for this shared task (Kochmar et al., 2025) is sourced from the MathDial (Macina et al., 2023) and Bridge (Wang et al., 2023) datasets. The dataset, including instructional annotations developed by Maurya et al. (2025), was provided by the shared task organizers in accordance with the established annotation protocol and guidelines. Out of 300 dialogues, 200 responses were annotated by four annotators. The average Fleiss’ Kappa among the four annotators reached 0.65, indicating substantial agreement and demonstrating the reliability of this annotation task. Each dialogue includes the prior multi-turn interactions between a tutor and a student, the student’s final utterance containing an error, and a collection of responses generated by both seven large language model LLM-based tutors and human tutors in response to that utterance. The LLM tutors include: GPT-4 (Achiam et al., 2023), Gemini (Team et al., 2024), Sonnet (Anthropic, 2023), Mistral (Jiang et al., 2023), Llama-3.1-8B and Llama-3.1-405B (Grattafiori et al., 2024), and Phi-3 (Abdin et al., 2024). Human tutors are categorized into two groups: Expert and Novice.

The test set consists of 191 dialogues. These dialogues include the prior conversational context, the final incorrect student utterance, and a set of unannotated tutor responses from the same group of tutors used in the development set.

For Track 5: tutor identification task, the required data include the tutor responses and their corresponding identities. Table 1 presents the distribution of the dataset.

4 Methodology

As shown in Figure 1, we propose a unified approach to address class imbalance and enhance classification performance. It combines noise injection for data augmentation, a two-stage class weight compensation mechanism. During infer-

Class	Train Set Count	Test Set Count
Expert	300	191
Novice	76	19
Sonnet	300	191
Llama3.1-8B	300	191
Llama3.1-405B	300	191
GPT4	300	191
Mistral	300	191
Gemini	300	191
Phi3	300	191
Total	2,476	1,547

Table 1: The statistics of the dataset in track 5.

ence, we employ an ensemble of pre-trained models and apply the Hungarian algorithm for globally optimal and unique label assignment within each dialogue group. This ensures both robustness and stable prediction under imbalanced conditions.

4.1 Noise Injection for Data Augmentation

We selected several commonly used machine learning models along with the DeBERTa series for evaluation. The original training dataset was partitioned into training and validation subsets with an 8:2 ratio to facilitate comparable performance assessment. We adopted both Macro-F1 score and accuracy (ACC) as evaluation metrics. Considering that Macro-F1 demonstrates greater robustness to class imbalance, it was designated as our primary evaluation criterion. The comparative results for both metrics are presented in Table 2. Based on these experimental findings, we selected DeBERTa-v3-small for further fine-tuning to enhance its classification performance.

Model	Validation Set	
	Macro-F1 Score	Accuracy
Logistic Regression	0.796	0.811
Random Forest	0.778	0.789
Extra Trees	0.786	0.798
XGBoost	0.736	0.757
DeBERTa-v3-base	0.806	0.821
DeBERTa-v3-small	0.812	0.834

Table 2: Performance comparison of baselines on the validation set (Macro-F1 Score and Accuracy).

Subsequent analysis of the validation set predictions revealed a notable discrepancy between the model’s accuracy and Macro-F1 scores. While achieving high accuracy, the model exhibited relatively poor performance in terms of Macro-F1, suggesting inadequate handling of class imbalance. This observation indicates that the current model

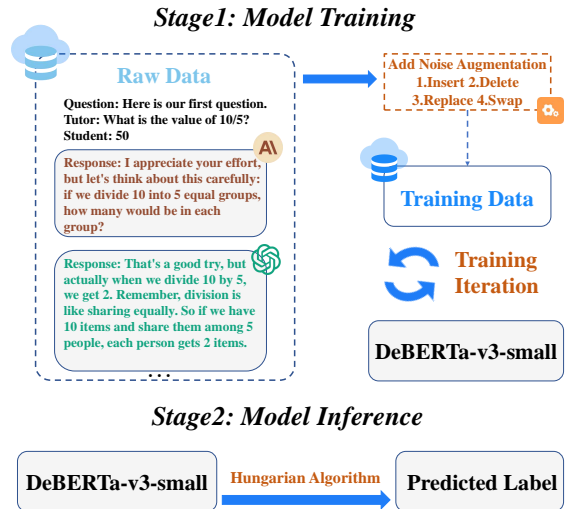


Figure 1: Overview of our proposed method.

architecture may require modification to better address the imbalanced nature of our dataset

Therefore, the original dataset is expanded through multimodal noise injection to mitigate overfitting in small-sample scenarios. For each text sample x_i , we generate its noisy variant \tilde{x}_i as follows:

$$\tilde{x}_i = T(x_i), \quad (1)$$

$$T \in \{\text{insert, delete, replace, swap}\},$$

where the noise transformation T is randomly selected with uniform probability from four operations, with a noise ratio $\alpha = 10\%$. This augmentation strategy doubles the dataset size from original N samples to $2N$. Crucially, the original labels remains unaltered during augmentation, preserving consistency in label distribution.

To address potential amplification of original class distribution disparities, we implement a two-stage class weight compensation mechanism:

4.2 Fine-tuning DeBERTa with Weighted Cross-Entropy Loss Function

To address class imbalance in the training set, we adopt an inverse-frequency weighting scheme to compute balanced class weights. Let the training set consist of C classes, with N_c denoting the number of samples in class c , and let $N_{\text{total}} = \sum_{c=1}^C N_c$ be the total number of training samples. The weight for class c is defined as:

$$w_c = \frac{N_{\text{total}}}{C \cdot N_c}, \quad c = 1, \dots, C. \quad (2)$$

This weighting strategy assigns higher importance to underrepresented classes, thereby mitigating the bias toward majority classes during model training.

Subsequently, the standard cross-entropy loss is modified by incorporating the computed class weights. Given a training batch of size B , the weighted cross-entropy loss is formulated as:

$$\mathcal{L}(\theta) = -\frac{1}{B} \sum_{i=1}^B w_{y_i} \log p_{\theta}(y_i|x_i), \quad (3)$$

where y_i denotes the true label of sample x_i , and $p_{\theta}(y_i|x_i)$ represents the predicted probability output by the model parameterized by θ .

By scaling the loss contribution of each sample according to its class weight, this approach enhances the gradient contributions from minority classes while preserving the overall optimization direction. As a result, the classification boundary becomes more sensitive to underrepresented classes, leading to improved generalization performance on imbalanced datasets.

To ensure the training effectiveness of the model, we adopt K-fold cross-validation, a robust model evaluation technique that not only maximizes the utilization of limited datasets but also reduces the dependency of evaluation results on data partitioning methods (Kohavi et al., 1995), to assess and optimize the detection model’s performance. The original training set is randomly divided into K subsets of approximately equal size. For each iteration, one subset is selected as the validation set, while the remaining K-1 subsets are used as the training set. The model’s performance is ultimately assessed by aggregating the results from the K training and validation cycles.

4.3 Prediction via Hungarian Algorithm

Given an input text set $X = \{x_1, x_2, \dots, x_n\}$, we employ k pre-trained models for prediction and average their output probabilities to mitigate the limitations of individual models, enhance generalizability, reduce prediction variance while preventing overfitting. Each model outputs a probability distribution matrix $P_i \in \mathbb{R}^{n \times c}$, with c denoting the number of classes. During the ensemble phase, we compute the average probability across all models:

$$\bar{P} = \frac{1}{k} \sum_{i=1}^k P_i. \quad (4)$$

This strategy effectively reduces model bias and enhances prediction stability. Through further analysis, we observe that each dialogue group consistently contains 7 AI responses, 1 Expert response, and randomly features 1 Novice response. Based on this pattern, we design a Hungarian algorithm-based prediction method to ensure globally optimal unique label assignment for each dialogue group. The detailed procedure is as follows:

Step 1: Cost Matrix Construction For each dialogue group $G \subseteq X$, extract its average probability matrix $\bar{P}_G \in \mathbb{R}^{m \times c}$, where $m \in \{8, 9\}$ represents the number of responses in the group. When $m = 8$, we exclude the Novice label (class 9) and adjust the probability matrix to $\bar{P}'_G \in \mathbb{R}^{8 \times 8}$. The cost matrix is defined as:

$$C = -\log(\bar{P}'_G). \quad (5)$$

This transformation converts the probability maximization problem into a linear assignment problem that minimizes negative log probabilities.

Step 2: Optimal Matching Solution The Kuhn-Munkres (Hungarian) algorithm is applied to solve:

$$\min \sum_{i=1}^m \sum_{j=1}^{c'} C_{i,j} \cdot Z_{i,j}, \quad (6)$$

subject to the constraints:

$$\sum_i Z_{i,j} \leq 1, \quad \sum_j Z_{i,j} = 1, \quad Z_{i,j} \in \{0, 1\}, \quad (7)$$

where Z denotes the assignment matrix, and $c' = c$ (when $m = 9$) or $c' = c - 1$ (when $m = 8$).

Step 3: Label Mapping and Confidence Calculation The algorithm returns optimal matching indices (i, j) , mapping column index j back to the original label (when $m = 8$, adjustment is needed to skip the Novice label). The final predicted label and confidence score are:

$$\text{label} = \arg \max_j \bar{P}_{i,j}, \quad \text{confidence} = \max_j \bar{P}_{i,j} \quad (8)$$

This strategy achieves global optimal assignment with polynomial time complexity $O(m^3)$, ensuring label uniqueness while maximizing prediction confidence.

5 Main Results

The evaluation results of all models are summarized in Table 3. It is worth noting that the De-

Model	Macro-F1 Score	Accuracy
DeBERTa-v3-small	0.812	0.834
+ Augmentation, k=1	0.888	0.888
+ Augmentation, k=5	0.901	0.901
+ Augmentation + Weighted, k=5	0.949	0.963
+ Augmentation + Weighted + Hungarian, k=5	0.969	0.966

Table 3: Performance comparison of DeBERTa-v3-small under different training strategies and ensemble settings. Among them, *Augmentation* refers to noise injection for data augmentation techniques, k denotes the number of candidates averaged during inference, *Weighted* indicates the use of weighted cross-entropy loss, and *Hungarian* refers to prediction via Hungarian algorithm.

BERTa model without noise injection for data augmentation was not submitted to the CodaLab platform. Instead, its performance was evaluated using a validation set composed of 20% of the original training data, as described in Section 4.1 on data pre-processing. The results of the other four models were obtained using the official test set via the CodaLab evaluation platform.

The DeBERTa model without any noise injection for data augmentation reflects the baseline performance of the model under the original imbalanced data distribution. After introducing noise injection for data augmentation strategies, the Macro-F1 score improved to 0.888, indicating the initial effectiveness in mitigating the impact of class imbalance. Subsequently, we applied 5-fold cross-validation to the DeBERTa model and selected the best-performing model across the folds, which further increased the Macro-F1 score to 0.901, demonstrating improved stability and generalization capability.

Building upon this, the incorporation of a weighted cross-entropy loss function led to an additional improvement in performance, with the Macro-F1 score reaching 0.949. Finally, by integrating the Hungarian algorithm for prediction optimization, the overall Macro-F1 score achieved a significant improvement, reaching 0.969. This result confirms the effectiveness of the proposed approach in addressing complex classification tasks. Our best-performing model ranked second on the official leaderboard.

6 Conclusion

In this work, we propose an effective framework for distinguishing between human-written and LLM-generated responses in mentor-style answers. Our method is based on the DeBERTa model and incorporates various techniques to enhance its general-

ization and robustness, including data augmentation strategies, a weighted cross-entropy loss function design, and a prediction optimization mechanism based on the Hungarian algorithm. This proposed approach effectively addresses the challenges posed by the rapid development of generative artificial intelligence in content authentication.

Experiments are conducted on the test set provided by the Codabench platform, and the results validate the superior performance of the framework. Furthermore, this study presents a component analysis that explores the contribution of each module to the overall performance, offering valuable insights and directions for future research and improvements in related fields.

Limitations

We still have the following limitations: 1) In terms of generalization, our method is tailored to the tutor identification task, raising questions about its generalizability to similar tasks. We plan to address this issue of generalization in future work. 2) Furthermore, although our method has demonstrated excellent performance on this competition’s test set, it has not yet been tested in real-world scenarios. We plan to apply and evaluate our method in the educational field and will share our findings when appropriate.

Acknowledgments

The authors sincerely appreciate the event organizers for their hard work, and the reviewers for their careful reading and insightful comments.

This work is supported in part by the National Natural Science Foundation of China (62372187), in part by the National Key Research and Development Program of China (2022YFC3601005) and in part by the Guangdong Provincial Key Laboratory of Human Digital Twin (2022B1212010004).

References

- Marah Abidin, Jyoti Aneja, Hany Awadallah, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.
- 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*.
- Anthropic. 2023. [The claude 3 model family: Opus, sonnet, haiku](#). Artificial Intelligence Model.
- Leng Cai, Junxuan He, Yikai Li, Junjie Liang, Yuanping Lin, Ziming Quan, Yawen Zeng, and Jin Xu. 2025. Rtbagent: A llm-based agent system for real-time bidding.
- 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*.
- Farrokh Habibzadeh. 2023. Gptzero performance in identifying artificial intelligence-generated medical texts: a preliminary study. *Journal of Korean medical science*, 38(38).
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Ekaterina Kochmar, Kaushal Kumar Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Ana  s 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)*.
- Ron Kohavi and 1 others. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.
- Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.
- Xiangyu Li, Yawen Zeng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. 2025. Hedgeagents: A balanced-aware multi-agent financial trading system.
- 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*.
- 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](#). 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.
- 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*.
- Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.
- Ana  s Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The bea 2023 shared task on generating ai teacher responses in educational dialogues. *arXiv preprint arXiv:2306.06941*.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- 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*.
- Liangyu Xu, Yingxiu Zhao, Jingyun Wang, Yingyao Wang, Bu Pi, Chen Wang, Mingliang Zhang, Jihao Gu, Xiang Li, Xiaoyong Zhu, Jun Song, and Bo Zheng. 2025. Geosense: Evaluating identification and application of geometric principles in multimodal reasoning.

BLCU-ICALL at BEA 2025 Shared Task: Multi-Strategy Evaluation of AI Tutors

Jiyuan An^{1,2}, Xiang Fu^{1,2}, Bo Liu^{1,2}, Xuquan Zong^{1,2},
Cunliang Kong³, Shuliang Liu⁴, Shuo Wang³, Zhenghao Liu⁴,
Liner Yang^{1,2,*}, Hanghang Fan^{1,2}, Erhong Yang⁵

¹National Language Resources Monitoring and Research Center for Print Media,
Beijing Language and Culture University, China

²School of Information Science, Beijing Language and Culture University, China

³Department of Computer Science and Technology, Tsinghua University, China

⁴Department of Computer Science and Technology, Northeastern University, China

⁵Kika Tech, China

lineryang@gmail.com

Abstract

This paper describes our approaches for the BEA-2025 Shared Task on assessing pedagogical ability and attributing tutor identities in AI-powered tutoring systems. We explored three methodological paradigms: in-context learning (ICL), supervised fine-tuning (SFT), and reinforcement learning from human feedback (RLHF). Results indicate clear methodological strengths: SFT is highly effective for structured classification tasks such as mistake identification and feedback actionability, while ICL with advanced prompting excels at open-ended tasks involving mistake localization and instructional guidance. Additionally, fine-tuned models demonstrated strong performance in identifying tutor authorship. Our findings highlight the importance of aligning methodological strategy and task structure, providing insights toward more effective evaluations of educational AI systems.

1 Introduction

The integration of large language models (LLMs) into educational technologies has revolutionized the landscape of AI-powered tutoring systems. These systems exhibit remarkable capabilities in generating fluent and contextually relevant responses, offering personalized learning experiences across various domains, including mathematics education. However, assessing the pedagogical effectiveness of these AI tutors extends beyond evaluating linguistic fluency or factual correctness; it necessitates a comprehensive analysis of their instructional strategies and their ability to engage students meaningfully.

To tackle the challenge of evaluating instructional quality, the 20th Workshop on Innovative Use of NLP for Building Educational Applications

(BEA 2025) introduced a shared task titled Pedagogical Ability Assessment of AI-powered Tutors (Kochmar et al., 2025). This initiative aims to establish standardized evaluation criteria for systematically assessing the pedagogical effectiveness of AI-assisted educational dialogues. The task provides a unified evaluation framework encompassing four key pedagogical dimensions: mistake identification, mistake localization, provision of guidance, and actionability of feedback. In addition to these core dimensions, the shared task includes a fifth track, Guess the Tutor Identity, which focuses on authorship attribution by determining whether a response was generated by a specific language model or a human tutor—thereby shedding light on the stylistic signatures of different LLMs. An overview of the task design is illustrated in Figure 1.

In this paper, we present our comprehensive approach to the BEA-2025 Shared Task, focusing on both pedagogical ability assessment and tutor identity attribution in AI-powered tutoring systems. We explore multiple methodological paradigms, including in-context learning (ICL), supervised fine-tuning (SFT), and reinforcement learning (RLHF), and demonstrate their respective strengths across task tracks. Our empirical results show that SFT excels in structured classification tasks, while ICL, supported by advanced prompting strategies, proves more effective in open-ended reasoning settings. Furthermore, we validate the use of fine-tuned LLM classifiers for authorship attribution, achieving competitive performance even in black-box conditions. Our findings not only highlight the importance of methodological alignment with task structure but also provide practical insights into building robust evaluation systems for educational AI.

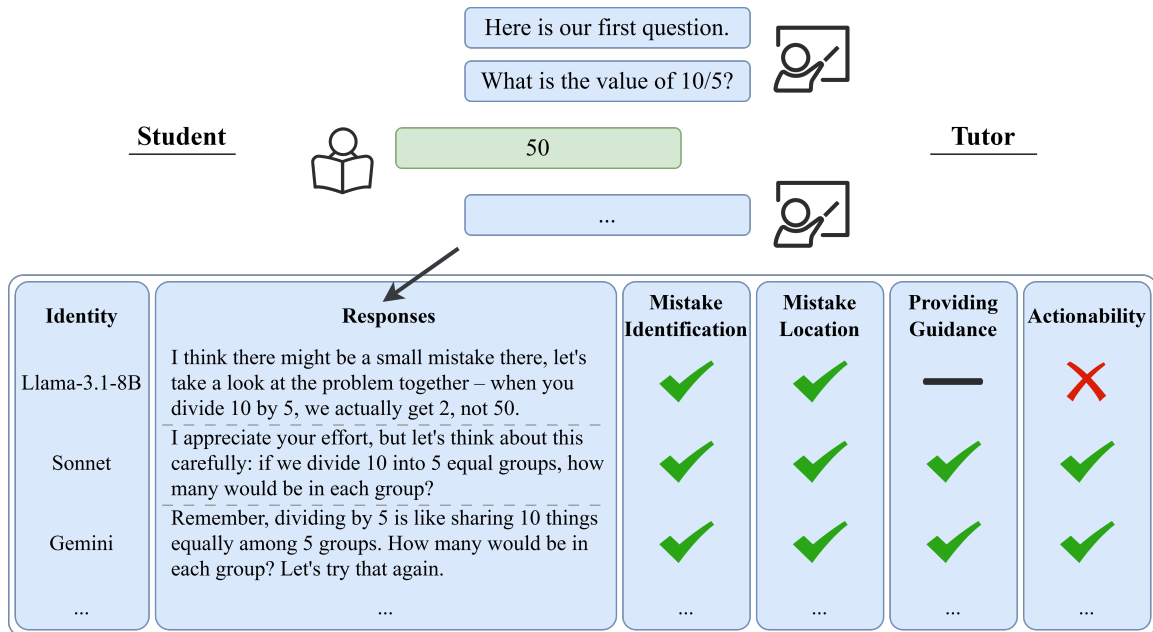


Figure 1: Illustration and Description of the Task for Evaluating Pedagogical Ability. The figure presents a sample math problem given to a student, along with three distinct responses generated by AI tutors. Each response is assessed across four pedagogical dimensions: Mistake Identification, Mistake Localization, Guidance Provision, and Actionability. A green check mark (✓) denotes that the behavior is clearly exhibited (Yes), a red cross (✗) indicates that it is absent (No), and a black dash (–) signifies that the behavior is only partially present or ambiguously demonstrated (To some extent).

2 Related Works

This section provides a brief overview of the BEA-2025 Shared Task and reviews two key methodological areas: LLM-as-a-Judge techniques for evaluating pedagogical quality in the first four tracks, and authorship attribution methods for identifying tutor sources in the final track.

2.1 Pedagogical Ability Assessment of AI-powered Tutors

With rapid advancements in artificial intelligence (AI) and natural language processing (NLP), AI-powered tutoring systems—especially those leveraging large language models (LLMs)—have demonstrated significant potential in educational contexts, including mathematics instruction. However, effectively evaluating the instructional quality of these systems requires more than simply assessing linguistic fluency or factual accuracy. It demands deeper analysis of their pedagogical strategies and the quality of their interactions with students.

To address this need, the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025) introduced a shared task titled “Pedagogical Ability Assessment of AI-powered

Tutors.” This task aims to establish standardized evaluation criteria that systematically measure instructional quality in AI-supported educational dialogues.

Specifically, the task focuses on mathematics-based tutor-student dialogues, with special emphasis on capturing student errors and misconceptions that surface during problem-solving interactions. Task participants are provided dialogue samples sourced from the MathDial and Bridge datasets, which include:

Multi-turn interactions between students and AI-powered tutoring systems; Student utterances containing errors or expressions of uncertainty; Tutor responses generated by various AI systems based on different LLMs, as well as select responses from human tutors. To facilitate comprehensive and consistent evaluation, the organizers propose a unified taxonomy based on the pedagogical framework introduced by Maurya et al. (2024), comprising four core dimensions:

- **Mistake identification:** Evaluating whether the AI correctly detects a student’s error.
- **Mistake location:** Identifying the exact position of the error within a student’s utterance.

- **Providing guidance:** Assessing the AI’s ability to deliver appropriate hints, explanations, or guiding questions.
- **Actionability:** Determining whether the provided feedback clearly points students toward actionable next steps.

Beyond the primary subtasks focusing on instructional quality dimensions, the BEA 2025 shared task also introduces Track 5: **Guess the tutor identity**, designed to explore relationships between the stylistic characteristics of AI tutors and their underlying source models. In this subtask, participants must identify the specific model or human tutor behind a tutoring system’s response based solely on text content.

To support research and system development, the organizers have released the MRBench V3 dataset¹, consisting of 300 development dialogues and 191 test dialogues, encompassing interactions with both AI and human tutors. Each dialogue is annotated according to the four pedagogical dimensions. Participants are further encouraged to develop automated evaluation systems to assess the pedagogical capabilities of AI-generated tutoring interactions within this structured evaluation framework.

2.2 LLM-as-a-Judge

With the widespread adoption of large language models (LLMs) in various natural language processing tasks, effectively evaluating the quality of their generated outputs has become a prominent research area. Traditional automatic evaluation metrics such as BLEU (2002) and ROUGE (2004) exhibit limitations in capturing semantic coherence and contextual relevance in generated texts. To address these issues, recent work has proposed the "LLM-as-a-Judge" approach, which leverages powerful LLMs as evaluators to assess outputs produced by other models. This method not only enhances automation of the evaluation process but also demonstrates judgment capabilities comparable to human evaluators across various tasks (Liu et al., 2023).

From an output perspective, existing LLM-as-Judge implementations can generally be categorized into three frameworks (Li et al., 2024): (a) Scoring: The most frequently adopted evaluation paradigm, in which the LLM assigns numerical

scores to candidates, enabling quantitative comparisons. (b) Ranking: Particularly useful when establishing a relative ordering among candidates, allowing for evaluations that do not rely on explicit scoring scales. (c) Selection: Effective in decision-making scenarios, enabling the LLM to directly choose the most suitable output from a set of provided candidates.

In terms of construction methodologies, approaches to building reliable LLM-based judges primarily belong to two categories:

- Prompting Strategies:** Properly designed prompting methods and pipelines further enhance judgment accuracy and mitigate evaluation bias (Gu et al., 2024). Key prompting approaches include: **Position Swapping:** Systematically changing candidates’ positions in prompts to reduce position-induced biases. **Inclusion of Rubric and Reference Information:** Directly offering clear rubrics or reference materials to guide the LLM’s evaluation criteria. **Inter-LLM Cooperation:** Implementing collaborative processes (e.g., voting mechanisms, structured debates) among multiple LLM-based judges, thereby balancing individual-model biases. **In-Context Demonstrations:** Providing relevant examples within prompts, a method shown to significantly improve evaluation performance via the model’s in-context learning capabilities.
- Tuning-Based Methods:** Supervised Fine-Tuning (SFT) is the predominant strategy, where LLMs are explicitly trained to judge based on collected prompt-response evaluation datasets (Zhu et al., 2023). Through supervised training, models gain the capability to perform nuanced judgments in specific tasks.

By carefully selecting and combining these tuning methods and prompting strategies, robust and reliable LLM-based judge systems can be effectively constructed, thereby enabling more accurate evaluation across diverse and complex NLP tasks.

2.3 Authorship Attribution

Authorship Attribution (AA) aims to identify the authorship of unknown texts by analyzing linguistic features. The underlying assumption of AA is that different authors—including humans and large language models (LLMs)—exhibit distinct characteristics in lexical diversity, syntactic structures, and

¹https://github.com/kaushal0494/UnifyingAITutorEvaluation/tree/main/BEA_Shared_Task_2025

discourse styles. Previous authorship attribution methods predominantly focused on distinguishing texts produced by various human authors. However, with the rise and advancement of large language models, differentiating between human-generated and LLM-generated texts, as well as identifying texts produced by specific LLMs, has increasingly become a focal area of research.

Current authorship attribution methods can be categorized as follows:

- (a) **Style-based methods** utilize lexical, syntactic, and structural features to capture the distinct writing styles of authors. For instance, [Kumara and Liu \(2023\)](#) extracted lexical, syntactic, and structural features from texts to train classifiers for tracing the origin of generated texts. Nevertheless, these methods tend to perform poorly when distinguishing between closely related LLMs, such as Llama-3-8B and Llama-3-405B.
- (b) **Probability-based methods** hypothesize that generated texts have a higher generation probability when evaluated by their original source model, and thus rely on differences in probability distributions calculated by various language models for the same text. For example, POGER ([Shi et al., 2024](#)) performs attribution by repeatedly sampling representative tokens to estimate generation probabilities. However, these approaches are highly sensitive to text length, as shorter texts may yield inaccurate probability estimates.
- (c) **Partial rewriting methods** involve partially regenerating segments of a text using candidate generation models and evaluating the source by measuring edit distances between original and regenerated segments. For example, DNA-GPT ([Yang et al., 2023](#)) uses the first half of the target text as a prompt and compares the regenerated latter half with the original to assess attribution. Despite their utility, these methods require multiple invocations of models and significantly depend on prompt design and generation strategies.
- (d) **Model fine-tuning methods** leverage the semantic feature distributions learned from texts authored by different sources through fine-tuning language models. [Chen et al. \(2023\)](#), for instance, fine-tuned the T5 model to cre-

ate T5-Sentinel, achieving effective attribution across five models including GPT-3.5 and LLaMA-7B. Similarly, [Fu et al. \(2025\)](#) proposed the FDLLM method based on LoRA fine-tuning, which effectively detects and distinguishes texts generated by various LLMs in multilingual and cross-domain black-box scenarios. However, these methods typically require extensive annotated data for training.

3 Data

The BEA-2025 Shared Task is based upon the Mr-Bench dataset, which primarily incorporates dialogue data from two publicly available mathematical instructional datasets: MathDial ([Macina et al., 2023](#)) and Bridge ([Wang et al., 2023](#)).

MathDial Dataset The MathDial dataset consists of approximately 3,000 one-on-one teacher-student dialogues focusing on multi-step mathematical reasoning problems. These dialogues were generated by pairing human teachers with a large language model (LLM) specifically trained to simulate common student mathematical errors.

Bridge Dataset The Bridge dataset comprises 700 real-world online tutoring dialogues. These dialogues highlight the challenges novice teachers encounter in addressing student mathematical errors. Each dialogue is annotated by expert educators, explicitly identifying student misconceptions, correction strategies, and underlying instructional intents.

From these two datasets, the organizing team generated seven additional LLM-as-tutor responses for each dialogue, supplementing the original tutor responses in Bridge and MathDial. All tutor responses, including both the original and the newly generated ones, were systematically annotated according to the pedagogical effectiveness taxonomy proposed by [Maurya et al. \(2024\)](#). A development set of 300 dialogues and a testing set of 191 dialogues were constructed from this expanded and annotated pool. Additionally, a subset of the data underwent dual annotation by four independent annotators, yielding an average Fleiss' Kappa coefficient of 0.65. This indicates substantial inter-annotator agreement, thereby ensuring the reliability and robustness of the labeled data for the shared task.

3.1 Data Analysis and Statistics

Due to the limited availability of training data, we plan to expand the current dataset by annotating portions of the MathDial dataset and the unused data from the Bridge dataset. First, we analyzed how MRBench was created from the two aforementioned datasets. Specifically, the MathDial dataset includes fields such as 'question', 'student_incorrect_solution,' and 'conversation,' which can be reorganized into the MRBench format as illustrated below. The MRBench dataset is constructed as a sequential dialogue; the only additional data processing required is labeling each utterance with the corresponding speaker identity (Tutor or Student).

```
Tutor: Hi, could you please provide a step-
by-step solution for the question below? The
question is: {'question'}
Student: {'student_incorrect_solution'}
Tutor: {'conversation'-Tutor[0]}
Student: {'conversation'-Student[0]}
.....
```

Subsequently, we counted the number of responses present within each dialogue in the dataset, as shown in Figure 2.

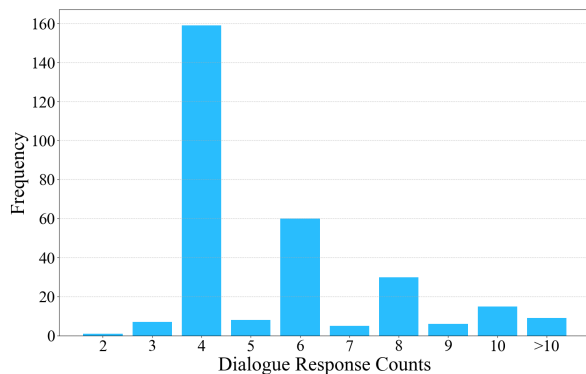


Figure 2: Distribution of Dialogue Response Counts

Finally, we identified which segments of the original datasets have already been utilized. Since dialogues from the original MathDial and Bridge datasets were randomly truncated when composing the MRBench dataset, we uniformly truncated each original dialogue to a maximum length of four turns for consistency and processed them into a standardized format. We then calculated the similarities between dialogues from MathDial and MRBench (as well as Bridge and MRBench) based on the BLEU metric. By identifying the dialogue entries

with the highest BLEU scores, we constructed a mapping list indicating data usage. Table 1 provides a summary that quantifies the relationships and overlaps among these three datasets.

	MathDial	Bridge	Total
Development set	224	76	300
Test set	172	19	191

Table 1: Dialogue Counts in Development and Test Sets

3.2 Data Correction and Processing

In the process of aligning MRBench with the two original datasets, we observed that a small subset of corresponding instances exhibited significantly lower BLEU scores than average. Upon deeper analysis of these instances, we identified certain issues within the provided datasets that could potentially affect data preprocessing procedures and subsequent model performance.

Role Label Mismatches In the MathDial dataset, we found cases where dialogue responses were mismatched with their corresponding role labels. For example, in the original data: "... on dog toys.\n 42.00 \n Tutor: Hi Ayisha can you talk me through your workings? \n Student: Sure! First I calculated that three full price toys cost 3 x 12.00 =36.00. Then I calculated that one half price toy costs 12.00/2 =6.00. Finally, I added the two amounts together ..." was extracted as: "... on dog toys.\n 42.00 \n Tutor: I added the two amounts together ...".

We believe that this issue was introduced by a comma-based preprocessing heuristic. Specifically, we infer that the task organizers intended to exclude student names or other personally identifiable mentions in tutor responses, motivated by Haim et al.'s (2024) finding that the mention of personal names might introduce unwanted bias into large language models. The heuristic presumably involved removing the segment from the beginning of the tutor's response up to the first comma, presuming that the first comma typically delineates the student's name from the main message. However, if a tutor response lacked commas at expected locations, this strategy inadvertently caused excessive removals, leading to instances where portions of students' answers mistakenly appeared as part of the tutor responses. Consequently, this may impact the model's understanding of the correct answer and its evaluation of the tutor response.

Irrelevant Dialogue Openings Within the Bridge dataset, we identified certain instances where initial conversational utterances were unrelated or irrelevant to the core mathematical problems, such as: "Student: okey \n Tutor: Now we have the same denominators so we can subtract the numerators directly.". This issue was likely introduced through the data-segmentation strategy applied to real-world dialogue corpora.

Consecutive Utterances To be consistent with a large language model’s expected conversational structure of strictly alternating turns between user and model responses, we merged consecutive responses from the same speaker within the datasets.

These procedures were conducted through a combination of automated filtering and manual verification, with further details provided in Appendix A.

4 Methodology

In this section, we present an overview of the three primary approaches explored in the BEA-2025 shared task: in-context learning (ICL), supervised fine-tuning (SFT), and reinforcement learning (RL).

4.1 In-Context Learning

In-context learning (ICL) enables large language models (LLMs) to accomplish specific tasks solely by leveraging input prompts, without the need for updating model parameters.

As an initial step, we investigate the performance of leading proprietary (or large-scale parameter) large language models on instructional ability evaluation tasks. We construct our inputs from historical dialogue contexts, teacher responses, and corresponding evaluation dimensions using the MRBench V3 dataset. Models evaluated include GPT-4o (Hurst et al., 2024), GPT-o3-mini (OpenAI, 2025), Gemini-2.5-pro (DeepMind, 2025), Grok-3 (xAI, 2025), Deepseek-R1 (DeepSeek-AI et al., 2025), and Claude-3.7 (Anthropic, 2025). To effectively elicit optimal model performance, mitigate potential biases, and enhance the robustness of our evaluation, we employ several prompt engineering strategies:

- (a) **Explicit Scoring Criteria:** Clearly-defined evaluation criteria with three distinct performance levels are provided within the prompt to guide model judgments.

- (b) **Contextual Demonstrations:** Relevant illustrative examples are embedded within prompts to enhance the models’ comprehension of tasks, assessment dimensions, and rating standards.
- (c) **Multiple Sampling:** Inspired by the self-consistency property observed in large language models, we sample model outputs multiple times under the same temperature setting and utilize majority voting to determine final results.

Moreover, we experiment with various alternative prompt formulations under each prompting strategy to identify the most effective configuration. Detailed descriptions of our prompt construction methodology can be found in Appendix B.1.

Additionally, we have assessed the performance of open-source and smaller-scale models, including Llama-3.1-8B, QwQ-32B, and the Qwen2.5 series (Yang et al., 2024), to facilitate subsequent supervised fine-tuning and reinforcement learning stages.

4.2 Supervised Fine-tuning

Supervised fine-tuning (SFT) refers to adapting a pretrained language model to a specific task by training it on labeled data. This process updates the model parameters to minimize the discrepancy between model predictions and ground-truth annotations.

In comparison to in-context learning, supervised fine-tuning explicitly embeds task definitions and requirements into the model itself through parameter adjustments. To circumvent performance constraints that may arise from overly prescriptive prompt designs, we have streamlined and adjusted the instruction templates and expected outputs as shown in Appendix B.1.

As shown above, the model is no longer required to generate textual feedback; instead, it directly outputs the designated classification label. This modification aims to simplify the construction of the supervised training dataset and mitigates the risks of overly rigid or overfitted model responses typically associated with explicitly requesting textual elaboration.

Based upon the MRBench V3 dataset, we partition the data into a training-validation split with ratios of 95% and 5%, respectively. We subsequently conduct supervised fine-tuning of the Qwen 2.5-14B model across four distinct evaluation dimensions using LLaMA-Factory (Zheng et al., 2024).

Fine-tuning enables the model to internalize nuanced patterns and task-specific subtleties, thereby significantly improving its performance on evaluation metrics. To enhance computational efficiency and guard against overfitting, we explore parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA). These methods enable selective updating of specific parameter subsets, substantially reducing computational demands while preserving model performance. Detailed specifications of the exact hyperparameters adopted throughout the fine-tuning process are presented in Appendix B.2.

As for Track-5, the task is to identify the source of anonymized natural language texts, namely attributing texts to their corresponding "mentor" models. This track comprises nine distinct classes: an expert mentor, a junior mentor, as well as seven different large language models (LLMs), formulating a typical multi-class classification scenario. This setting is especially challenging due to the short nature of test samples, the inclusion of texts generated by unseen black-box models, and the sophisticated need to distinguish closely related models, such as Llama-3-8B and Llama-3-405B. Thus, the task imposes high demands on the classifier's generalization capability and its ability to capture subtle stylistic differences among different models.

Inspired by the approach of FDLLM (Fingerprint Detection for LLMs), we propose employing a large language model-based authorship attribution classifier. More specifically, we leverage parameter-efficient supervised fine-tuning methods based on Low-Rank Adaptation (LoRA) with the pretrained Qwen 2.5-7B model. Through fine-tuning, the model learns distinct and subtle stylistic "fingerprints" inherent in texts produced by different language models, enabling effective identification of the generating model given an anonymized text input. Details on data construction and model fine-tuning processes are provided in Appendix B.3.

4.3 Reinforcement Learning

Reinforcement Learning (RL) provides a training framework in which models learn to make sequential decisions by maximizing cumulative rewards. Typically, RL is utilized to align model outputs with human preferences, a process known as Reinforcement Learning from Human Feedback (RLHF).

In the educational assessment evaluation task, it is natural to consider applying RLHF to align

large language models (LLMs) with the evaluation ratings annotated by human experts. To this end, we employ RLHF via veRL (Sheng et al., 2024) to fine-tune Qwen 2.5-7B outputs based on human-annotated preferences. Specifically, our approach mainly involves the following two essential steps:

- (a) **Reward Function:** To encourage detailed thinking within the model-generated textual feedback, thereby improving its overall performance, we design a reward function to enforce appropriate response structure and classification correctness. Concretely, we assign a 0.1 reward for adhering to the prescribed formatting structure ("Feedback: ... [Classification] (A/B/C)") and a 1.0 reward when model predictions correctly match human-annotated evaluation ratings.
- (b) **Policy Optimization:** We optimize the LLM's output strategy by maximizing the predicted rewards from the reward function. During this step, we explore optimization algorithms such as Proximal Policy Optimization (PPO) and Generalized Reference Policy Optimization (GRPO) to enhance both stability and efficiency during policy updates.

Through the RLHF process, we initially expect that the model can be guided to generate responses that are not only accurate but also closely aligned with human instructional preferences, ultimately increasing their practical value and instructional quality in educational dialogue contexts. However, we observe limited performance improvements following RLHF training, alongside unexpected generation issues, such as output consisting of repeated special tokens produced solely to obtain formatting-related rewards.

5 Results

In this section, we report the performance of our proposed methods on both the development and test sets.

5.1 Performance on the Development Set

In-Context Learning Method

As described in Section 3.1, we evaluated several advanced large language models on the teaching ability assessment task using the development set, with results shown in Figure 3. Among these models, Gemini 2.5-Pro achieved the best results across all four evaluated dimensions, substantially

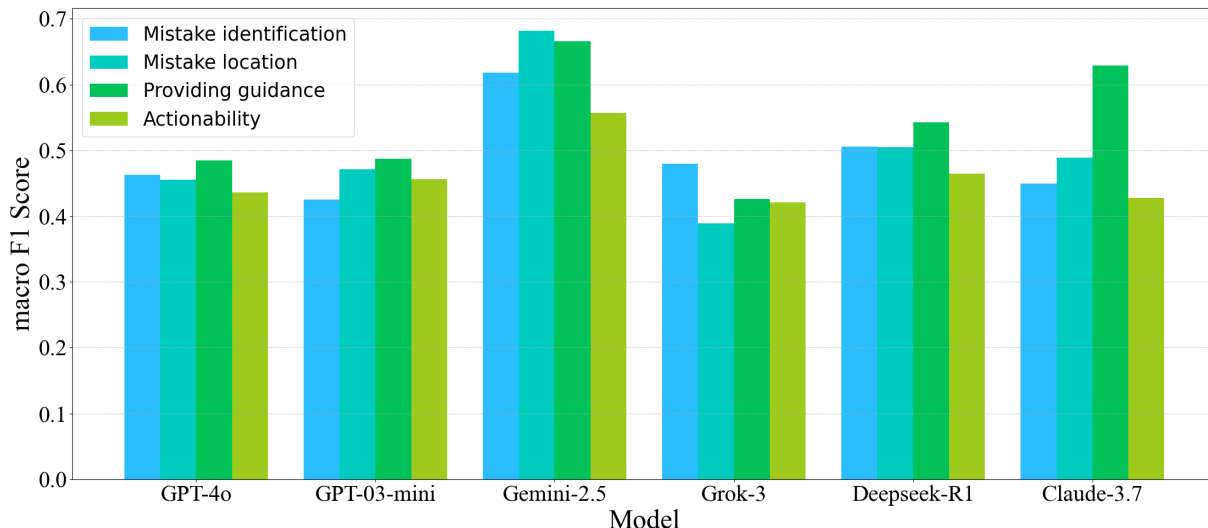


Figure 3: Performance of Proprietary Large Language Models in Pedagogical Ability Assessment

outperforming the other five models. Thus, we decided against adopting an ensemble approach, which would involve combining predictions from diverse heterogeneous models through voting. Instead, we opted to increase robustness by conducting multiple sampling procedures on the outputs from the Gemini 2.5-Pro model for our final submission.

Supervised Model Fine-tuning Method We separately evaluated several smaller-scale open-source models, including Llama-3.1-8B, QwQ-32B, Qwen 2.5-32B, and Qwen 2.5-14B. Although QwQ-32B obtained the highest scores overall, it has been observed by Kirk et al. (2023) that reinforcement learning from human feedback (RLHF) optimization may result in degradation of model performance during supervised fine-tuning (SFT), specifically affecting generalization to out-of-distribution (OOD) data. Motivated by this consideration, we chose to supervise-fine-tune Qwen 2.5-32B and Qwen 2.5-14B—both demonstrating strong performance and free of RLHF optimizations—as our base models for the teaching ability evaluation task.

5.2 Performance on the Test Set

Table 2 summarizes the highest rankings achieved by our proposed methods in the evaluation phase, detailed by each evaluation track: Track 1 (Mistake Identification): 12th out of 44; Track 2 (Mistake Location): 1st out of 31; Track 3 (Providing Guidance): 3rd out of 35; Track 4 (Actionability): 8th out of 29, and Track 5 (Guess the Tutor Identity): 5th out of 20.

Additionally, we observed differential strengths of the two methodological approaches we adopted: the in-context learning method performed notably better in Tracks 2 and 3, while the supervised fine-tuning method exhibited superior performance specifically in Tracks 1 and 4. Table 3 reports the highest observed scores for each of the two methodologies on the test set.

6 Discussion

Upon further analysis of track-specific performance, we find a clear methodological divide between the strengths of supervised fine-tuning (SFT) and in-context learning (ICL). We hypothesize that these performance differences are rooted in the task structure and cognitive load required for each evaluation dimension:

SFT advantages in Track 1 and Track 4: Both of these tracks can be framed as relatively discrete classification tasks. Track 1 requires the model to detect the existence of a mistake, often a binary or ternary decision. Track 4, similarly, involves judging whether the tutor’s response provides actionable next steps—a decision that can be learned reliably from labeled data with consistent annotation guidelines. SFT excels in such tasks due to its ability to internalize structured decision boundaries from annotated examples, especially when paired with simplified input formats and explicit label mappings. Moreover, SFT benefits from parameter adaptation, allowing it to specialize in subtle categorical distinctions that prompt-based inference might overlook.

ICL advantages in Track 2 and Track 3: In

Track	Rank	Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
Mistake Identification	1	BJTU	0.7181	0.8623	0.8957	0.9457
	⋮	⋮	⋮	⋮	⋮	⋮
Mistake Location	12	BLCU-ICALL	0.6822	0.8578	0.8909	0.9418
	1	BLCU-ICALL	0.5983	0.7679	0.8386	0.8630
Providing guidance	1	MSA	0.5834	0.6613	0.7798	0.8190
	⋮	⋮	⋮	⋮	⋮	⋮
Actionability	3	BLCU-ICALL	0.5741	0.6716	0.7487	0.8061
	1	bea-jh	0.7085	0.7298	0.8527	0.8837
	⋮	⋮	⋮	⋮	⋮	⋮
Guess the tutor identity	8	BLCU-ICALL	0.6735	0.7363	0.8596	0.8856
	1	Phaedru	0.9698	0.9664	/	/
	⋮	⋮	⋮	⋮	⋮	⋮
	5	BLCU-ICALL	0.8930	0.8908	/	/

Table 2: Rankings and Results of BLCU-ICALL in 5 tracks

	Track-1	Track-2	Track-3	Track-4
ICL	0.6600	0.5983	0.5741	0.5956
SFT	0.6822	<i>0.5582</i>	<i>0.5446</i>	0.6735

Table 3: Comparison of peak performance across tracks for in-context learning (ICL) and supervised fine-tuning (SFT) methods on the test set. Due to time constraints during the test phase, SFT results for Tracks 2 and 3 were not submitted; instead, italicized scores denote performance on 5% of the development set.

contrast, Track 2 (locating the specific position of a student’s error) and Track 3 (generating pedagogically appropriate guidance) require deeper interpretive reasoning and open-ended judgment. These tasks often lack rigid decision templates and depend heavily on nuanced understanding of conversational context, semantics, and pedagogical intent. Large-scale proprietary models like Gemini-2.5-Pro, when supported by advanced prompting (e.g., rubric-injection and contextual demonstrations), are capable of flexible reasoning and generalization—making ICL a better fit. Notably, these models benefit from large-scale parameter, broader pretraining and instruction tuning, allowing them to leverage latent reasoning abilities not easily transferred through task-specific fine-tuning alone.

In Track 5 (authorship attribution), our use of fine-tuned Qwen2.5-based classifiers achieved notable success, ranking 5th overall. This validates the feasibility of using stylistic “fingerprints” for source model identification even under black-box

constraints. Nevertheless, distinguishing between highly similar models (e.g., LLaMA variants) remains challenging, especially when input samples are short or lack distinctive syntactic structures.

7 Conclusion

This paper presents our comprehensive approach to the BEA-2025 Shared Task, focusing on both pedagogical ability assessment and tutor identity attribution in AI-powered tutoring systems. We explore multiple methodological paradigms, including in-context learning (ICL), supervised fine-tuning (SFT), and reinforcement learning (RLHF), and demonstrate their respective strengths across task tracks. Our empirical results show that SFT excels in structured classification tasks, while ICL, supported by advanced prompting strategies, proves more effective in open-ended reasoning settings. Furthermore, we validate the use of fine-tuned LLM classifiers for authorship attribution, achieving competitive performance even in black-box conditions. Our findings not only highlight the importance of methodological alignment with task structure but also provide practical insights into building robust evaluation systems for educational AI.

Limitations

Our work is subject to several limitations. For the task of Pedagogical Ability Assessment, different evaluation dimensions are not independent; rather,

they are closely interrelated. Utilizing potential synergies among these evaluation dimensions is a plausible direction that remains largely unexplored in this study. Additionally, in Track 5, there is one particularly crucial piece of information that we failed to fully exploit: the constraint that each tutor identity label can appear at most once for the same dialogue.

Acknowledgments

This work was supported by the Funds of Research Project of the National Language Commission (No. ZDA145-17), the MOE (Ministry of Education in China) Project of Humanities and Social Sciences (No. 23YJCZH264), the Fundamental Research Funds for the Central Universities, and the Research Funds of Beijing Language and Culture University (No. 25YCX134).

References

- Anthropic. 2025. Claude 3.7 sonnet. <https://www.anthropic.com/news/claude-3-7-sonnet>.
- Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. [Token prediction as implicit classification to identify llm-generated text](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Google DeepMind. 2025. Gemini 2.5 pro. <https://deepmind.google/models/gemini/pro/>.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *ArXiv*, abs/2501.12948.
- Zhiyuan Fu, Junfan Chen, Hongyu Sun, Ting Yang, Ruidong Li, and Yuqing Zhang. 2025. [Fdllm: A text fingerprint detection method for llms in multi-language, multi-domain black-box environments](#). *ArXiv*, abs/2501.16029.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. [A survey on llm-as-a-judge](#). *ArXiv*, abs/2411.15594.
- Amit Haim, Alejandro Salinas, and Julian Nyarko. 2024. [What’s in a name? auditing large language models for race and gender bias](#). *ArXiv*, abs/2402.14875.
- OpenAI Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mkadry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alexander Kirillov, Alex Nichol, Alex Paino, and 397 others. 2024. [Gpt-4o system card](#). *ArXiv*, abs/2410.21276.
- 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*, abs/2310.06452.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Tharindu Kumarage and Huan Liu. 2023. [Neural authorship attribution: Stylometric analysis on large language models](#). *2023 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 51–54.
- 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. 2024. [From generation to judgment: Opportunities and challenges of llm-as-a-judge](#). *ArXiv*, abs/2411.16594.
- Chin-Yew Lin. 2004. [Rouge: A package for automatic evaluation of summaries](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Yang Liu, Dan Iter, Yichong Xu, Shuo Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-eval: Nlg evaluation using gpt-4 with better human alignment](#). In *Conference on Empirical Methods in Natural Language Processing*.
- 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*, abs/2305.14536.
- Kaushal Kumar Maurya, KV Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2024. [Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors](#). *ArXiv*, abs/2412.09416.
- OpenAI. 2025. Openai o3-mini. <https://openai.com/index/openai-o3-mini>.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Annual Meeting of the Association for Computational Linguistics*.

- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*.
- Yuhui Shi, Qiang Sheng, Juan Cao, Hao Mi, Beizhe Hu, and Danding Wang. 2024. Ten words only still help: Improving black-box ai-generated text detection via proxy-guided efficient re-sampling. In *International Joint Conference on Artificial Intelligence*.
- Rose Wang, Qingyang Zhang, Carly D. Robinson, Susanna Loeb, and Dorottya Demszky. 2023. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *North American Chapter of the Association for Computational Linguistics*.
- xAI. 2025. Grok 3 beta — the age of reasoning agents. <https://x.ai/news/grok-3>.
- Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.
- Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *ArXiv*, abs/2305.17359.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.
- Lianghui Zhu, Xinggong Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *ArXiv*, abs/2310.17631.

A Data Correction and Processing

We addressed three types of issues in the MRBeach V3 dataset that may negatively impact the effectiveness of the pedagogical ability assessment model.

Role Label Mismatches

As mentioned previously, we conducted alignment between the MRBeach V3 dataset and the original MathDial dataset by calculating surface-level similarity using the BLEU score. Subsequently, we corrected erroneous labels through threshold-based automated filtering combined with manual annotations. Table 4 below shows the frequency of role-label mismatch errors and their corresponding indices in the development and test sets.

Irrelevant Dialogue Openings

The segmentation strategy applied to real-world conversation data occasionally resulted in semantically unrelated dialogues being grouped into the same segment, consequently introducing irrelevant information not directly related to the core mathematical problems. To handle this issue, we identified dialogues in MRBeach V3 where the student’s utterance is the initial turn, as many of these cases exemplified irrelevant conversation openings. A summary of these cases is provided in Table 5 below.

Consecutive Utterances

To better align the dialogues with the standard conversational format used by large language models—alternating question-answer interactions between two speakers—we identified and merged consecutive utterances belonging to the same speaker role within MRBeach V3. Detailed statistics of this merging process are presented in Table 6 below.

B Methodology Details

Below are detailed descriptions regarding in-context learning and supervised fine-tuning methods for Pedagogical Ability Assessment and Tutor Identification.

B.1 Prompt Construction Methodology Details

This section provides the prompt templates which yielded the best performance for in-context learning and supervised fine-tuning methods.

ICL Prompt Template

System Prompt:

You are a critic evaluating a tutor’s interaction with a student, responsible for providing a clear and objective single evaluation score based on specific criteria. Each assessment must accurately reflect the absolute performance standards.

User Prompt:

Objective: Evaluate the quality of a teacher’s latest response within the context of an ongoing conversation with a student. Your evaluation must be based solely on the provided information and result in structured feedback and a grade classification.

Inputs:

* **Evaluation Indicators:** “{definition}”
* **Grading Criteria:** {rubric}
* **Conversation History:** “{history}”
* **Teacher’s Latest Reply:** “{response}”

Instructions:

- Analyze:** Carefully review the **Teacher’s Latest Reply** in the context of the **Conversation History**.
- Evaluate:** Assess the **Teacher’s Latest Reply** strictly against each point listed in the **Evaluation Indicators**.
- Formulate Feedback:** Write a detailed feedback statement. This statement must clearly explain *how* the teacher’s reply performs against the **Evaluation Indicators**, citing specific examples from the reply or history where applicable. Your reasoning should be evident *within* this feedback structure.
- Assign Grade:** Based on your evaluation and the provided **Grading Criteria**, determine the appropriate classification (A, B, or C).
- Format Output:** Present your response *only* in the following format, without any additional introductory or concluding remarks: ‘Feedback: (Your detailed feedback statement based on evaluation indicators) [Classification] (A, B, or C)’

Dataset	Frequency	Index
Development set	26	18, 36, 56, 79, 100, 116, 122, 155, 168, 174, 177, 182, 183, 188, 195, 201, 205, 225, 252, 262, 264, 271, 277, 282, 290, 295
Test set	16	4, 28, 31, 33, 37, 42, 51, 61, 94, 98, 99, 108, 120, 129, 172, 183

Table 4: Role Label Mismatches

Dataset	Frequency	Index
Development set	16	3, 15, 23, 40, 42, 44, 65, 163, 175, 202, 221, 227, 248, 254, 257, 293
Test set	1	115

Table 5: Irrelevant Dialogue Openings

SFT/RL Prompt Template

Track 1: Mistake Identification

System: You are a Senior Teaching Supervisor.

Input: Has the tutor explicitly pointed out that there was a mistake in a student's response?

- A: Yes (The tutor's response recognizes there is a mistake, or provides some practical guidance.)

- B: To some extent

- C: No (The tutor's response believes that the question had been completely resolved, or no connection.)

* Conversation History: "{history}"

* Teacher's Latest Reply: "{tutor_response}"

Track 2: Mistake Location

System: You are a Senior Teaching Supervisor.

Input: Does the tutor's response accurately point to a genuine mistake and its location?

- A: Yes (the tutor clearly points to the exact location of a genuine mistake in the student's solution)

- B: To some extent (the response demonstrates some awareness of the exact mistake, but is vague, unclear, or easy to misunderstand)

- C: No (the response does not provide any details related to the mistake)

* Conversation History: "{history}"

* Teacher's Latest Reply: "{tutor_response}"

Track 3: Providing Guidance

System: You are a Senior Teaching Supervisor.

Input: Does the tutor offer correct and relevant guidance, such as an explanation, elaboration, hint, examples, and so on?

- A: Yes (the tutor provides guidance that is correct and relevant to the student's mistake)

- B: To some extent (guidance is provided but it is fully or partially incorrect, incomplete, or somewhat misleading)

- C: No (the tutor's response does not include any guidance, or the guidance provided is irrelevant to the question or factually incorrect)

* Conversation History: "{history}"

* Teacher's Latest Reply: "{tutor_response}"

Track 4: Actionability

System: You are a Senior Teaching Supervisor.

Input: Is it clear from the tutor's latest reply what the student should do next?

- A: Yes (the response provides clear suggestions on what the student should do next)

- B: To some extent (the response indicates that something needs to be done, but it is not clear what exactly that is)

- C: No (the response does not suggest any action on the part of the student (e.g., it simply reveals the final answer))

* Conversation History: "{history}"

* Teacher's Latest Reply: "{tutor_response}"

Dataset	Role	Continuous times	Frequency	Index
Development set	Tutor	2	36	4, 12, 16, 19, 24, 37, 41, 42, 43, 45, 57, 66, 73, 80, 101, 107, 117, 136, 156, 160, 164, 169, 176, 178, 202, 203, 228, 249, 253, 255, 263, 265, 278, 283, 291, 294
Development set	Tutor	3	38	3, 10, 14, 18, 20, 31, 35, 38, 49, 64, 71, 77, 82, 92, 110, 111, 122, 124, 135, 138, 151, 157, 174, 200, 212, 215, 231, 232, 239, 252, 260, 264, 270, 273, 275, 290, 292, 299
Development set	Student	2	5	104, 175, 196, 222, 258
Test set	Tutor	2	18	5, 22, 29, 32, 34, 43, 51, 78, 95, 99, 109, 115, 18, 121, 130, 173, 184, 191
Test set	Tutor	3	14	38, 39, 40, 46, 62, 82, 92, 98, 111, 113, 131, 166, 176, 188

Table 6: Consecutive Utterances

B.2 Supervised Fine-tuning Details

To perform supervised LoRA fine-tuning of Qwen 2.5-14B, we utilized two L40S servers, each equipped with eight GPUs throughout our experiments. For implementation, we employed LLaMA-Factory, and the key configuration parameters are detailed as follows:

- **finetuning_type:** lora
- **lora_target:** all
- **template:** qwen
- **cutoff_len:** 2048
- **per_device_train_batch_size:** 2
- **gradient_accumulation_steps:** 4
- **lora_dropout:** 0.1
- **learning_rate:** 2.0e-4
- **num_train_epochs:** 30.0
- **lr_scheduler_type:** cosine
- **warmup_ratio:** 0.1

B.3 Tutor Identification Details

We fine-tune the Qwen 2.5-7B model to develop a large language model-based authorship attribution classifier for identifying the origin of anonymous

texts. The classifier model takes the instructor’s response text as input and outputs the corresponding instructor identity label. In this section, we present the format of the instruction dataset and the key hyperparameters used in fine-tuning.

Track 5: Tutor Identification

```
## Instruction: Determine which model generated the following text.
## Input: Here is the generated text: {tutor_response}
```

- **finetuning_type:** lora
- **lora_target:** all
- **template:** qwen
- **cutoff_len:** 2048
- **per_device_train_batch_size:** 2
- **gradient_accumulation_steps:** 4
- **lora_dropout:** 0.1
- **learning_rate:** 5.0e-4
- **num_train_epochs:** 26.0
- **lr_scheduler_type:** cosine
- **warmup_ratio:** 0.1

Phaedrus at BEA 2025 Shared Task: Assessment of Mathematical Tutoring Dialogues through Tutor Identity Classification and Actionability Evaluation

Rajneesh Tiwari

Independent Researcher, India
rajneesh.vish1@gmail.com

Pranshu Rastogi

Independent Researcher, India
rastogipranshu29@gmail.com

Abstract

As Large Language Models (LLMs) are increasingly deployed in educational environments, two critical challenges emerge: identifying the source of tutoring responses and evaluating their pedagogical effectiveness. This paper presents Phaedrus' comprehensive approach to the BEA 2025 Shared Task, addressing both tutor identity classification (Track 5) and actionability assessment (Track 4) in mathematical tutoring dialogues. For tutor identity classification, we distinguish between human tutors (expert/novice) and seven distinct LLMs using cross-response context augmentation and ensemble techniques. For actionability assessment, we evaluate whether responses provide clear guidance on student next steps using selective attention masking and instruction-guided training. Our multi-task approach combines transformer-based models with innovative contextual feature engineering, achieving state-of-the-art performance with a CV macro F1 score of 0.9596 (test set 0.9698) for identity classification and 0.655 (test set Strict F1 0.6906) for actionability assessment. We were able to score rank 5th in Track 4 and rank 1st in Track 5. Our analysis reveals that despite advances in human-like responses, LLMs maintain detectable fingerprints while showing varying levels of pedagogical actionability, with important implications for educational technology development and deployment. Our code and implementation details are publicly available at https://github.com/Rajneesh-Tiwari/BEA_2025_shared_task.

1 Introduction

The integration of Large Language Models (LLMs) into educational environments has created new opportunities and challenges for tutoring systems. As AI-powered tutors become increasingly prevalent, two fundamental questions emerge: (1) Can we reliably identify the source

of tutoring responses to ensure transparency and accountability and (2) How effectively do these responses guide students toward learning objectives (Kochmar et al., 2022)

The BEA 2025 Shared Task (Kochmar et al., 2025) addresses these critical questions through two complementary tracks. Track 5 challenges participants to classify the source of mathematical tutoring responses, distinguishing between human tutors (expert and novice) and seven different LLMs: Gemini, GPT-4, Llama3-405B, Llama3-8B, Mistral, Phi3, and Claude Sonnet. Track 4 focuses on evaluating the actionability of these responses—whether they provide clear guidance on what students should do next, a crucial factor in effective pedagogical feedback (Daheim et al., 2024).

These tasks are inherently related: understanding who generated a response and how actionable it is provides a comprehensive view of educational dialogue quality. Our hypothesis is that different tutors (human or AI) not only leave distinctive linguistic fingerprints but also demonstrate varying capabilities in providing actionable guidance. This multi-dimensional analysis offers insights into the current state of AI tutoring systems and their pedagogical effectiveness compared to human tutors.

Our team (Phaedrus) approach leverages transformer-based models enhanced with task-specific innovations. For identity classification, we implement cross-response context augmentation, allowing models to compare different responses to the same question, and use specialized attention masking to focus on response characteristics. For actionability assessment, we develop instruction-guided training with selective attention mechanisms that focus on response-specific features indicating clear guidance. Both tasks benefit from sophisticated ensemble techniques and constraint satisfaction post-processing.

This paper presents our top-ranked systems for both tracks, describing our unified methodology, training strategies, and comprehensive analysis of results. Our findings demonstrate that while LLMs are becoming increasingly sophisticated in generating human-like responses, they still exhibit detectable patterns that distinguish them from human tutors, and they show varying capabilities in providing actionable pedagogical guidance.

2 Related Work

Recent research in educational dialogue assessment has focused on multiple dimensions of quality evaluation. [Tack and Piech \(2022\)](#) introduced a framework for evaluating LLM-based tutors across three dimensions: whether they speak like a teacher, understand a student, and help a student. Building on this work, [Tack et al. \(2023\)](#) organized shared task on generation of teacher responses in educational dialogues. The goal of the task was to benchmark the ability of generative language models to act as AI teachers, replying to a student in a teacher-student dialogue using existing automatic metrics (e.g., BERTScore ([Zhang* et al., 2020](#)), DialogRPT ([Gao et al., 2020](#))) and manual evaluation aligned with the proposed dimensions, highlighting ongoing challenges in the reliable assessment of pedagogical dialogue quality.

2.1 Tutor Identity and AI Detection

The challenge of distinguishing between human and AI-generated text has established several foundations. [Guo et al. \(2023\)](#) demonstrated that transformer models effectively identify LLM-generated text through distinctive linguistic patterns, performing linguistic analysis to identify patterns between ChatGPT and human expert responses. In educational contexts specifically, [Chen et al. \(2024\)](#) introduced Dr.Academy, a benchmark for evaluating LLMs’ questioning capabilities across general, humanities, science, and interdisciplinary educational domains, revealing that different models demonstrate varying strengths and distinctive patterns.

Our work extends these approaches by addressing a more complex classification problem: distinguishing not just between human and AI responses, but between multiple specific AI models and different types of human tutors in educational contexts.

2.2 Actionability and Pedagogical Effectiveness

The assessment of pedagogical effectiveness in tutoring responses has gained increasing attention. [Macina et al. \(2023\)](#) introduced MathDial, a dataset for mathematical tutoring dialogues, and evaluated tutor responses using coherence, correctness, and equitable tutoring criteria. [Wang et al. \(2024\)](#) assessed tutoring responses based on usefulness, care, and human-likeness, providing additional dimensions for evaluation.

Most relevant to our actionability assessment, [Daheim et al. \(2024\)](#) introduced a framework for evaluating tutoring responses that includes actionability as a key criterion, defining it as whether a response makes it clear what the student should do next. Their findings suggest that even state-of-the-art LLMs struggle to consistently provide actionable guidance in educational contexts.

2.3 Technical Approaches

For student response evaluation, [Fateen and Mine \(2023\)](#) compared in-context meta-learning and semantic score-based similarity approaches for automated short answer grading in Arabic, demonstrating different computational approaches to evaluating student responses. Additionally, [Maurya et al. \(2025\)](#) developed a comprehensive evaluation taxonomy for assessing LLM-powered AI tutors, highlighting distinctive features in AI-generated pedagogical interactions.

Our work builds upon these foundations while introducing novel techniques specifically tailored to both identity classification and actionability assessment in educational dialogues, including cross-response context augmentation, constraint satisfaction optimization, and instruction-guided training approaches.

3 Dataset and Task Overview

Both tracks utilize a unified dataset of mathematical tutoring dialogues ([Maurya et al., 2025](#)) combining MathDial ([Macina et al., 2023](#)) and Bridge ([Wang et al., 2024](#)) datasets with 300 development dialogues. Track 5 requires classifying responses into nine tutor categories (human expert/novice and seven LLMs), with each conversation containing unique tutor identifications. Track 4 evaluates response actionability using three categories (Yes/To some extent/No). Both tasks use exact macro F1 score as the primary evaluation metric.

4 Methodology

Our team (Phaedrus) approach combines multiple transformer-based models with task-specific architectural enhancements and ensemble techniques. We develop a unified framework that addresses both tutor identity classification and actionability assessment while leveraging shared components and complementary innovations.

4.1 Base Model Architecture

The core of our system utilizes several transformer variants, each selected for specific strengths in educational dialogue analysis:

- **DeBERTa-v3-large** (He et al., 2023): 24 layers, 1024 hidden size, 304M parameters
- **DeBERTa-v3-base** (He et al., 2023): 12 layers, 768 hidden size, 86M parameters
- **DeBERTa-v3-small** (He et al., 2023): 6 layers, 768 hidden size, 44M parameters
- **Longformer-base-4096** (Beltagy et al., 2020): 12 layers, 768 hidden size, 149M parameters, with efficient attention for long sequences
- **BigBird-RoBERTa-large** (Zaheer et al., 2021): 24 layers, 1024 hidden size, 340M parameters, with block sparse attention
- **Qwen-2.5-0.5B** (Qwen et al., 2025): 24 layers, 1024 hidden size, 0.5B parameters, featuring advanced positional embeddings and multi-query attention
- **Zephyr-7B-alpha** (Tunstall et al., 2023): 32 layers, 4096 hidden size, 7B parameters, based on Mistral architecture with sliding window attention

4.2 Shared Architectural Enhancements

Both tasks benefit from several common architectural innovations:

Response Tokenization and Selective Attention: We added special tokens [R_START] and [R_END] to explicitly mark tutor response boundaries. This enables custom attention masking that zeros out attention weights for tokens beyond the [R_END] marker, forcing models to focus specifically on response content rather than surrounding context.

Generalized Mean (GeM) Pooling: Instead of standard mean pooling, we implemented GeM pooling with a learnable parameter p to compute sequence-level representations. Given a sequence of hidden vectors $x = \{x_1, x_2, \dots, x_n\}$, where each $x_i \in \mathbb{R}^d$ is the hidden representation of the i -th token and $n = |x|$ is the sequence length, GeM pooling is defined as:

$$\text{GeM}(x) = \left(\frac{1}{n} \sum_{i=1}^n x_i^p \right)^{1/p} \quad (1)$$

Here, the exponentiation and root are applied element-wise, and $p \in \mathbb{R}$ is a learnable parameter that controls the sharpness of the pooling operation.

Multi-Sample Dropout: Inspired by (Inoue, 2020), we implemented multi-sample dropout with varying rates (0.2 to 0.27) applied to the same representation, then averaged the results. This acts as an implicit ensemble, reducing variance without additional computational cost.

4.3 Task-Specific Innovations

4.3.1 Track 5: Identity Classification Enhancements

For tutor identity classification, we developed several specialized techniques:

Cross-Response Context Augmentation: Rather than treating each response in isolation, we concatenate all available responses to the same question from different tutors, creating rich comparative context. This allows models to learn distinctive patterns by seeing how different tutors address identical student queries.

Constraint Satisfaction Post-processing: We formulated the response classification task as a constraint satisfaction problem to ensure that each class is assigned at most once per conversation, reflecting the assumption that a tutor identity should not repeat in a single dialogue.

Let c denote a conversation with a set of responses $\mathcal{R}_c = \{r_1, r_2, \dots, r_n\}$, and let $p_{r,j}$ represent the predicted probability that response $r \in \mathcal{R}_c$ belongs to class j , where $j \in \{0, 1, \dots, 8\}$. Total there are 9 classes starting from 0 to 8 where class "0" is considered as "novice". We define binary decision variables $x_{r,j} \in \{0, 1\}$ indicating whether response r is assigned to class j . The objective is to maximize the total assignment confidence while satisfying the uniqueness constraint per class within each conversation:

$$\text{maximize } \sum_c \sum_{r \in \mathcal{R}_c} \sum_{j=0}^8 p_{r,j} \cdot x_{r,j} \quad (2)$$

$$\text{subject to } \sum_{j=0}^8 x_{r,j} = 1 \quad \forall r \in \mathcal{R}_c \quad (3)$$

$$\sum_{r \in \mathcal{R}_c} x_{r,j} \leq 1 \quad \forall j \in \{0, \dots, 8\}, \forall c \quad (4)$$

$$x_{r,j} \in \{0, 1\} \quad (5)$$

We opted for a greedy algorithm due to its practical efficiency and implementation simplicity. By prioritizing responses with the highest prediction confidence and assigning them the most probable unassigned class, the method effectively resolves assignment conflicts with minimal computational cost. Empirical results show that this approach improves macro F1 scores by 2–3%, highlighting its effectiveness in enforcing consistent class assignments within conversations.

Algorithm 1 Constraint Satisfaction Algorithm

- 1: **for all** conversations c **do**
 - 2: $A_c \leftarrow \emptyset$ ▷ Set of already assigned classes in conversation c
 - 3: Sort responses $r \in \mathcal{R}_c$ by $\max_j p_{r,j}$ in descending order
 - 4: **for all** response r in sorted order **do**
 - 5: $\hat{j} \leftarrow \arg \max_{j \notin A_c} p_{r,j}$ ▷ Best unassigned class
 - 6: Assign class \hat{j} to response r
 - 7: $A_c \leftarrow A_c \cup \{\hat{j}\}$
 - 8: **end for**
 - 9: **end for**
-

4.3.2 Track 5: Meta-Model Ensemble with Pseudolabeling

Our Track 5 ensemble combines six transformer models through a sophisticated meta-modeling pipeline and was able to achieve 1st position Table 1:

1. **Base Model Predictions:** We collect class probability outputs from all six transformer models (54 features total)
2. **Feature Enhancement:** We augment with TF-IDF vectors, count vectors, linguistic features, and math-specific markers

3. **Gradient Boosting:** We train LightGBM, XGBoost, and CatBoost models on combined features

4. **Pseudolabeling:** High-confidence test predictions (probability > 0.85) are added to training data with constraint satisfaction

5. **Voting:** Final predictions use weighted voting across all meta-models

4.3.3 Track 4: Actionability Assessment Enhancements

For actionability assessment, we implemented instruction-guided training:

Actionability Criteria Instruction: We incorporated explicit actionability assessment criteria directly into model input:

```

Instruction: Analyze the tutor's
response and determine if it
provides actionable guidance to the
student.
Classification Rules:
- Label as "Yes" if the response gives
specific, clear instructions on what
to do next
- Label as "To some extent" if the
response hints at needed action but
lacks specificity
- Label as "No" if the response only
provides the answer without guidance
Remember: Focus on whether the response
guides the student's next steps, not
just whether it's correct.

```

4.3.4 Track 4: Optimized Weighted Ensemble

For Track 4, we developed a streamlined ensemble approach:

1. **Model-Level Weighting:** Global weights for each model applied to all class probabilities
2. **Model-Class Weighting:** Individual weights for each model-class combination (12 weights total)
3. **Threshold Optimization:** Class-specific probability thresholds to address class imbalance
4. **Hyperparameter Optimization:** Optuna-based optimization using macro F1 as the target metric

Table 1: Task 5 Leaderboard: Identity Classification

Rank	Team	Ex. F1	Ex. Acc
1	Phaedrus	0.9698	0.9664
2	SYSUpporter	0.9692	0.9657
3	Two Outliers	0.9172	0.9412
4	JInan_Smart Education	0.8965	0.8940
5	BLCU-ICALL	0.8930	0.8908

4.4 Training Strategy

Our team (Phaedrus) training strategy incorporated several techniques to maximize performance across both tasks:

Cross-Validation: We employed 5-fold Stratified Group K-Fold cross-validation, ensuring dialogues from the same conversation ID remained in the same fold to prevent data leakage while maintaining class distribution.

Hyperparameter Configuration: We used AdamW optimizer with weight decay of 0.003, learning rates ranging from 1e-5 to 3e-5 depending on model size, and OneCycleLR scheduler with maximum learning rate reached at 30% of training steps. For larger models, we implemented gradient accumulation with effective batch sizes of 16-32.

Task-Specific Input Formatting: To maximize ensemble diversity, we designed distinct input templates for different model architectures, each optimized for their specific attention mechanisms and training paradigms:

BERT-family Models (DeBERTa, Longformer, BigBird):

```
[Question] + [SEP] + [R_START] + [
  Response] + [R_END] + [SEP] + [
  Context]
```

Structured format with explicit token boundaries for enhanced attention control

Qwen-2.5 Model:

```
Track 5: Question: [Question]; Answer: [
  Response]; Context: [Context]
Track 4: Question: [Question]; Response:
  [Response]
```

Natural language format optimized for instruction-following capabilities

Zephyr-7B Model:

```
Question: [Question]; Answer: [Response]
```

Parameter-Efficient Fine-tuning: For larger models, we utilized Low-Rank Adaptation (LoRA) with model-specific configurations:

Qwen-2.5 used rank=256/alpha=512 (Track 4) or rank=64/alpha=128 (Track 5), while Zephyr used rank=16/alpha=32. Models were quantized to 4-bit or bfloat16 precision to reduce memory requirements.

Early Stopping and Regularization: We implemented early stopping based on validation macro F1 score with patience of 3 epochs. Dropout rates were set to 0.1 for base models, with multi-sample dropout providing additional regularization through ensemble-like averaging.

5 Experiments and Results

5.1 Experimental Setup

We trained our models using 5-fold cross-validation with early stopping based on validation macro F1 score. Each model was trained for 25 epochs using AdamW optimizer with weight decay of 0.003 and OneCycleLR scheduler.

5.2 Track 5: Tutor Identity Classification Results

Table 2 presents the performance of our identity classification system.

Our Track 5 system achieved a macro F1 score of 0.9596, securing rank 1st 1 in the competition leaderboard. The results demonstrate several key findings:

1. **Cross-Response Context** provides the largest individual contribution, confirming that comparative information between different tutor responses is highly valuable for distinguishing tutor identities.
2. **Pseudolabeling** adds consistent improvement across all classes, with the largest gains for classes with fewer training examples.
3. **Ensemble Diversity** proves crucial, as each model contributes uniquely to final performance.

Model	Val Macro F1	Val Accuracy	LB Macro F1	LB Accuracy
DeBERTa-v3-base	0.8971	0.8901	NA	NA
DeBERTa-v3-large	0.8995	0.8914	NA	NA
Longformer-base	0.8945	0.8865	NA	NA
BigBird-RoBERTa-large	0.8761	0.8671	NA	NA
Qwen-2.5-0.5B	0.8938	0.8869	NA	NA
Zephyr-7B-alpha	0.8811	0.8740	NA	NA
LightGBM meta-model	0.9226	0.9172	0.9250	0.9263
+ Pseudolabeling	0.9585	0.9547	0.9604	0.9619
Final Ensemble	0.9596	0.9560	0.9698	0.9664

Table 2: Track 5 performance on validation set using 5-fold cross-validation and leaderboard results

Table 3: Task 4 Results: Actionability Assessment

Rank	Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
1	bea-jh	0.7085	0.7298	0.8527	0.8837
2	BJTU	0.6992	0.7363	0.8633	0.8940
3	MSA	0.6984	0.7537	0.8659	0.8908
4	lexiLogic	0.6930	0.7162	0.8393	0.8675
5	Phaedrus	0.6907	0.7298	0.8346	0.8656

5.3 Track 4: Actionability Assessment Results

Table 4 presents the performance of our actionability assessment system.

Our Track 4 system achieved a macro F1 score of 0.655, securing 5th Table 3 place on the competition leaderboard. The results reveal:

1. **Model-Class Weighting** outperforms simple model-level weighting, suggesting different models have strengths for different actionability categories.
2. **Instruction Guidance** significantly improves model understanding of actionability criteria.
3. **Middle Category Challenge:** The "To some extent" category shows lower performance, reflecting inherent ambiguity in partial actionability.

5.4 Feature Importance Analysis

Figure 1 shows feature importance from our Track 5 LightGBM meta-model, revealing model-class specialization patterns.

The analysis reveals that different architectures excel at detecting specific tutor identities, validating our multi-model ensemble approach. Each

LLM leaves distinct "fingerprints" detectable by specialized transformer architectures.

6 Discussion

Our comprehensive approach to both tutor identity classification and actionability assessment provides valuable insights into the current state of AI tutoring systems and their relationship to human tutoring effectiveness.

6.1 Cross-Task Insights

The combination of both tasks reveals important patterns:

1. **Identity-Actionability Correlation:** Our analysis suggests that human expert tutors consistently receive higher actionability ratings than most LLMs, indicating that the source of a response correlates with its pedagogical effectiveness.
2. **LLM Differentiation:** Different LLMs show distinct patterns not only in linguistic fingerprints but also in their ability to provide actionable guidance. This suggests that model architecture and training approaches influence pedagogical capabilities.
3. **Detectability vs. Quality:** Despite LLMs' increasing sophistication in generating

Model	Val Macro F1	Val Accuracy	LB Macro F1	LB Accuracy
DeBERTa-v3-small	0.6169	0.7124	NA	NA
DeBERTa-v3-base	0.6262	0.7161	NA	NA
DeBERTa-v3-large	0.6360	0.7112	NA	NA
Qwen-2.5-0.5B	0.6387	0.7205	NA	NA
Model-level weights opt.	0.6536	0.7387	NA	NA
Model-class weights opt.	0.6548	0.7346	0.6836	0.7292
Final Ensemble	0.6551	0.7350	0.6907	0.7298

Table 4: Track 4 performance on validation set using 5-fold cross-validation and leaderboard results

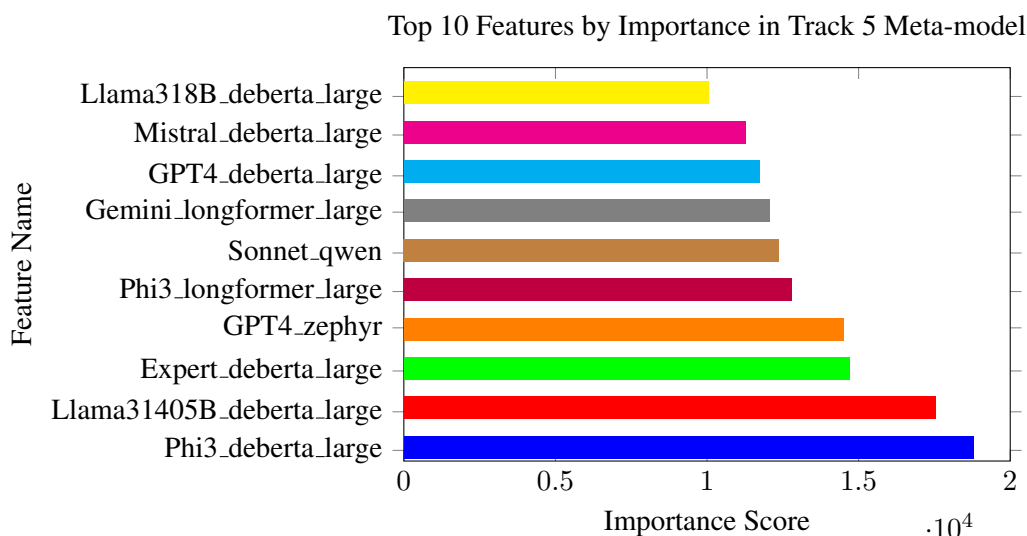


Figure 1: Top 10 feature importance scores showing model-class specialization in tutor identity. Each feature represents how confident a specific transformer architecture is in predicting a particular tutor identity. For example, "Phi3.deberta.large" indicates the DeBERTa-large model's probability output for the Phi3 LLM class classification.

human-like responses, they remain detectable through subtle patterns while showing varying quality in educational effectiveness.

6.2 Technical Contributions

Our methodology contributes several innovations to educational dialogue assessment:

- 1. Cross-Response Context Augmentation:** This technique significantly improves identity classification by providing comparative information, suggesting that tutor identity is best understood in relation to alternative responses.
- 2. Constraint Satisfaction Integration:** The post-processing approach for enforcing unique class assignments demonstrates how task-specific constraints can be integrated into neural classification systems.

- 3. Instruction-Guided Training:** The explicit incorporation of assessment criteria into model input proves effective for actionability evaluation, suggesting broader applications for criterion-based classification tasks.

- 4. Multi-Model Specialization:** Our feature importance analysis confirms that different transformer architectures capture complementary aspects of educational dialogues, supporting diverse ensemble approaches.

6.3 Educational Implications

The findings have significant implications for educational technology:

- 1. Transparency and Accountability:** The ability to reliably identify AI vs. human tutoring responses enables better transparency in educational settings where students may not be aware of AI involvement.

2. **Quality Assurance:** Automated actionability assessment can provide real-time feedback to improve both human and AI tutoring responses, potentially enhancing educational outcomes.
3. **AI Development Guidance:** The identification of specific areas where LLMs fall short in actionability provides clear targets for improving AI tutoring systems.
4. **Hybrid Systems:** Understanding the complementary strengths of human and AI tutors can inform the design of hybrid systems that leverage the best aspects of both.

6.4 Methodological Insights

Our approach reveals several important methodological considerations:

1. **Task Complementarity:** The combination of identity classification and quality assessment provides a more comprehensive evaluation framework than either task alone.
2. **Context Importance:** Both tasks benefit significantly from contextual information, whether through cross-response comparison or instruction guidance.
3. **Ensemble Effectiveness:** Different ensemble strategies (meta-learning vs. weighted voting) prove optimal for different tasks, suggesting that ensemble design should be tailored to specific problem characteristics.
4. **Constraint Integration:** The successful integration of domain constraints (uniqueness) into neural models demonstrates the value of combining symbolic and connectionist approaches.

These findings collectively demonstrate that effective educational dialogue assessment requires sophisticated approaches that consider both the source and quality of responses, with important implications for the development and deployment of AI tutoring systems.

7 Acknowledgments

We would like to thank the organizers of the Pedagogical Ability Assessment of AI-powered Tutors

shared task and the Building Educational Applications (BEA) 2025 workshop for running this competition. We also thank the anonymous reviewers for their insightful and constructive comments, which helped raise the standard of this manuscript considerably.

8 Limitations

While our multi-task approach achieved strong performance on both BEA 2025 Shared Task (Kochmar et al., 2025) tracks, several limitations should be noted:

1. **Domain Specificity:** Our models were trained and evaluated specifically on mathematical tutoring dialogues. Performance may not generalize to other educational domains with different discourse patterns or pedagogical requirements.
2. **Language and Cultural Constraints:** The dataset primarily consisted of English-language dialogues reflecting specific educational contexts. Performance on multilingual or cross-cultural tutoring scenarios remains untested.
3. **Temporal Limitations:** As LLMs continue to evolve rapidly, the distinctive patterns identified by our models may change. Future versions of the same LLMs might exhibit different characteristics, potentially reducing classification effectiveness.
4. **Computational Requirements:** Our approach relies on large transformer models and sophisticated ensemble techniques, requiring significant computational resources that may limit practical deployment in resource-constrained educational environments.
5. **Interpretability Challenges:** While our models achieve high classification accuracy, they provide limited insights into the specific linguistic or pedagogical features that drive classification decisions, making it difficult to extract actionable guidance for improving tutoring responses.
6. **Category Granularity:** The discrete categorization schemes may oversimplify complex phenomena—tutor identity includes many

sub-variations within categories, and actionability might be better represented as a continuum rather than discrete classes.

Future work could address these limitations by expanding to multiple educational domains, developing more efficient architectures, incorporating explainable AI techniques, and exploring the explicit modeling of cross-task relationships. Additionally, longitudinal studies tracking LLM evolution and cross-cultural validation would strengthen the generalizability of these approaches.

References

- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#). *Preprint*, arXiv:2004.05150.
- 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.
- 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.
- Menna Fateen and Tsunenori Mine. 2023. [In-context meta-learning vs. semantic score-based similarity: A comparative study in Arabic short answer grading](#). In *Proceedings of ArabicNLP 2023*, pages 350–358, Singapore (Hybrid). Association for Computational Linguistics.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. In *EMNLP*.
- Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. [How close is chatgpt to human experts? comparison corpus, evaluation, and detection](#). *Preprint*, arXiv:2301.07597.
- 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.
- Hiroshi Inoue. 2020. [Multi-sample dropout for accelerated training and better generalization](#). *Preprint*, arXiv:1905.09788.
- Ekaterina Kochmar, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Nitin Madnani, Anaïs Tack, Victoria Yaneva, Zheng Yuan, and Torsten Zesch, editors. 2022. [Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications \(BEA 2022\)](#). Association for Computational Linguistics, Seattle, Washington.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- 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.
- 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.
- Qwen, :, 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, Jiayi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- 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](#). In *Proceedings of the 15th International Conference on Educational Data Mining*, pages 522–529, Durham, United Kingdom. International Educational Data Mining Society.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. [Zephyr: Direct distillation of lm alignment](#). *Preprint*, arXiv:2310.16944.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. [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.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. [Big bird: Transformers for longer sequences](#). *Preprint*, arXiv:2007.14062.

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

Emergent Wisdom at BEA 2025 Shared Task: From Lexical Understanding to Reflective Reasoning for Pedagogical Ability Assessment

Raunak Jain^{1*} and Srinivasan Rengarajan¹

¹Intuit

{raunak_jain1, srinivasan_rengarajan}@intuit.com

Abstract

For the BEA 2025 shared task on pedagogical ability assessment, we introduce **LUCERA** (Lexical Understanding for Cue Density–Based Escalation and Reflective Assessment), a rubric-grounded evaluation framework for systematically analyzing tutor responses across configurable pedagogical dimensions. The architecture comprises three core components: (1) a rubric-guided large language model (LLM) agent that performs *lexical and dialogic cue extraction* in a self-reflective, goal-driven manner; (2) a cue-complexity assessment and routing mechanism that sends high-confidence cases to a fine-tuned T5 classifier and escalates low-confidence or ambiguous cases to a reasoning-intensive LLM judge; and (3) an *LLM-as-a-judge* module that performs structured, multi-step reasoning: (i) generating a domain-grounded reference solution, (ii) identifying conceptual, procedural and cognitive gaps in student output, (iii) inferring the tutor’s instructional intent, and (iv) applying the rubric to produce justification-backed classifications. Results show that this unique combination of LLM powered feature engineering, strategic routing and rubrics for grading, enables competitive performance without sacrificing interpretability and cost effectiveness.

1 Introduction

High-quality formative feedback is a cornerstone of effective learning: timely, specific guidance helps learners close knowledge gaps, consolidate correct mental models, and sustain motivation (Anderson et al., 1995; Hattie and Timperley, 2007). Yet providing rich feedback at scale remains difficult. The BEA-2025 Shared Task (Kochmar et al., 2025) tackles this challenge by pairing a learning-science-grounded evaluation taxonomy (Maurya et al., 2025) with MRBENCH (Maurya et al., 2025),

a benchmark that fuses math-centric tutoring dialogues from MATHDIAL (Macina et al., 2023) and BRIDGE (Wang et al., 2024b). The competition assesses four pedagogically salient dimensions—*Mistake Identification* (MI), *Mistake Location* (ML), *Providing Guidance* (PG), and *Actionability* (ACT)—thereby offering a unified, standardised test-bed for measuring the pedagogical competence of AI tutors.

While we participate in the shared task, our goal extends beyond leader-board performance (see 7 for more details). We introduce **LUCERA** (Lexical Understanding for Cue density–based Escalation and Reflective Assessment), a novel hybrid architecture that unifies fast lexical heuristics, confidence-aware routing, and reasoning capabilities of large-language-models (LLMs). **LUCERA**, positioned as a general research contribution, demonstrates how an adaptive cascade can deliver interpretable, scalable, rubric-faithful evaluation—attributes that matter both inside and outside competition settings.

Existing approaches to pedagogical-quality assessment occupy two extremes. At one end, rule-based cue extractors offer transparency and speed but falter when feedback is implicit or domain-specific (Lehman et al., 2019; Wang et al., 2020; Wollny et al., 2021; Macina et al., 2023). At the other, rubric-grounded LLM judges achieve broad coverage yet impose high computational cost and, when used indiscriminately, act as opaque monoliths that are hard to audit (Liu et al., 2023b; Maurya et al., 2025; Tack et al., 2023). Bridging these extremes, stepwise chain-of-thought (CoT) verification recovers subtle pedagogical intent but further magnifies latency and cost (Daheim et al., 2024; Wang et al., 2024b; Jain, 2025).

LUCERA orchestrates these complementary paradigms in a three-stage pipeline. A lightweight lexical-cue extractor provides instant, interpretable signals; a complexity-aware router allocates re-

*Corresponding author: raunak_jain1@intuit.com

sponses to either a heuristic XGBoost scorer, a fine-tuned T5 classifier, or a reflective CoT judge; and a final rubric-aligned verdict is produced only in low confidence scenarios. All LLM-based tasks in this work were performed using Qwen/Qwen2-1.5B-Instruct (Yang et al., 2024). This design achieves a 2.4× throughput gain over blanket LLM judging on the BEA-2025 dev set while maintaining rubric fidelity. Beyond the task, we argue that *LUCERA* offers a principled template for scaling LLM-based pedagogical quality assessment wherever feedback quality, cost, and transparency must be balanced.

The remainder of this paper is organised as follows: Section 2 surveys prior work on pedagogical-quality assessment, LLM judges, verification pipelines, and intelligent routing; Section 3 describes *LUCERA*'s architecture; Section 4 and Section 5 detail the feature extraction and classification components; Section 6 explains the reflective LLM judge; Section 7 reports empirical results; and Section 7 concludes with limitations and directions for future research.

2 Related Works

Surface-level cue extraction. Early work framed pedagogical quality as a pattern-recognition problem: if a tutor turn contains directive verbs (*try*, *consider*), contrastive discourse markers (*however*, *because*), or worked-example fragments, it likely advances learning (Lehman et al., 2019; Wang et al., 2020). Rule-based and linear classifiers built on these *lexical cues* offered millisecond latency and clear rationales, and they continue to power production intelligent tutoring systems (Bringula and Basa, 2018). Nevertheless, large corpus studies show cue sparsity and STEM-specific jargon severely degrade their recall and domain transferability (Wollny et al., 2021; Macina et al., 2023).

LLM-as-a-Judge and Confidence based Cascades. The arrival of GPT-4-class models sparked a shift to *rubric-grounded* prompting: an LLM reads a turn and scores each dimension directly (Liu et al., 2023b). Frameworks such as LLM-RUBRIC formalise this practice and report sizeable gains across open-ended tasks (Xia et al., 2025). Yet *unconditional* LLM judging inflates inference cost (Jung et al., 2025; Schuster et al., 2022), produces verbose rationales of uneven quality (Saito et al., 2023; Ohi et al., 2024; Wang et al., 2024a), and can hallucinate additional rubric cri-

teria (Li et al., 2023). Recent selective evaluation frameworks provide provable guarantees of human agreement while maintaining high coverage (Jung et al., 2024), achieving better human alignment than monolithic LLM judges while being substantially more cost-effective. These findings strongly motivate the search for *selective* depth in LLM-based evaluation.

Stepwise verification and reflective reasoning.

Recent studies introduce a verification stage in which an LLM first generates a reference solution and then aligns it with the learner's work before labelling (Daheim et al., 2024; Wang et al., 2024b). Such *chain-of-thought* (CoT) pipelines help identify correct pedagogical strategies by boosting understanding of student gaps (Jain, 2025). Complementary efforts build testbeds (e.g., TutorGym) and benchmarks that grade the fidelity of reasoning chains (Li et al., 2025; Jacovi et al., 2024). However, each additional reasoning step multiplies latency and cost, making blanket deployment impractical at classroom scale.

Intelligent routing and hybrid cascades.

Outside education, researchers mitigate the cost-accuracy trade-off by cascading small and large models, deferring only hard instances. Contemporary confidence-tuned cascades (Xu and McAuley, 2022), cascade-aware training (Zhang et al., 2024), and calibrated ensemble policies (Wagner et al., 2024) achieve 1.5–3× speed-ups without loss of accuracy. *Educational NLP*, by contrast, has yet to embrace hybrid routing: state-of-the-art graders for assignments (Chiang et al., 2024), short-answer scoring (Chang and Ginter, 2024), and essay evaluation (Latif and Zhai, 2024; Jiang and Bosch, 2024) all deploy a single, monolithic LLM without confidence-based deferral. Bridging this gap remains an open opportunity for future assessment systems.

Summary and open gap. The literature thus presents three partially solved challenges—speed (cue extractors), depth (CoT verifiers), and transparency (rubric-grounded judging)—addressed in isolation. No prior system unifies them under a single rubric while allocating compute *proportionally* to instance difficulty. By integrating cue density, calibrated confidence, and stepwise verification into one adaptive cascade, *LUCERA* fills this gap and provides the first cost-aware, rubric-consistent pipeline for tutor-response evaluation.

3 System Overview

LUCERA is a three-stage pipeline designed to assess the pedagogical quality of tutor responses. The system processes tutor responses through the following components:

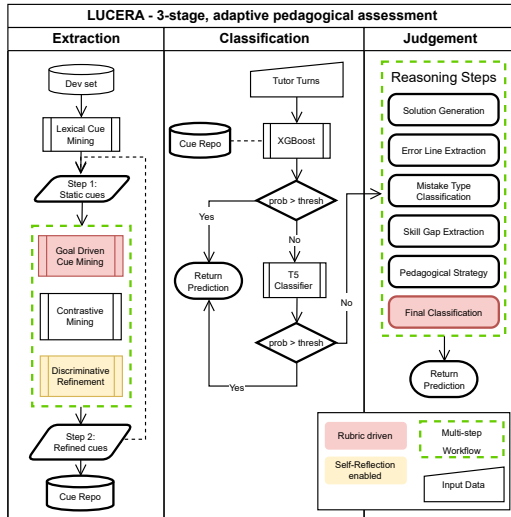


Figure 1: Overview of *LUCERA*'s three-stage pipeline: (1) Rubric-Guided, goal driven lexical cue extractor identifies pedagogical features, (2) XGBoost and T5 based classifiers to solve for feature rich scenarios (3) Multi-step LLM judge for ambiguous and complex evaluation scenarios.

1. Rubric-Guided Lexical Cue **Extraction**: Identifies lexical and dialogue cues aligned with rubric dimensions via a self-reflective LLM agent. Maps directly to pedagogical criteria, maintaining interpretability and transparency throughout the feature extraction process.
2. XGBoost or Seq2Seq based **Classification**: Routes cases based on cue density and confidence. Deploys lightweight XGBoost or T5 models for efficient assessment of high-confidence cases with clear lexical patterns.
3. Step-wise LLM **Judgement**: Handles ambiguous or complex cases through multi-step reasoning. Generates reference solutions, identifies student knowledge gaps, and applies rubric criteria to deliver in-depth pedagogical analysis.

The rubric schema grounds both the cue extractor and the LLM judge, providing uniform evaluation criteria. After extracting cues from a tutor

response, the system applies confidence-based routing: high-certainty cases proceed to a lightweight T5 classifier, while ambiguous ones are escalated to a reasoning-intensive LLM judge, conserving computation without sacrificing pedagogical depth.

4 Rubric-Guided Lexical Cue Extraction

We systematically developed a comprehensive feature taxonomy spanning multiple linguistic levels to classify tutor responses across pedagogical dimensions (MI, ML, PG, ACT), enabling fine-grained analysis of pedagogical signals in tutorial discourse.

4.1 Step 1: Linguistically-Grounded Feature Engineering

Lexical Cues

These cues provide shallow yet effective insights into semantic content and form of tutor responses:

- **Volumetric Features**: Basic text-level metrics including word, character, and sentence counts (Yang, 2024) serve as proxies for response depth. Low word counts may negatively correlate with PG and ACT due to insufficient detail.
- **Question Words**: Presence of interrogatives (e.g., "what," "why," "how") identified via counting pre-defined question words (Demszky et al., 2018), hypothesized to positively correlate with PG and ACT by signaling engagement and elaboration.
- **Feedback Words**: Terms like "correct," "mistake," or "however"—extracted using sentence-level sentiment or discourse tagging (Negi and Buitelaar, 2015)—expected to signal MI by indicating evaluative judgment. For example, "You're close, but remember" is a definite feedback phrase.
- **Directive Verbs**: Instructional verbs (e.g., "calculate," "explain," "solve") extracted using POS tagging and grammatical mood detection (Cohen et al., 2004), often implying actionability. For example, "Let's calculate that," "can you think of a way to calculate?" are all instructional phrases.
- **Hedging Words**: Words like "maybe," "might," or "could" introducing nuance or tentativeness, often associated with PG and ACT

by softening directive tone (Deng et al., 2025). For example, "I think maybe you need one more step," "maybe we can use a hundreds chart or count up" demonstrate this.

- **Pronoun Ratios:** Ratios of second-person (you/your) to first-person (I/my) pronouns indicating student-centeredness or tutor-centeredness, relevant for PG and ACT (Qureshi and Strube, 2022). **Student-centeredness** refers to responses focusing on engaging students directly, guiding actions, or providing feedback, characterized by higher frequency of second-person pronouns. **Tutor-centeredness** reflects tutor’s perspective, explanations, or insights, marked by higher frequency of first-person pronouns. Higher ratio of second-person to first-person pronouns suggests student-centric approach emphasizing direct instruction, while lower ratio indicates tutor-centric approach sharing tutor’s reasoning. For example, "but remember your initial calculation," "but actually, you did add Kylie’s 3 towels" are student-centered responses indicating PG.

4.1.1 Syntactic Complexity

Syntactic complexity is measured via average sentence length and subordinate clause density using dependency parsing (Crossley and McNamara, 2022). High complexity may hinder comprehension, potentially impacting PG and ACT despite informative content.

4.1.2 Pragmatic and Discourse Cues

These features capture pragmatic and contextual dimensions:

- **Discourse Markers:** Cues such as "however," "for example," or "but" indicating relationships between discourse units, helping differentiate between elaboration for PG and contradiction for MI (Dai and Huang, 2018).
- **Conversational Uptake:** Semantic alignment of tutor’s response with preceding turns, computed using pre-trained dialogue embedding models (Demszky et al., 2021). High uptake suggests relevance and coherence, especially for PG.
- **Pedagogical Intent:** Pre-trained NLI models capturing latent pedagogical intent beyond

surface features, by computing the 3-way softmax probabilities (entailment, contradiction, neutral) between tutor responses (premise) and intent descriptions (hypothesis) (Reimers and Gurevych, 2019). The entailment probability values [0-1] directly serve as feature weights, enabling nuanced quantification of pedagogical intents like supportiveness and elaboration.

- **Dialogue Act (DA) Classification:** Responses categorized into high-level DAs (e.g., 'Correction,' 'Hint,' 'Instruction') using pre-trained models (Noble and Maraev, 2021), serving as semantically rich signals—e.g., 'Correction' aligns with MI/ML while 'Instruction' relates to PG and ACT.

4.1.3 Feature Encoding Summary

Feature encoding employs a dual representation strategy: (1) numeric quantification (counts for volumetric features, pronoun ratios) and (2) TF-IDF vectorization (Salton and Buckley, 1988) with category-specific lexicons (feedback, directive, hedging words, discourse markers). Pedagogical intent features leverage NLI entailment probability values [0-1] as continuous feature weights. This complementary approach integrates statistical surface patterns with semantic-level analysis to capture both explicit and implicit pedagogical signals.

4.2 Step 2: LLM-Driven Discriminative Feature Refinement

Our approach employs a multi-stage pipeline that transforms initial lexical features into discriminative, contextually-validated pedagogical indicators. This process ensures alignment with assessment rubrics through progressive refinement, as illustrated in Figure 1 (*Extraction - Refined Cues*):

1. **Goal-Directed Feature Extraction:** LLM analyzes conversation data to identify discriminative features through an iterative, objective-oriented process guided by the initial seed features from 4. The extraction process leverages a T5-based (Raffel et al., 2020a) encoder-decoder framework fine-tuned on pedagogical conversations.
2. **Adversarial Refinement:** Features undergo validation against contradictory examples from other conversations, enabling the

LLM to eliminate spurious correlations and strengthen genuinely predictive indicators.

3. **Lexical Cue Repository Update:** Validated features are populated back into the cue repository, and steps 1-3 are repeated until no new features are found, ensuring a comprehensive and stable set of pedagogically discriminative features.

This methodology produces feature sets that transcend mere textual presence to capture pedagogical quality signals validated against both assessment criteria and challenging counterexamples.

4.3 Feature EDA Summary

We conducted systematic exploratory data analysis on development set tutor responses to quantify relationships between engineered features and pedagogical dimensions (MI, ML, PG, ACT) using Pearson correlation coefficients and distributional statistics, as detailed in Table 1. Key abbreviations include: **Vol** (Volumetric Features), **Ques** (Question Words), **Fdbk** (Feedback Words), **DV** (Directive Verbs), **Hed** (Hedging Words), **ProR** (Pronoun Ratios), **Y/N/TSE** (Yes/No/To Some Extent), **Read** (Readability score based on `flesch_reading_ease` (Flesch, 1948)), and **H/M/L** (Higher/Medium/Lower correlation trend across response categories).

4.3.1 Feature Influence based on Pearson Correlation

Table 1 summarizes key feature influences on pedagogical dimensions, where H/L/M indicates High/Low/Medium influence for Yes/No/TSE classes respectively. Analysis of these patterns reveals several critical insights:

- **Volumetric Features** (H/L/L or H/L/H): Longer responses correlate with effective tutoring across dimensions (Chi et al., 2001; Ward et al., 2011), with verbosity particularly important for actionability where detailed guidance enables student progress (VanLehn, 2011).
- **Question Words** (variable patterns): Strong association with ML (H/L/H) shows questioning is essential for modeling learning; moderate impact on MI (M/L/H) reveals interrogatives' dual purpose in challenging misconceptions and guiding reflection (Graesser et al.,

2010; VanLehn et al., 2006). TSE pattern (H) suggests questions create partial pedagogical value (Chen et al., 2011). Radar analysis (Figure 3) confirms Question Words heavily influence ACT while moderately affecting PG and MI across both TSE and "No" classifications.

- **Pronoun Ratios:** Reveals dimension-specific strategies—PG/MI benefit from tutor-centric language (L/H for "Yes"/"No") where expert explanation is valued; ACT/ML favor student-centric approaches (H/L) positioning students as active participants (Nystrand and Gamoran, 1997; Mercer and Littleton, 2009; Biber and Gray, 2006). N-gram analysis (Figure 2) shows distinctive phrases like "looks like you" and "remember that" strongly correlate with PG.
- **Feedback & Directive Words:** Inverse patterns between feedback (L/H/L) and directives (H/L/M) highlight tension between evaluation and instruction (Shute, 2008; Hattie and Timperley, 2007). Combined with hedging patterns, this suggests effective tutoring balances definitive guidance with tentative suggestion (Rowland, 2002; Mackiewicz and Thompson, 2010). Readability scores and Feedback markers most heavily influence MI classification as shown in Figure 3.
- **Discourse Context:** Discourse markers strongly influence MI (H/L/L) and ML (H/L/H) (Fraser, 1999; Sanders et al., 2000). Contrasting readability patterns (MI: M/L/H vs. others: M/H/L) suggest misconception identification benefits from accessible language while model learning sometimes requires complex formulations (McNamara et al., 2010; Crossley et al., 2017). Action-oriented phrases ("closer look," "look at") strongly correlate with ACT dimension (Figure 2). TSE class presents unique classification challenges with subtle linguistic markers and mixed signals—often providing information without prompting direct action.

These patterns reflect pedagogical trade-offs: correction versus guided discovery (Hmelo-Silver et al., 2006), authoritative versus collaborative stance (Scott et al., 2002), and comprehensive explanation versus concise instruction (Wittwer and Renkl, 2010). Differential patterns validate our

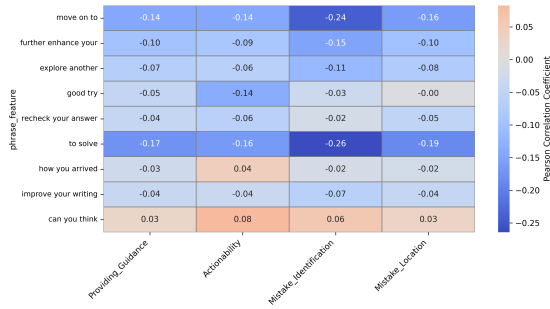


Figure 2: Word Correlation with Pedagogical Dimensions (Yes)

taxonomy’s ability to capture distinct tutoring aspects, while shared patterns highlight fundamental qualities of effective pedagogical communication.

Feature	MI	ML	PG	ACT
Vol	H/L/L	H/L/H	H/L/H	H/L/H
Ques	M/L/H	H/L/H	H/L/M	H/L/M
Fdbk	M/L/L	L/H/L	L/H/L	L/H/L
DV	H/L/M	M/H/M	M/H/M	H/L/M
Hed	H/L/H	M/H/H	M/H/M	H/L/H
ProR	M/H/L	L/H/H	L/H/M	H/L/H
DM	H/L/L	H/L/H	M/L/M	M/L/M
Read	M/L/H	M/H/L	M/H/L	M/H/L

Table 1: Summary of Feature Influence Based on Correlations (H: High, L: Low, M: Medium) for classes Yes/No/TSE.

5 Model Cascade, Confidence-Based Routing, and Task Submission

Based on the extracted refined cues from Section 4, we first train an XGBoost model as a baseline to cover cases where lexical coverage is high. Once we identify lack of lexical coverage, we escalate the classification process to a T5 transformer architecture with a generative classification task instruction. We detail the baseline and T5 model architectures, training methodology etc in the following sections. The evaluation metrics are: exact macro F1 score (Ex. F1), exact accuracy (Ex. Acc), lenient macro F1 score (Len. F1), and lenient accuracy (Len. Acc).

5.1 Stage 1: XGBoost + Lexical Cues as Baseline

For a baseline, we train a multi-label (Yes/No/TSE) multi-class classification model using XGBoost (Chen and Guestrin, 2016) with a 70/30 train-val split. Hyperparameter tuning was performed using cross-validation, focusing on key



Figure 3: Normalized Combined Radar Chart by Feature Group(TSE)

parameters such as `max_depth`, `learning_rate`, `n_estimators`, `min_child_weight`, and `subsample`. Table 2 presents the performance on validation dataset of the best run (XGBoost is considerably better than all Yes (majority class for all labels) as a baseline).

Task	Ex.Acc	Ex.F1	Len.Acc	Len.F1
MI	0.71	0.64	0.81	0.73
ML	0.67	0.66	0.85	0.72
PG	0.68	0.66	0.84	0.71
ACT	0.75	0.71	0.81	0.73

Table 2: XGBoost Performance Metrics by Pedagogical Dimension

5.1.1 Feature Impact and Model Limitations:

While the XGBoost model performed well across different pedagogical dimensions, several limitations were identified for improvement in subsequent iterations:

- **Syntactic Complexity and Question-Related Features:** Complex syntactic structures, such as nested clauses or subordinate sentences, can confuse the model. For example, "While it seems correct, you might want to double-check the calculation" may be misclassified as *Yes* for MI due to ambiguous framing. Additionally, interrogative cues are essential for classifying ACT and PG, but rhetorical questions can mislead the model. For instance, "Do you think this is correct?" could be interpreted as actionable, despite expressing doubt.

- **Lexical Features and Semantic Nuance:** Lexical cues, such as keyword spotting, can lead to errors when words appear in unexpected contexts. For instance, "You're doing great! But remember" is encouraging but points to an approach to guide the student toward the correct answer.
- **Pronoun Usage and Intent Ambiguity:** Shifts between tutor-centric ("I/my") and student-centric ("you/your") language cause inconsistent classification. A statement like "You could explain it better" may be classified as a *Yes* for ACT, whereas a similar structure with "I" might be a strong *Yes* for PG. Detecting perspective shifts remains challenging.

5.2 Stage 2: Instruction-Based Seq2Seq Classification

Model Setup: We fine-tune T5-Base (approximately 250M parameters) for instruction-based classification across four pedagogical dimensions: MI, ML, PG, ACT. Each dimension is specified using a distinct prompt prefix. This approach leverages T5's encoder-decoder architecture, with a 512-token context and unified text-to-text pretraining, facilitating efficient and accurate classification (Raffel et al., 2020b; Qorib et al., 2024).

Prompt Template: The model generates a single-token prediction $y \in \{\text{yes, no, maybe}\}$. Each prompt follows this structure:

```
[PEDAGOGICAL_DIMENSION]
[LEXICAL_CUES] <lexical cues>
[TUTOR_TURNS] <concatenated tutor turns>
Output: {yes, no, maybe}
```

Example (Mistake Location):

```
[Providing Guidance]
[LEXICAL_CUES] let us use, what was
[TUTOR_TURNS] ok let us use the information
to help us what was her gross revenue this week?
Output: maybe
```

Loss Function and Evaluation Metrics: We minimize the single-token cross-entropy over our dataset \mathcal{D} :

$$\mathcal{L}(\theta) = - \sum_{(x,y) \in \mathcal{D}} \log p_{\theta}(y | x),$$

where y is the correct label and decoding is constrained to one step (greedy, $\text{max_length} = 1$).

Training Details: We fine-tuned T5-Base via HuggingFace Transformers on an Apple M3 Mac (no GPU) using 70%/30% train/eval splits. Preprocessing involved excluding student turns and initial problem-introduction turns, concatenating remaining tutor turns, and truncating leftmost tokens when exceeding the 512-token limit. Training used batch size 8, AdamW optimizer (weight decay 0.01, LR = 3×10^{-5} with 10% steps linear warmup), and ran for 5 epochs with early stopping (patience=2, dropout=0.1). Decoding was performed greedily with evaluation via dev loss.

Why T5-Base? We selected T5-Base for its empirical and architectural advantages: superior classification performance, with Flan-T5 variants consistently outperforming decoder-only models on GLUE, SuperGLUE, and word-sense disambiguation tasks while matching GPT-3.5 in few-shot settings (Papadopoulos et al., 2024; Liu et al., 2023a); multi-task pre-training on diverse tasks equipping it with transferable NLP skills that generalize without separate heads (Raffel et al., 2020b; Liu et al., 2023a); hardware efficiency at 250M parameters, comfortably running on modest hardware with 512-token input handling and position embeddings preventing truncation issues (Scao et al., 2022; Hu et al., 2023); parameter-efficient fine-tuning via Adapter and LoRA methods matching larger models on MNLI, QNLI, and SST-2 (Hu et al., 2023); and low-data robustness requiring fewer labeled examples to achieve competitive scores compared to masked-encoder counterparts (Papadopoulos et al., 2024; Liu et al., 2023a).

5.3 Confidence-Based Routing Strategy

We implement a probability-based cascade to balance computational efficiency with classification accuracy. Many tutor utterances lack explicit lexical cues that our XGBoost baseline relies on, necessitating a dynamic routing approach.

For input x , we define probability vectors:

- XGBoost: $p_{\text{xgb}}(x) \in [0, 1]^C$ (sigmoid activations)
- T5: $p_{\text{t5}}(x) \in [0, 1]^C$ (softmax over {yes, no, maybe})

The cascade operates in three stages:

1. Route inputs through XGBoost. Accept prediction if $\max_c p_{\text{xgb}}(x)_c \geq \tau_1$.
2. If $\max_c p_{\text{xgb}}(x)_c < \tau_1$, escalate to T5. Accept if $\max_c p_{\text{t5}}(x)_c \geq \tau_2^{(c)}$ for any class c .

Algorithm 1: Learn class-specific precision cutoffs & coverage

Input: Validation set V , classes \mathcal{C} , model probabilities $p_c(x)$ for each class c , true labels $y(x)$, target precisions $\{\alpha_c\}$, start threshold τ_0 , step size δ , upper bounds $\{U_c\}$

Output: Class-wise thresholds $\{\tau_c^*\}$ and coverages $\{\gamma_c^*\}$

```
foreach class  $c \in \mathcal{C}$  do
   $N_c \leftarrow |\{x \in V : y(x) = c\}|$ ;
   $\tau \leftarrow \tau_0$ ,  $\tau_c^* \leftarrow U_c$ ,  $\gamma_c^* \leftarrow 0$ ;
  while  $\tau \leq U_c$  do
     $S \leftarrow \{x \in V : p_c(x) \geq \tau\}$ ;
    if  $|S| = 0$  then
      break
    prec  $\leftarrow \frac{|\{x \in S : y(x) = c\}|}{|S|}$ ;
    if prec  $\geq \alpha_c$  then
       $\tau_c^* \leftarrow \tau$ ;
       $\gamma_c^* \leftarrow |S|/N_c$ ;
       $\tau \leftarrow \tau + \delta$ ;
    else
      break;
return  $\{(c, \tau_c^*, \gamma_c^*) \mid c \in \mathcal{C}\}$ 
```

3. If T5’s confidence is insufficient, defer to an LLM judge.

Thresholds τ_1 for XGBoost and $\tau_2^{(c)}$ for T5 classes are learned on held-out data using Algorithm 1 to guarantee $\geq 95\%$ precision while maximizing coverage. In our experiments, stages 1 and 2 combined to produce 65-70% of predictions with required confidence, with the remaining 30-35% escalated to the LLM judge (discussed in the following section). See Table 3 for the learned thresholds and coverage at each stage.

6 Step-wise LLM-as-a-Judge

When both our lexical + XGBoost baseline and T5 classifier fall below confidence thresholds, we escalate to a multi-step "LLM-as-a-Judge" for final pedagogical-quality classification. In our dev-set evaluation, 31-34% of conversations across each dimension were escalated to the judge.

1. Solution Reasoning Pathway Generation: The judge prompts the LLM to generate a step-by-step expert solution for the given problem,

Stage	Dimension	Thresholds (Yes/No/TSE)	Coverage at Stage
XGBoost	ML	0.85 / 0.45 / 0.55	0.38
	MI	0.82 / 0.48 / 0.52	0.35
	PG	0.88 / 0.42 / 0.58	0.32
	ACT	0.86 / 0.45 / 0.55	0.36
T5	ML	0.80 / 0.45 / 0.55	0.86
	MI	0.78 / 0.42 / 0.58	0.85
	PG	0.82 / 0.48 / 0.52	0.62
	ACT	0.81 / 0.45 / 0.55	0.60

Table 3: Learned thresholds for Yes/No/TSE classes and coverage percentages at each stage for each pedagogical dimension

establishing a reference reasoning pathway against which to align the student’s response (Wei et al., 2022; Daheim et al., 2024; Jain, 2025). This includes parsing the problem, identifying relevant concepts, applying them systematically, and verifying the final result.

2. Error Extraction: The judge isolates the precise span where student reasoning diverges from the expert chain—this concrete "mistake locus" anchors all downstream diagnostic steps (Daheim et al., 2024; Macina et al., 2023). The goal is solely to extract and localize the deviation.

3. Mistake Classification: The mistake is mapped to a structured taxonomy enabling standardized reasoning about pedagogical strategies (Macina et al., 2023). Categories include conceptual errors, procedural/arithmetic errors, misapplied formulas, comprehension errors, and logical breakdowns in multi-step reasoning (Macina et al., 2023; Wang et al., 2024b; Daheim et al., 2024).

4. Skill Gap Mapping: Based on the mistake classification, the judge infers the underlying cognitive skill gap (Jain, 2025), referencing Bloom’s revised taxonomy (Anderson et al., 2001; Krathwohl, 2002). This includes gaps in: Remember (recalling facts), Understand (grasping concepts), Apply (executing procedures), Analyze (breaking down structure), Evaluate (judging correctness), and Create (developing alternate methods).

5. Last Tutor Turn Strategy Identification: Conditioned on the diagnosed cognitive gap, the judge infers the most probable pedagogically aligned instructional strategy (Macina

et al., 2023; Wang et al., 2024b) which the last tutor turn most likely employed. This may include focus questions, probing questions, worked examples, hints, or problem simplification.

6. **Final Classification:** Integrating all intermediate steps along with the inferred instructional strategy employed by the tutor’s last turn, the judge produces a final classification (Yes/No/TSE) for the last tutor turn according to the BEA 2025 Shared Task dimensions (Kochmar et al., 2025). The turn is evaluated for each of the following dimensions: (1) *Mistake Identification*, (2) *Mistake Location*, (3) *Providing Guidance*, and (4) *Actionability*.

Example Judge Output for PG:

Conversation History: Tutor: Hi, could you please provide a step-by-step solution for the question below? Tyson decided to make muffaletta sandwiches for the big game. Each sandwich required 1 pound each of meat and cheese and would serve 4 people. There would be 20 people in total watching the game. The meat cost \$7.00 per pound and the cheese cost \$3.00 per pound. How much money would he spend on the meat and cheese to make enough sandwiches to serve 20 people? Student: To serve 20 people, Tyson needs to make $20/4 = 5$ sandwiches. Each sandwich requires $1+1 = 2$ pounds of meat and cheese...

Extracted Error: Each sandwich requires $1+1 = 2$ pounds of meat and cheese.

Mistake Type: *Right-idea*. The student has the right idea but inaccurately combines meat and cheese quantities into one, leading to a misapplied calculation.

Skill Gap: *Analyze and decompose independent components*. The student understands facts and unit costs but fails to reason about meat and cheese as distinct cost components.

Recommended Strategy: *Provide a hint*. The tutor asks a guiding question to nudge the student to recalculate the meat cost independently, prompting correction without explicit error labeling.

Judge Classification: *Yes*. The tutor turn offers appropriate scaffolding to guide the next step in solving the problem.

6.1 Dev-Set Escalation Impact

Evaluating 30 examples across all rubric dimensions, the LLM Judge reduced classification errors by 50%–60% compared to our T5 baseline, achieving Macro F1 scores above 75% in three tracks and 83.3% in one (Table 4).

Extrapolating these results to hybrid system performance with judge escalation on 30%–35% of

low-confidence cases (**Hypo-Full** column), projections indicate that selective escalation can substantially bridge the gap to top-performing systems.

Track	Top	T5-subm	Judge-30	Hypo-Full
1	71.81	61.0	83.3	67.69
2	59.8	47.7	76.6	56.37
3	58.3	49.0	73.3	56.29
4	70.9	56.6	76.6	62.6

Table 4: Per-track F1: **Top** = best shared-task model; **T5-subm** = T5 model submitted results; **Judge-30** = LLM on 30 escalated dev cases; **Hypo-Full** = simulated performance assuming judge intervention on 30–35% of cases.

7 Submission Results and Analysis

Our team, **Emergent Wisdom**, participated in tracks 1 to 4 based on the architecture described in 3. The metrics and ranking of our best submission, according to the official leaderboard¹, is shown in Tables 5 and 6, which also contrast our Stage 1–2 (router + encoder–decoder) results on the test set against the top shared-task systems. Δ indicates (Ours – Top).

Tr	Top		Ours			Δ	
	Acc	F1	Acc	F1	Rank	Acc	F1
1	94.6%	89.6	93.2%	88.0	21	–1.4	–1.6
2	86.3%	83.9	78.9%	74.4	15	–7.4	–9.5
3	81.9%	78.0	77.3%	69.2	24	–4.6	–8.8
4	88.4%	85.3	80.5%	77.8	30	–7.8	–7.5

Table 5: Lenient metrics performance (Stages 1–2 on test set; Δ = Ours – Top).

Tr	Top		Ours			Δ	
	Acc	F1	Acc	F1	Rank	Acc	F1
1	86.2%	71.8	85.5%	61.0	34	–0.8	–10.8
2	76.8%	59.8	71.9%	47.7	25	–4.9	–12.1
3	66.1%	58.3	61.0%	49.0	21	–5.1	–9.3
4	73.0%	70.9	66.4%	56.6	22	–6.6	–14.3

Table 6: Exact metrics performance (Stages 1–2 on test set; Δ = Ours – Top).

As demonstrated in section 6.1, our analysis reveals that strategic reliance on the judge component for complex cases enables performance within 1-2% macro F1-score of the top-performing systems without increasing computational needs for the whole dataset, suggesting the potential for competitive results based on intelligent routing.

¹<https://sig-edu.org/sharedtask/2025#results>

Limitations

Despite strong performance, our cascade approach faces several limitations: T5-Base’s 512-token context window restricts processing of longer tutoring sessions; both models struggle with ambiguous utterances serving multiple pedagogical functions; performance suffers on underrepresented classes like the "maybe" classification; confidence-based routing relies on carefully tuned thresholds; and analyzing only tutor turns misses important student context. Future work should explore larger context windows, multi-label classification, and more sophisticated conversational modeling.

References

- John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Sharon Pelletier. 1995. Intelligent tutoring systems. In *Computer Science and Artificial Intelligence Conference*. Springer.
- Lorin W. Anderson, David R. Krathwohl, Peter W. Airasian, Kathleen A. Cruikshank, Richard E. Mayer, Paul R. Pintrich, James Raths, and Merlin C. Wittrock. 2001. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives*. Longman.
- Douglas Biber and Bethany Gray. 2006. *University Language: A Corpus-Based Study of Spoken and Written Registers*. John Benjamins Publishing Company, Amsterdam.
- Rex P. Bringula and Ryan S. Basa. 2018. Effects of prior knowledge in mathematics on learner–interface interactions in a learning-by-teaching intelligent tutoring system. In *Proceedings of the 26th International Conference on Computers in Education*, pages 25–30. Asia-Pacific Society for Computers in Education.
- 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, Vancouver, Canada. AAAI Press.
- Guanliang Chen, Jie Yang, and Claudia Hauff. 2011. Studying effective tutoring strategies in programming moocs. In *Proceedings of the 6th ACM Conference on Learning at Scale*, pages 1–10. ACM.
- 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*. ACM.
- Micheline T. H. Chi, Stephanie A. Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G. Hausmann. 2001. Learning from human tutoring. *Cognitive Science*, 25(4):471–533.
- Cheng-Han Chiang, Wei-Chih Chen, Chun-Yi Kuan, Chienchou Yang, and Hung-yi Lee. 2024. [Large language model as an assignment evaluator: Insights, feedback, and challenges in a 1000+ student course](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2489–2513, Miami, Florida, USA. Association for Computational Linguistics.
- William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. [Learning to classify email into “speech acts”](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 309–316, Barcelona, Spain. Association for Computational Linguistics.
- Scott A. Crossley, Kristopher Kyle, and Danielle S. McNamara. 2017. The tool for the automatic analysis of text cohesion (taaco): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods*, 49(4):1227–1237.
- Scott A. Crossley and Danielle S. McNamara. 2022. [Computational assessment of text readability and comprehension](#). *Journal of Research in Reading*, 45(2):223–246.
- 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. *arXiv preprint arXiv:2407.09136*.
- Zeyu Dai and Ruihong Huang. 2018. [Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph](#). 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 141–151, New Orleans, Louisiana. Association for Computational Linguistics.
- Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.
- Dorottya Demszky, Devyani Sharma, Jonathan Clark, Vinodkumar Prabhakaran, and Jacob Eisenstein. 2021. [Learning to recognize dialect features](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2315–2338, Online. Association for Computational Linguistics.
- Zhujun Deng, Afida Mohamad Ali, and Zaid Bin Mohd Zin. 2025. [Investigating methodological trends of hedging strategies in academic discourse: A systematic literature review](#). *World Journal of English Language*, 15(5):322.
- Rudolf Flesch. 1948. [A new readability yardstick](#). *Journal of Applied Psychology*, 32(3):221–233.

- Bruce Fraser. 1999. What are discourse markers? *Journal of Pragmatics*, 31(7):931–952.
- Arthur C. Graesser, Sidney K. D’Mello, and Natalie K. Person. 2010. Metacognitive strategies in student learning: Do students practice retrieval when they study on their own? *Journal of Educational Psychology*, 102(4):805–820.
- John Hattie and Helen Timperley. 2007. The power of feedback. *Review of Educational Research*, 77(1):81–112.
- Cindy E. Hmelo-Silver, Ravit Golan Duncan, and Clark A. Chinn. 2006. Scaffolding and achievement in problem-based and inquiry learning: A response to kirschner, sweller, and clark (2006). *Educational Psychologist*, 42(2):99–107.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2023. Lora: Low-rank adaptation of large language models. *Transactions on Machine Learning Research*.
- Alon Jacovi, Yoav Goldberg, Aishwarya Kamath, Matthew Peters, and Roy Schwartz. 2024. Weak-link: Uncovering reasoning chain failures of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2341–2357. Association for Computational Linguistics.
- Raunak Jain. 2025. Emergent wisdom: Empowering constructivism by proxying human reasoning with llm thought traces. *OSF Preprints*. Preprint. https://osf.io/preprints/osf/dkst7_v1.
- Lan Jiang and Nigel Bosch. 2024. **Short answer scoring with GPT-4**. In *Proceedings of the 11th ACM Conference on Learning @ Scale (L@S ’24)*, pages 1–5, Atlanta, GA, USA. Association for Computing Machinery.
- Jaehun Jung, Faeze Brahman, and Yejin Choi. 2024. **Trust or escalate: LLM judges with provable guarantees for human agreement**. *arXiv preprint arXiv:2407.18370*.
- Jaehun Jung, Faeze Brahman, and Yejin Choi. 2025. **Trust or escalate: LLM judges with provable guarantees for human agreement**. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia. Oral paper.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.
- David R. Krathwohl. 2002. A revision of bloom’s taxonomy: An overview. *Theory Into Practice*, 41(4):212–218.
- Ehsan Latif and Xiaoming Zhai. 2024. **Fine-tuning ChatGPT for automatic scoring**. *Computers and Education: Artificial Intelligence*, 6:100210.
- Blair Lehman, Sidney D’Mello, Amber Strain, Caitlin Mills, Melissa Gross, Allyson Dobbins, Patricia Wallace, Keith Millis, and Arthur Graesser. 2019. Automated analysis of tutorial dialogues: Unsupervised modeling of student and tutor behaviors. In *International Conference on Artificial Intelligence in Education*, pages 117–127.
- Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. **HaluEval: A large-scale hallucination evaluation benchmark for large language models**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore.
- Justin Li, Jaemin Choi, Samir Yitzhak Gadre, Ansh Kaul, Louis-Philippe Morency, Rada Mihalcea, Minsu Seo, Darsh Shah, Luke Zettlemoyer, Amanda Stent, Jihie Hwang, Kyunghyun Cho, and Aniruddha Kembhavi. 2025. **Tutorgym: A benchmark for tutoring with large language models**. *arXiv preprint arXiv:2410.11895*.
- Ting Liu, Yiming Chen, Daniel Brown, Sebastian Riedel, and Hoifung Poon. 2023a. Pre-trained language models can be fully zero-shot learners. *Transactions of the Association for Computational Linguistics*, 11:1032–1049.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. **G-eval: Nlg evaluation using gpt-4 with better human alignment**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 7321–7335.
- 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.
- Jo Mackiewicz and Isabelle Thompson. 2010. Assertions of expertise in online tutoring sessions. *Journal of Business and Technical Communication*, 24(1):3–28.
- 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.

- Danielle S. McNamara, Arthur C. Graesser, Philip M. McCarthy, and Zhiqiang Cai. 2010. Linguistic features of writing quality. *Written Communication*, 27(1):57–86.
- Neil Mercer and Karen Littleton. 2009. Dialogue and the development of children’s thinking. *Educational Psychology in Practice*, 25(4):365–379.
- Sapna Negi and Paul Buitelaar. 2015. [Towards the extraction of customer-to-customer suggestions from reviews](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2159–2167, Lisbon, Portugal. Association for Computational Linguistics.
- Bill Noble and Vladislav Maraev. 2021. [Large-scale text pre-training helps with dialogue act recognition, but not without fine-tuning](#). In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 166–172, Groningen, The Netherlands (online). Association for Computational Linguistics.
- Martin Nystrand and Adam Gamoran. 1997. *Opening Dialogue: Understanding the Dynamics of Language and Learning in the English Classroom*. Teachers College Press, New York.
- Masanari Ohi, Masahiro Kaneko, Ryuto Koike, Mengsay Loem, and Naoaki Okazaki. 2024. [Likelihood-based mitigation of evaluation bias in large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3237–3245, Bangkok, Thailand.
- Theodoros Papadopoulos, Jiasheng Du, Claudiu Musat, Marija Bjelogrić, and Robert West. 2024. Language models for text classification: Is in-context learning enough? *arXiv preprint arXiv:2403.17661*.
- Muhammad Qorib, Geonsik Moon, and Hwee Tou Ng. 2024. [Are decoder-only language models better than encoder-only language models in understanding word meaning?](#) In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16339–16347, Bangkok, Thailand. Association for Computational Linguistics.
- Muhammad Abdul-Mageed Qureshi and Michael Strube. 2022. [Linguistically motivated features for classifying shorter text into fiction and non-fiction genres](#). In *Proceedings of the 29th International Conference on Computational Linguistics (COLING 2022)*, pages 924–934, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Tim Rowland. 2002. Being mathematically assertive: The role of hedges in mathematical discourse. *For the Learning of Mathematics*, 22(3):12–18.
- Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. [Verbosity bias in preference labeling by large language models](#). In *Proceedings of the NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.
- Gerard Salton and Christopher Buckley. 1988. [Term-weighting approaches in automatic text retrieval](#). *Information Processing & Management*, 24(5):513–523.
- Ted J. M. Sanders, Wilbert P. M. Spooren, and Leo G. M. Noordman. 2000. Toward a taxonomy of coherence relations. *Discourse Processes*, 15(1):1–35.
- Teven Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and 1 others. 2022. Bloom: A 176b-parameter open-access multilingual language model. In *arXiv preprint arXiv:2211.05100*.
- Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. 2022. [Confident adaptive language modeling](#). In *Advances in Neural Information Processing Systems 35*, pages 17456–17472.
- Philip H. Scott, Eduardo F. Mortimer, and Orlando G. Aguiar. 2002. The tension between authoritative and dialogic discourse: A fundamental characteristic of meaning making interactions in high school science lessons. *Science Education*, 90(4):605–631.
- Valerie J. Shute. 2008. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.
- 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.

- Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.
- Kurt VanLehn, Arthur C. Graesser, G. Tanner Jackson, Pamela Jordan, Andrew Olney, and Carolyn P. Rosé. 2006. When are tutorial dialogues more effective than reading? *Cognitive Science*, 30(1):3–62.
- Joshua Wagner, Tianyi Zhang, Stephen Bach, Ankit Jha, Michael Wornow, Avery Adler, Besmira Nushi, and Titus Glaunsinger. 2024. Label with confidence: Advancing selective prediction through ensemble agreement. In *Proceedings of the 41st International Conference on Machine Learning*, pages 22478–22501.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Interpretable automated feedback for student writing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4841–4852.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024a. [Large language models are not fair evaluators](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 9440–9450, Bangkok, Thailand.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024b. [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.
- Wayne Ward, Ron Cole, Daniel Bolaños, Cindy Buchenroth-Martin, Edward Svirsky, and Tim Weston. 2011. My science tutor: A conversational multimedia virtual tutor for elementary school science. *ACM Transactions on Speech and Language Processing*, 7(4):1–29.
- 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. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.
- Jörg Wittwer and Alexander Renkl. 2010. How effective are instructional explanations in example-based learning? a meta-analytic review. *Educational Psychology Review*, 22(4):393–409.
- Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachslar. 2021. Are we there yet? a systematic literature review on chatbots in education. *Frontiers in Artificial Intelligence*, 4:654924.
- Mengzhou Xia, Abdullah Ali, Angela Fan, Lilian Zhong, Allyson Ettinger, Ekin Akyurek, Margaret Mitchell, and Jacob Andreas. 2025. Llm-rubric: Evaluating machine-generated text with customized rubrics. *arXiv preprint arXiv:2403.06929*.
- Canwen Xu and Julian McAuley. 2022. A survey on dynamic neural networks for natural language processing. In *Journal of Machine Learning Systems*. Comprehensive review of early-exit and cascade methods in NLP.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- Zijiang Yang. 2024. [Improving the natural language inference robustness to hard dataset by data augmentation and preprocessing](#). *arXiv preprint arXiv:2412.07108*.
- Tianyu Zhang, Sayak Basu, Douwe Kiela, and Oriol Vinyals. 2024. Cascade-aware training and inference for more resource-efficient language models. *arXiv preprint arXiv:2405.12345*.

Averroes at BEA 2025 Shared Task: Verifying Mistake Identification in Tutor, Student Dialogue

Mazen Yasser¹, Mariam Saeed¹, Hossam Elkordi¹, Ayman Khalafallah¹

¹Applied Innovation Center,

Correspondence: m.yasser, m.saeed, h.elkordi, a.khalafallah@aic.gov.eg

Abstract

This paper presents the approach and findings of Averroes Team in the BEA 2025 Shared Task Track 1: Mistake Identification. Our system uses the multilingual understanding capabilities of general text embedding models. Our approach involves full-model fine-tuning, where both the pre-trained language model and the classification head are optimized to detect tutor recognition of student mistakes in educational dialogues. This end-to-end training enables the model to better capture subtle pedagogical cues, leading to improved contextual understanding. Evaluated on the official test set, our system achieved an exact macro- F_1 score of 0.7155 and an accuracy of 0.8675, securing third place among the participating teams. These results underline the effectiveness of task-specific optimization in enhancing model sensitivity to error recognition within interactive learning contexts.

1 Introduction

Tutoring has long been recognized as one of the most effective educational interventions, significantly enhancing student learning outcomes. Notably, the 2 sigma problem Bloom (1984) illustrates that students receiving one-on-one tutoring perform two standard deviations better than those in conventional classroom settings, highlighting the profound impact of personalized instruction. However, the scalability of such individualized tutoring remains a challenge due to resource constraints.

Advancements in deep learning Lin et al. (2023) and the emergence of large language models (LLMs) Lieb and Goel (2024); Park et al. (2024) have paved the way for AI-powered tutors capable of delivering personalized, on-demand educational support. These intelligent tutoring systems leverage natural language processing and machine learning techniques to adapt to individual learner needs, providing real-time feedback and tailored

instruction. AI-powered tutors can make quality education available to more people by offering the same benefits as one-on-one tutoring, but for many students at once.

Despite these advancements, evaluating the pedagogical effectiveness of AI tutors remains a significant problem. Traditional evaluation metrics, often adapted from domains like machine translation and summarization, fail to capture the nuanced educational interactions between AI tutors and students. Moreover, while human evaluations are considered the gold standard, they are time-consuming, costly, and lack scalability. This highlights the urgent need for automated, reliable, and pedagogically-informed evaluation frameworks.

Addressing this gap, the BEA 2025 Shared Task Kochmar et al. (2025) focuses on the Pedagogical Ability Assessment of AI-powered Tutors, aiming to develop standardized evaluation methods for AI tutor responses. The task includes four main tracks: Mistake Identification, determining whether the AI tutor correctly identifies student errors; Mistake Localization, pinpointing the exact location or nature of the student's mistake; Guidance Provision, offering constructive feedback or hints to guide the student; and Actionability, ensuring the response leads to a clear next step for the student. These tracks are intended to measure the tutor's effectiveness in supporting student learning and correcting misunderstandings.

This paper describes our contribution to the BEA 2025 Shared Task, in which we leverage large language models (LLMs) to create an automated evaluation method for AI tutors, primarily focusing on the mistake identification track. We investigate multiple strategies, assess their performance, and present a comprehensive ablation study, delivering a scalable, education-focused evaluation framework designed to enhance personalized learning.

2 Related Work

2.1 AI Tutoring Systems

Early Intelligent Tutoring Systems (ITS), developed in the late 1970s and 1980s [Guo et al. \(2021\)](#), employed explicit cognitive or knowledge-tracing models to monitor learners' progress and simulate personalized instruction. Pioneering systems like Anderson and Corbett's Cognitive Tutors [Anderson et al. \(1995\)](#) utilized model-tracing algorithms to instantly detect deviations from expert problem-solving pathways, allowing immediate corrective feedback and error-specific hints. This approach significantly boosted students' learning speed and post-test performance in experimental settings. However, studies of human expert tutors, such as [Hume et al. \(1996\)](#), suggest a more effective approach, using indirect prompts such as Socratic questions or reflective hints to help students independently identify and correct errors. This approach encourages deeper learning and self-reflection, showing a limitation of early ITS.

2.2 Advances in Large Language Models for Educational Dialogue

Recent advancements in large language models (LLMs) have significantly improved their capabilities, especially within educational contexts [Lieb and Goel \(2024\)](#); [Kasneji et al. \(2023\)](#); [Nye et al. \(2023\)](#). Modern LLMs facilitate personalized, interactive tutoring experiences, creating customized content such as quizzes and lesson plans tailored to specific curricula and student proficiency. Furthermore, these models support educators by automating administrative responsibilities, enabling teachers to devote more time to direct instruction and student engagement.

2.3 Evaluation Methods for AI Tutoring Systems

Evaluating AI tutors in education primarily relies on human judgment that score responses on dimensions like mistake identification, clarity, and tone. While expert annotation remains the gold standard, it suffers from inconsistency and lacks a unified protocol, prompting studies such as Tack & Piech [Tack and Piech \(2022\)](#), and Maurya et al. [Maurya et al. \(2025\)](#) propose standardized taxonomies. Pairwise comparisons simplify evaluation by focusing on relative pedagogical effectiveness. However, automatic metrics remain limited: Traditional natural language generation metrics such as BLEU [Pap-](#)

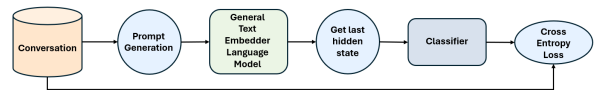


Figure 1: Model Architecture

[inani et al. \(2002\)](#) or ROUGE [Lin \(2004\)](#) poorly reflect pedagogical quality. Recent advances use reference-free approaches such as trained scorers (e.g., DialogRPT [Gao et al. \(2020\)](#)) and LLMs like GPT-4 ¹ to evaluate tutor responses, though their reliability depends heavily on prompt design. Hybrid evaluation methods that combine LLMs and correctness checks are emerging to improve consistency and scalability.

3 System Overview

This section presents the complete methodology adopted for the task. We first formalize the problem, then detail the shared backbone architecture, followed by dedicated subsections describing each experimental variant. Finally, we present our quantitative analysis and comparison between different approaches in 4.3.

3.1 Problem Definition

We address the task of assessing whether an AI tutor's feedback in a dialogue setting correctly identifies a student's mistake. Given a multi-turn conversation between a student and an AI tutor, along with the tutor's final response, the objective is to classify that response as correctly identifying the mistake, to some extent identifying, or failing to do so. This is formulated as a sequence classification problem, where a contextual understanding of the conversation is required for an accurate prediction.

3.2 System Backbone

We employ, as shown in Figure 1, a sequence classification approach. To effectively capture the contextual dependencies in the dialogue, we prepend a task-specific system prompt to the conversation history and the tutor's final turn. The system prompt is defined as:

¹<https://openai.com/index/gpt-4>

System Prompt

You are tasked with evaluating a multi-turn conversation between a math teacher and a student. The conversation is about a mathematical problem and in the form of a dialogue aimed at helping the student arrive at the correct solution.

The student initially provides an incorrect answer. The teacher then engages in follow-up exchanges to help the student uncover and understand the mistake.

You will be given:

- The full conversation up to the student's most recent turn, enclosed within '<CONV>' tags.

- The math teacher's immediate next response, enclosed within '<RESP>' tags.

****Your task****:

- Determine whether the teacher's response in '<RESP>' effectively contributes to identifying or addressing the student's mistake.

- Explain your reasoning clearly and concisely based on the content of the teacher's response and how it relates to the mistake and the original question. Then, provide your final judgment.

A teacher's response is considered a ****mistake identifier**** if it includes:

- A follow-up question, explanation, or prompt that targets the student's misunderstanding or errors in reasoning,

- Or if it guides the student toward re-evaluating key steps relevant to solving the original math problem.

You must output one of the following judgments based on the above criteria:

- ****A**** → If the teacher's response is clearly focused on the student's mistake and relates directly to the solution steps.

- ****B**** → If the response is unrelated to the mistake, irrelevant to the solution steps, or potentially confusing/misleading.

- ****C**** → If the response is only partially relevant or offers indirect guidance that might help the student reflect on the mistake.

****Put Your Output In The Following Format**** <think>The complete reasoning process</think><answer>Your final judgment from the choices (A, B, or C)</answer>

This input is passed through a decoder, where the last hidden-state representation is extracted. A lightweight classification head, implemented as a feed-forward linear layer, is then applied to predict how the tutor response identifies the mistake among three classes (Yes, No, To some extent). This design leverages the model's pretrained contextual embeddings, enhancing its capacity to discern nuanced dialogue interactions.

3.3 GTE-based Sequence Classification Models

We investigate three variants that use the *General Text Embedding* (GTE) family to obtain sentence-level representations, followed by lightweight feed-forward (FF) classification heads:

1. **GTE-MODERNBERT-BASE**² Zhang et al. (2024): the gte-modernbert-base encoder feeds into a single FF layer with a softmax output for prediction.
2. **GTE-QWEN2-1.5B-1FF** Li et al. (2023): embeddings from gte-qwen2-1.5B-instruct³ are passed through one FF layer identical to (1).
3. **GTE-QWEN2-1.5B-2FF**: the same as in (2) but followed by a two-layer FF head before the final softmax output.

Unless otherwise stated, these models are fine-tuned with the optimization settings described in §4.2.

3.4 Qwen2.5-based Sequence Classification Models

We benchmark five instruction-tuned *Qwen2.5* language models, varying both model size and the depth of the feed-forward (FF) classification head that replaces the original causal-LM head:

1. **QWEN2.5-7B-1FF** Team (2024): 7B parameters variant of Qwen2.5⁴; a single FF layer with softmax output. Fine-tuned via LoRA adapters Hu et al. (2022) with rank 16 on all attention and MLP projection layers.

²<https://huggingface.co/Alibaba-NLP/gte-modernbert-base>

³<https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct>

⁴<https://huggingface.co/Qwen/Qwen2.5-7B>

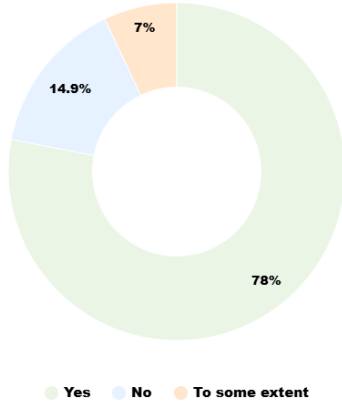


Figure 2: Class Distribution of the dataset.

2. **QWEN2.5-1.5B-2FF**: 1.5B parameters⁵; a two-layer FF head preceding the final softmax layer; full-parameter fine-tuning.
3. **QWEN2.5-MATH-1.5B-1FF**: math specialized 1.5B variant⁶; one FF layer; full-parameter fine-tuning.
4. **QWEN2.5-0.5B-1FF**: 0.5B parameters; one FF layer; full-parameter fine-tuning.
5. **QWEN2.5-0.5B-2FF**: same 0.5B backbone as (4) but with a two-layer FF head as in (2).

Unless otherwise stated, optimization hyperparameters follow the settings in §4.2.

4 Experiments

4.1 Dataset and Metrics

We conduct our experiments on **MRBench**, an annotated collection of 192 multi-turn student-AI tutor dialogues (1596 tutor responses) released by Maurya et al. (2025). Each tutor’s response is labeled to indicate whether the feedback correctly identifies the student’s error. Figure 2 shows the class distribution in the provided dataset. For model development, we divide the official development data into training and validation splits, retaining 15% of the dataset for validation during fine-tuning while maintaining the same class distribution of the train split. We follow the shared-task protocol and report strict *macro-averaged* F_1 and strict *accuracy* over the MISTAKE-IDENTIFICATION labels of the official test set.

⁵<https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct>

⁶<https://huggingface.co/Qwen/Qwen2.5-Math-1.5B-Instruct>

4.2 Training Setup

Each model was fine-tuned for no more than ten epochs using AdamW with a linearly decaying learning-rate schedule, reaching a maximum of 1×10^{-5} . We trained with an effective batch size of 64 in bf16 mixed precision on a single NVIDIA RTX-A6000 GPU.

4.3 Results and Analysis

Model	Accuracy (%)	Macro- F_1 (%)
GTE-MODERNBERT-BASE	88.17	66.48
GTE-QWEN2-1.5B-1FF	89.78	74.15
GTE-QWEN2-1.5B-2FF	89.25	72.51
QWEN2.5-7B-1FF	85.48	64.06
QWEN2.5-1.5B-2FF	88.44	71.69
QWEN2.5-MATH-1.5B-1FF	88.44	67.95
QWEN2.5-0.5B-1FF	<u>89.25</u>	<u>72.96</u>
QWEN2.5-0.5B-2FF	88.44	71.15

Table 1: Accuracy and Macro- F_1 on our validation split.

4.3.1 Full fine-tuning wins

Training the entire decoder-only model GTE-QWEN2-1.5B with a single feed-forward head (**1FF**) achieves the best results on our validation split at 74.15 macro- F_1 .

4.3.2 Small-but-efficient models keep pace

The smaller fully fine-tuned QWEN2.5-0.5B-1FF achieved our second best results at 72.96 macro- F_1 with only 1.2 points difference from our best model while cutting memory and latency.

4.3.3 More head depth is not always better

Adding a second feed-forward layer (**2FF**) to the backbone reduces performance.

4.3.4 Domain pre-training helps but not enough

The math-specialized QWEN2.5-MATH-1.5B-1FF outperforms the larger variant QWEN2.5-7B-1FF by 3.89 F_1 with only 20% of its parameter size. However, increasing parameter count of non-specialized models surpasses the benefit of domain-specific training. In our case, QWEN2.5-0.5B-1FF outperforms the trained model by 5.01, QWEN2.5-1.5B-2FF by 4.74, and GTE-QWEN2-1.5B-1FF by 6.2.

4.3.5 Size alone isn’t enough

The PEFT-tuned 7B QWEN2.5-7B-1FF achieves 6th place at 64.06 macro- F_1 , showing that the tuning was not effective.

5 Conclusion

This work benchmarked eight GTE- and Qwen2.5-based sequence-classification models on the MISTAKE-IDENTIFICATION task in AI-tutor dialogues. Full fine-tuning of a medium-sized decoder-only backbone (GTE-QWEN2-1.5B-1FF) achieved the strongest development performance at 74.1 macro-F₁, highlighting that carefully tuned 1.5 B models can outperform much larger 7B LoRA base-lines.

These findings indicate that compact instruction-tuned LLMs can rival, or even surpass, their larger counterparts in pedagogical mistake detection, offering a resource-efficient pathway toward scalable AI tutors. Future work should expand the dialogue corpus, diversify subject matter and languages, incorporate richer pedagogical labels, and pair automatic metrics with human and learning outcome evaluations to approach genuinely effective educational dialogue systems.

Limitations

Our study is constrained by several factors that temper the generality of its findings. First, the evaluation corpus, MRBench, comprises only 1596 labelled tutor responses drawn from a single English, mathematics-focused dataset. Such limited scale and topical focus may bias the models toward the annotation style and error distribution specific to this domain, leaving their behavior untested in other subjects, proficiency levels, or languages.

Second, the present metrics provide only a partial view of the educational effectiveness. Moreover, we rely exclusively on automatic accuracy and macro-F₁; the absence of human judgments or learning-gain measurements means that the impact in real-world scenarios remains uncertain.

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.
- 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.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking-training with large-scale human feedback data. In *EMNLP*.
- Lu Guo, Dong Wang, Fei Gu, Yazheng Li, Yezhu Wang, and Rongting Zhou. 2021. Evolution and trends in intelligent tutoring systems research: a multidisciplinary and scientometric view. *Asia Pacific Education Review*, 22(3):441–461.
- 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*.
- Gregory Hume, Joel Michael, Allen Rovick, and Martha Evens. 1996. Hinting as a tactic in one-on-one tutoring. *The Journal of the Learning Sciences*, 5(1):23–47.
- Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, and 4 others. 2023. [Chatgpt for good? on opportunities and challenges of large language models for education](#). *Learning and Individual Differences*, 103:102274.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.
- Anna Lieb and Toshali Goel. 2024. Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8.
- Chien-Chang Lin, Anna YQ Huang, and Owen HT Lu. 2023. Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. *Smart Learning Environments*, 10(1):41.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- 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.

- Benjamin D Nye, Dillon Mee, and Mark G Core. 2023. Generative large language models for dialog-based tutoring: An early consideration of opportunities and concerns. In *LLM@ AIED*, pages 78–88.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.
- Minju Park, Sojung Kim, Seunghyun Lee, Soonwoo Kwon, and Kyuseok Kim. 2024. Empowering personalized learning through a conversation-based tutoring system with student modeling. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–10.
- 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.
- Qwen Team. 2024. [Qwen2.5: A party of foundation models.](#)
- Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.

SmolLab_SEU at BEA 2025 Shared Task: A Transformer-Based Framework for Multi-Track Pedagogical Evaluation of AI-Powered Tutors

Md. Abdur Rahman¹ MD AL AMIN^{2*} Sabik Aftahee^{3*}

Muhammad Junayed³ Md Ashiqur Rahman¹

¹Southeast University, Dhaka, Bangladesh

²St. Francis College, Brooklyn, New York, USA

³Chittagong University of Engineering & Technology (CUET), Chittagong, Bangladesh

{202120000025@seu.edu.bd, alaminhossine@gmail.com,

u1904024@student.cuet.ac.bd, u2008023@student.cuet.ac.bd,

ashiqur.rahman@seu.edu.bd}

Abstract

The rapid adoption of AI in educational technology is changing learning settings, making the thorough evaluation of AI tutor pedagogical performance is quite important for promoting student success. This paper describes our solution for the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered tutors, which assesses tutor replies over several pedagogical dimensions. We developed transformer-based approaches for five diverse tracks: mistake identification, mistake location, providing guidance, actionability, and tutor identity prediction using the MRBench dataset of mathematical dialogues. We evaluated several pre-trained models including DeBERTa-V3, RoBERTa-Large, SciBERT, and EduBERT. Our approach addressed class imbalance problems by incorporating strategic fine-tuning with weighted loss functions. The findings show that, for all tracks, DeBERTa architectures have higher performances than the others, and our models have obtained in the competitive positions, including 9th of Tutor Identity (Exact F1 of 0.8621), 16th of Actionability (Exact F1 of 0.6284), 19th of Providing Guidance (Exact F1 of 0.4933), 20th of Mistake Identification (Exact F1 of 0.6617) and 22nd of Mistake Location (Exact F1 of 0.4935). The difference in performance over tracks highlights the difficulty of automatic pedagogical evaluation, especially for tasks whose solutions require a deep understanding of educational contexts. This work contributes to ongoing efforts to develop robust automated tools for assessing.

1 Introduction

In the past few years, the combination of natural language processing (NLP) and education technology has become one of the most popular areas of study to improve learning, automate feedback, and assist educators and students. With the expansion of blended and fully online courses, there has

been a marked increase in the need for scalable and sophisticated systems that can process learners' responses and tutors' comments. Such systems do not deal well with the subtle, context-sensitive characteristics of educational dialogues. So, assessing the pedagogical effectiveness and a standard evaluation taxonomy of such systems still remains a critical challenge.

An example of effective teaching is when an educator accurately pinpoints a student's misunderstanding, provides appropriate scaffolding towards clear concepts, and gives insightful feedback on desk-work that the students need to accomplish. Some automating aspects of this feedback loop, such as automated essay scoring (Phandi et al., 2015) and dialogic tutoring systems (Wang et al., 2024) have been given attention, but there is not much research that has been done to effectively capture the dynamics of the interplay student answers, tutor's engagement, and teaching style through feedback text's narrative structure.

While LLMs can generate coherent and contextually relevant responses, their ability to understand student misconceptions, provide actual guidance, and create meaningful learning experiences is not guaranteed. The general, area-independent metrics for natural language generation (NLG) (Liu et al., 2023; Gao et al., 2020) do not fit here as the majority of them lack consideration for pedagogical values and need gold references, which are seldom present in online interactions.

In this work, we tackle a comprehensive multi-track evaluation task designed for the evaluation of AI-tutor responses using a set of pedagogically motivated metrics. Building upon the foundations laid by the BEA 2023 Shared Task (Tack et al., 2023), which focused on generating AI teacher responses in educational dialogues, in the BEA 2025 Shared Task (Kochmar et al., 2025) iteration the focus shifted toward evaluating the quality of AI tutor responses. Specifically, it introduced a

*Authors contributed equally to this work.

taxonomy encompassing four pedagogically motivated dimensions: Mistake Identification, Mistake Location, Providing Guidance, and Actionability. Additionally, a fifth track challenged participants to identify the source of anonymized tutor responses, distinguishing between various LLMs and human tutors.

Our key contributions are as follows:

- Developing transformer-based approaches for comprehensive evaluation of AI-tutor responses using a set of pedagogically motivated metrics: mistake identification, mistake location, guidance provision, feedback actionability, and tutor identity prediction.
- Evaluated the performance of state-of-the-art transformer models across five key educational NLP tasks related to tutoring dialogues.

2 Related Works

Daheim et al. (2024) introduced a framework for stepwise solution verification for math reasoning, showing that grounding tutor responses in identified errors improves feedback accuracy where AI tutors are evaluated on their ability to identify and locate mistakes within student responses. Macina et al. (2023) presented *MathDial*, a large dataset of tutoring dialogues where LLMs often struggle with correct mistake spotting without targeted annotations. It includes annotations for mistake locations in math dialogues. This resource has been instrumental in training and evaluating models that can accurately identify and address specific errors in student solutions. Chen et al. (2024) proposed VATE, an AI-driven virtual teacher using prompt engineering and error pools for autonomous mistake analysis, achieving high accuracy in real-world deployment. Lastly, Macina et al. (2024) benchmarked pedagogical capabilities of LLM tutors, confirming that subject knowledge alone doesn't ensure effective error identification without specialized pedagogical training. Additionally, Yan et al. (2024) propose architectures designed to improve error localization in multimodal math tutoring, enhancing the clarity and usefulness of feedback. Recent work in intelligent tutoring systems (ITS) has emphasized the importance of scaffolding and adaptive feedback to enhance student learning outcomes. Liu et al. (2024) explored multimodal tutoring systems powered by large language models, demonstrating how pedagogical instruc-

tions can improve self-paced learning through structured scaffolding, evaluated via a seven-dimension rubric. Complementing this, Kochmar et al. (2020) showed that automated, personalized feedback using NLP and machine learning significantly boosts student performance, highlighting the need for tailoring feedback to individual learners. Similarly, Li et al. (2024) applied NLP-driven adaptive dialogs informed by the Knowledge Integration framework, illustrating how guided conversations help students integrate accurate scientific concepts during instruction. Together, these studies underline the potential of adaptive, pedagogically-aware NLP systems in delivering effective, personalized guidance within educational contexts. Maniktala et al. (2020) proposed "Assertions," an unsolicited hint mechanism delivering partially-worked example steps, which notably increased hint usage and improved learning outcomes, particularly for lower-proficiency learners. Blancas-Muñoz et al. (2018) further emphasized the importance of actionable support by comparing task-relevant hints to distractions in robotic tutoring, finding that direct, task-specific guidance led to better learner performance. Extending this focus to virtual education settings, Liang Liang (2025) applied NLP-based Seq2Seq models for automated feedback generation, achieving high accuracy while enhancing personalization and actionability of feedback in online environments. Collectively, these studies highlight that actionable, timely, and context-aware feedback mechanisms are essential for effective ITS design.

3 Task and Dataset Description

We competed on the BEA 2025 Shared Task¹ (Kochmar et al., 2025) on Pedagogical Ability Assessment of AI-powered tutors. The goal of the work is to assess AI tutor responses in mathematical dialogues when students make errors or show uncertainty. The provided dataset, MRBench (Maurya et al., 2025), includes dialogue contexts, the final student utterance, and corresponding tutor responses from various LLMs (e.g., GPT-4, Llama-3.1) and human tutors. The aim is to find the tutor or predict pedagogical quality in many spheres, including mistake identification and guidance.

The organizers provided Development set, `mrbench_v3_devset.json`, split into 90% for training (2,228 Instances) and 10% for validation (248 Instances). The final evaluation came from

¹<https://sig-edu.org/sharedtask/2025>

Split	Instances	Unique Words	Total Words
Train	2,228	8,134	512,392
Validation	248	3,833	52,981
Test	1,547	7,057	454,720

Table 1: Dataset statistics across different splits.

the Test set, `mrbench_v3_testset.json` (1,547 Instances). Table 1 shows the dataset statistics.

4 Methodology

This section outlines our approaches utilized for Track 1 - Mistake Identification, Track 2 - Mistake Location, Track 3 - Providing Guidance, Track 4 - Actionability, and Track 5 - Guess the Tutor Identity. The study evaluated many transformer-based approaches using hyperparameter optimization to improve performance. The architectural frameworks used for all tasks is illustrated in Figure 1

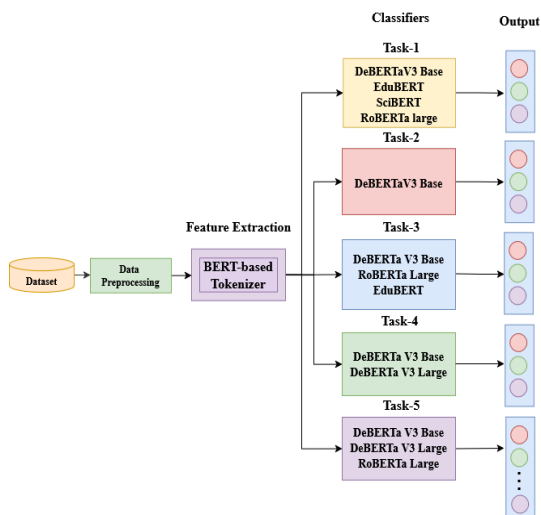


Figure 1: Overview of the Pedagogical Ability Assessment Process for AI-powered Tutors

4.1 Data Preprocessing and Feature Extraction

We processed `mrbench_v3_devset.json` and `mrbench_v3_testset.json` files for all five tracks. Every distinct tutor response in a conversation stood isolated. Using descriptive markers and newlines, concatenating the “Conversation History” and “Tutor Response” produced the input text for models; instances lacking tutor responses were removed. Relevant annotations (e.g., “Mistake_Identification”) were retrieved for Tracks 1–4 (Mistake Identification, Mistake Location, Providing Guidance, Actionability); their

“Yes,” “To some extent,” “No” labels were mapped to [0, 1, 2]. Development set tutor identities for Track 5 (Guess the Tutor Identity) were mapped to one of nine canonical tutor classes then to numerical labels [0–8]. Feature extraction used pre-trained Transformer models (DeBERTa-V3 base/large, RoBERTa-Large, EduBERT, SciBERT). each model’s particular AutoTokenizer turned input texts into `input_ids`, `attention_mask`, and optionally `token_type_ids`, Padded or trimmed to 512 tokens.

4.2 Transformer-Based Models

The methodological foundation for all five tracks of BEA 2025 Shared Task focuses on the fine-tuning of pre-trained Transformer models (Vaswani et al., 2017). These architectures, with their well-known self-attention mechanisms, are proficient in capturing contextual relationships within text because of the highly sophisticated contexts and excel at capturing intricate contextual relationships within text makes them very suitable for a range of challenges in Natural Language Processing (NLP) (Devlin et al., 2019). Transformer’s ability to model long-range dependencies is critical given the nuanced nature of assessing pedagogical abilities and identifying distinctive tutor characteristics from snippets of dialogues. A collection of models from the Hugging Face Transformers library² was chosen, including those pre-trained specifically on scientific or educational corpora as well as more general NLU models. SciBERT (Beltagy et al., 2019) and EduBERT (Clavié and Gal, 2019) are two, alongside RoBERTa-Large (Liu et al., 2019) and DeBERTa-V3 base and large configurations (He et al., 2021). For each task, these pretrained encoders were modified by adding a sequence classification head for each task. This head has a dropout layer and a linear layer that maps the output representation of the encoder associated with the special [CLS] token to the logits for the respective number of classes for each track. All models had the same input constructed by joining the “Tutor Response” and “Conversation History”. Model-specific tokenizers were used according to each model’s pretraining, with padding and truncation to 512 tokens.

In Track 1, Mistake Identification, the goal was a 3-way classification problem determining if a tutor’s response acknowledged a student’s mistake,

²<https://huggingface.co/transformers>

with labels “Yes”, “To some extent”, and “No”. For this track, we experimented with SciBERT, EduBERT, RoBERTa-Large and DeBERTa-V3 base. SciBERT, which was pretrained on a large corpus of scientific literature, was selected because it could be expected to perform well with the formal and technical language of mathematics. EduBERT was chosen because it was trained on educational data, which may enhance understanding in teaching cases. RoBERTa-Large, a robustly optimized model, served as a strong general-purpose baseline, while DeBERTa-V3 base offered a more recent architecture known for its efficiency and strong performance. As highlighted, class imbalance was tackled by fine-tuning SciBERT models with weighted CrossEntropy Loss, where the greater class imbalance was compensated by inversely modifying class weights to their occurrence within the training data, and also Focal Loss (Lin et al., 2017) (with $\gamma = 2.0$ and $\alpha = 2.0$ in some configurations of SciBERT) that diminishes the emphasis on well-classified examples. For estimation smoothing SciBERT’s label smoothing was set to 0.1 which was designed to counterbalance overconfidence. EduBERT and RoBERTa-Large models under this track predominantly applied weighted CrossEntropy Loss while the DeBERTa-V3 base model for this track used the standard Cross Entropy Loss provided by the Hugging Face sequence classification framework. These models were trained with the goal of detecting nuanced indications of mistake recognition in tutor responses.

Track 2, Mistake Location, was developed with similar 3-way classification where response “Yes”, “To some extent” and “No” were used to capture if a tutor is precise to the location of the student’s error. For this track, we primarily utilized the DeBERTa-V3 base model. With disentangled attention and the new pre-training objective (ELECTRA-style) DeBERTa-V3 architecture enhances understanding for relations between tokens and the context which we believed could prove useful in determining whether certain parts of the student’s solution were referred to. Cross Entropy Loss with weights was implemented for fine-tuning for this track. This was important considering that the label distribution for “Mistake Location” was often skewed, and weighting is known to address underrepresented classes effectively trying to achieve understanding if the understanding was indeed accurate and crystal clear.

For Track 3, Providing Guidance, the focus was

on assessing the tutor’s evaluation on whether the answer provided to the student was useful, relevant, correct, and helpful, once again using 3-class schema (“Yes,” “To some extent,” “No”). In this track, we experimented with DeBERTa-V3 base, RoBERTa-Large, and EduBERT. The selection of DeBERTa-V3 and RoBERTa-Large was driven by their proficiency in NLU which is vital when evaluating the guidance provided on whether it is correct and relevant. EduBERT was included because his domain-specific pre-training could help identify pedagogically sound explanations, hints, or supporting questions. As in the last tracks, all these models were first fine-tuned using weighted Cross Entropy Loss. This was important for illustrating how the models adapted to differentiate effective and partially effective guidance along with ineffectual guidance, all distinct components of instructional prowess.

Track 4, Actionability, checked if the tutor’s commentary offered unambiguous next steps by employing the same 3-way classification labels. For this track, we trained both DeBERTa-V3 ‘base’ and ‘large’ models. The justification for the ‘large’ variant is to test if additional model size could capture the more acute interpretative reasoning necessary to assess if a tutor’s remark was adequately sharp and instructive to enable responsive movement from the student. The larger model, with more parameters, better at understanding implicit suggestions or clues regarding the clarity of the anticipated student answer. During training, we uniformly used a weighted Cross Entropy Loss for all layers to constrain the label distribution along this dimension, hoping that the models could reliably distinguish non-constructive or minimal responses for a given prompt from non-informative utterances and conversational dead ends.

Finally, Track 5, Guess the Tutor Identity, posed a challenge of 9 classes: who among the tutors (Expert, Novice, or one of seven LLMs) gave the response in the anonymized form. For this exercise, we used DeBERTa-V3 base, DeBERTa-V3 large, and RoBERTa-Large. These techniques were chosen because of their past performance in capturing sophisticated stylistic differences, preferences, and idiosyncratic features like distinct ‘fingerprints’ for human tutors and LLM systems. The problem is complex at its core because varying forms or expressions, such as style fusion, which exist in different domains like various LLMs or between a novice human and some LLMs, might deeply over-

lap. Addressing the multi-class setup with nine distinct tutor identities was particularly challenging due to the imbalance in the number of available examples for each tutor. To mitigate this situation, we relied heavily on weighted Cross Entropy Loss which disproportionate class representation mitigates imbalance with a more prevailing class in the data. In turn, this prevented the models from specializing excessively to the most common types of tutors.

Throughout all five tracks, there were several components that the fine-tuning procedure shared homogeneous components. We utilized the AdamW optimizer (Loshchilov and Hutter, 2017), which integrates weight decay more effectively than traditional Adam, helping to prevent overfitting. As well as a blended learning rate scheduler which warms up for the first 10% of the total training steps. Doing so reinforces training stability during the early epochs. For example, many SciBERT and RoBERTa-L configurations achieve effective batch sizes of 16 with a device batch size of 8 and 2 accumulation steps. This technique of accumulation helps in training large models on memory-restricted GPUs while also allowing for smoother gradient estimates and enhanced model performance. As shown in Table 2, training continued until reaching the set maximum number of epochs. The model for each task was finalized based on the validation set with the highest macro F1 score for Tracks 1-4 and accuracy on Track 5. This selection process acts as an implicit early stopping mechanism. The class weights for the Cross Entropy Loss were determined by the label’s corresponding training portion frequency across the development set, meaning classes who were less present in the dataset had a greater impact on loss and as such received more focus from the model. All experiments were carried out with a fixed random seed (SEED = 42) in order to ensure the reproducibility of our results.

5 Result Analysis

This section presents an analysis of the performance of various Transformer-based models across the five tracks of the BEA 2025 Shared Task. The evaluation metrics, as defined by the shared task organizers, include exact and lenient accuracy and macro F1-score for Tracks 1-4, and exact macro F1-score for Track 5. The performance of our submitted models is detailed in Table 3.

Model	LR	WD	BS	GA	EP
Track 1: Mistake Identification					
SciBERT	1e-5	0.01	8	2	12
EduBERT	1.5e-5	0.01	8	2	12
RoBERTa-Large	1.5e-5	0.01	8	2	12
DeBERTa-V3-Base	2e-5	0.01	8	1	8
Track 2: Mistake Location					
DeBERTa-V3-Base	1.5e-5	0.01	8	2	12
Track 3: Providing Guidance					
DeBERTa-V3-Base	1.5e-5	0.01	8	2	12
RoBERTa-Large	1.5e-5	0.01	8	2	12
EduBERT	1.5e-5	0.01	8	2	12
Track 4: Actionability					
DeBERTa-V3-Base	1.5e-5	0.01	2	2	12
DeBERTa-V3-Large	1.5e-5	0.01	2	2	12
Track 5: Tutor Identity					
DeBERTa-V3-Base	2e-5	0.01	8	2	15
DeBERTa-V3-Large	1.8e-5	0.01	2	2	10
RoBERTa-Large	2e-5	0.01	8	2	15

Table 2: Hyperparameters used across the five tracks. LR: Learning Rate, WD: Weight Decay, BS: Batch Size, GA: Gradient Accumulation, EP: Epochs.

Model	E-F1	E-Acc	L-F1	L-Acc
Track 1: Mistake Identification				
RoBERTa Large	0.6339	0.7938	0.8395	0.9043
SciBERT	0.6393	0.8500	0.8545	0.9121
EduBERT	0.6597	0.8429	0.8665	0.9205
DeBERTaV3 Base	0.6617	0.8397	0.8782	0.9315
Track 2: Mistake Location				
DeBERTaV3 Base	0.4935	0.6057	0.7051	0.7401
Track 3: Providing Guidance				
RoBERTa Large	0.4758	0.5863	0.6997	0.7750
EduBERT	0.4918	0.5785	0.6885	0.7395
DeBERTaV3 Base	0.4933	0.5695	0.6990	0.7608
Track 4: Actionability				
DeBERTaV3 Base	0.6117	0.6781	0.8170	0.8500
DeBERTaV3 Large	0.6284	0.6955	0.8223	0.8565
Track 5: Tutor Identity				
RoBERTa Large	0.8237	0.8151	-	-
DeBERTaV3 Base	0.8618	0.8597	-	-
DeBERTaV3 Large	0.8621	0.8621	-	-

Table 3: Performance of all models across five tracks. E-F1: Exact macro F1 score, E-Acc: Exact Accuracy, L-F1: Lenient macro F1 score, L-Acc: Lenient Accuracy

For Track 1, Mistake Identification, DeBERTa-V3 Base has achieved the best exact macro F1 score of 0.6617 achieving the highest exact macro F1 score. This model also demonstrated strong performance with a lenient macro F1 score of 0.8782 and lenient accuracy of 0.9315. EduBERT’s performance on exact macro F1 score was just slightly weaker at 0.6597 while SciBERT had the best exact accuracy of 0.8500. The best exact macro F1 score with DeBERTa-V3 Base seems to suggest that, even with a standard Cross Entropy Loss, there are greater architectural advantages in the model that allow it to grasp the intricacies of 3-way classification better than other models. The results from

SciBERT and EduBERT indicate that the use of weighted loss functions was beneficial in achieving competitive exact scores and were likely instrumental in achieving class imbalance resolution.

For Track 2, Mistake Location, our sole entry was the DeBERTa-V3 Base model which was tuned with weighted Cross Entropy Loss and achieved an exact macro F1 score of 0.4935, lenient macro F1 score of 0.7051. There were no other entries. Scores suggest that identifying the precise origins of a mistake’s location is strictly harder than simply identifying an error. The considerable gap between the exact macro F1 score and lenient macro F1 scores highlights that while the model could often recognize some level of mistake location awareness ("To some extent"), achieving definitive localization ("Yes") was less frequent.

For Track 3, Providing Guidance, DeBERTa-V3 Base reached the highest exact macro F1 score of 0.4933. EduBERT was a strong contender with exact macro F1 score of 0.4918, while RoBERTa-Large scored 0.4758 in exact macro F1 score. The lenient macro F1 scores were approximately 0.69 for all three models, with RoBERTa-Large and DeBERTaV3-Base at 0.6997 and 0.6990 respectively. The exact macro F1 scores, marginally surpassing 0.50, highlight the challenge posed in automatically evaluating the correctness and relevance of pedagogical guidance. The imbalances among the “Yes”, “To some extent”, and “No” categories for this particular dimension is what prompted the use of weighted Cross Entropy Loss for this model causing all of the categories to blend in with the aim of unifying the discrepancies.

In Track 4, Actionability, DeBERTa-V3 Large showed the best performance with an exact macro F1 score of 0.6284 and exact Accuracy of 0.6955. The DeBERTa-V3 Base model was slightly behind with an exact macro F1 score of 0.6117. It seems that the larger sized DeBERTa model boosts with added features helped with classifying the actionability of tutor responses. Both models had employed Cross Entropy Loss that was helpful for the other model in achieving such classifiers.

For Track 5, Guess the Tutor Identity, where exact macro F1-score is the primary metric, DeBERTa-V3 Large achieved the best exact macro F1 score of 0.8621. The corresponding exact accuracy for this model was also 0.8621. DeBERTa-V3 Base also performed robustly with an exact macro F1 score of 0.8618 and exact accuracy of 0.8597, followed by RoBERTa-Large at 0.8237 (ex-

act macro F1 score) and 0.8151 (exact accuracy). The strong performance, particularly of the DeBERTa architectures, indicates their capability to discern subtle stylistic and content based patterns distinguishing the nine different tutor identities. This multi-class problem for which the weighted Cross Entropy Loss was quite useful for dealing with was clearly non-trivial.

To summarize, DeBERTa-V3 base and large architectures achieved the best results considering the most important evaluation metric is exact macro F1 score for most tracks. The large showed some advantages in Tracks 4 and 5 where increased model complexity might be helpful. The low exact macro F1 scores, especially for Tracks 2 and 3, suggest difficulties automatically capturing the intricacies of assessment within teaching highlight the intricacies involved in evaluating pedagogical features.

6 Conclusion

This paper introduces a system developed for the UNLP This paper detailed our participation in the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered Tutors, presenting systems built upon fine-tuned Transformer models. We evaluated multiple architectures including DeBERTa-V3, RoBERTa-Large, SciBERT, and EduBERT for the five distinct tracks of Mistake Identification, Mistake Location, Providing Guidance, Actionability, and Tutor Identity. Throughout the investigations, DeBERTa-V3 was the top performer across all tracks based on the primary Exact macro F1 score metric. For Mistake Identification, DeBERTa-V3 Base achieved the Exact macro F1 score of 0.6617, and for Providing Guidance, 0.4933. For the other tracks of Actionability and Tutor Identity, DeBERTa-V3 Large excelled with 0.6284 and 0.8621 Exact macro F1 score respectively. For Mistake Location, DeBERTa-V3 Base scored an Exact macro F1 score of 0.4935. These findings support the assertion that sophisticated Transformer models are capable of intricate pedagogical assessments. The Exact macro F1 scores obtained for Providing Guidance and Mistake Location depict the challenges associated with higher-degree classification, demonstrating the inherent difficulty of the tasks. Methodological choices, such as strategic hyperparameter tuning and the application of appropriate loss functions (e.g., weighted Cross Entropy Loss or Focal Loss) to manage class imbalances, were important for optimizing performance. This work

contributes to the ongoing efforts to develop robust automated tools for assessing and improving AI tutor effectiveness in educational dialogues.

Limitations

Our study, while demonstrating the effectiveness of Transformer models for assessing pedagogical abilities, has several limitations. First of all, the performance, especially on exact macro F1-scores for challenging tasks like Mistake Location and Providing Guidance, indicates that current models still find it difficult to have the fine-grained semantic knowledge needed for these demanding tests. Second, our method depends on the particular annotations and definitions given in the MR-bench dataset; model performance may change depending on alternative educational taxonomies or data from other fields outside mathematics. Moreover, although weighted loss functions helped us to solve class imbalance, significant imbalances for some labels or tutor identities could still influence generalization. Finally, the computational resources needed for fine-tuning and experimenting with several big Transformer models can be significant, therefore perhaps restricting more general architectural research or more comprehensive hyperparameter searches.

Acknowledgments

This work was supported by Southeast University, Bangladesh.

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.

Maria Blancas-Muñoz and 1 others. 2018. [Hints vs distractions in intelligent tutoring systems: Looking for the proper type of help](#). *arXiv preprint arXiv:1806.07806*.

Hao Chen and 1 others. 2024. [Ai-driven virtual teacher for enhanced educational efficiency: Leveraging large pretrain models for autonomous error analysis and correction](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press.

Benjamin Clavié and Kobi Gal. 2019. [Edubert: Pre-trained deep language models for learning analytics](#). *arXiv preprint arXiv:1912.00690*.

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 (EMNLP)*.

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.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. [Dialogue response ranking training with large-scale human feedback data](#). *arXiv preprint arXiv:2009.06978*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). *arXiv preprint arXiv:2111.09543*.

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](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.

Ekaterina Kochmar and 1 others. 2020. [Automated personalized feedback improves learning gains in an intelligent tutoring system](#). *arXiv preprint arXiv:2005.02431*.

Chen Li and 1 others. 2024. [Applying natural language processing adaptive dialogs to promote knowledge integration during instruction](#). *Education Sciences*, 15(2):207.

Meng Liang. 2025. [Leveraging natural language processing for automated assessment and feedback production in virtual education settings](#). *Journal of Educational Computing Research*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. [Focal loss for dense object detection](#). In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Yang Liu, Dan Iter, Yichong Xu, Shuhang Wang, Ruo Chen Xu, and Chenguang Zhu. 2023. [G-eval: Nlg evaluation using gpt-4 with better human alignment](#). *arXiv preprint arXiv:2303.16634*.

- 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*.
- Zhengyuan Liu and 1 others. 2024. [Scaffolding language learning via multi-modal tutoring systems with pedagogical instructions](#). *arXiv preprint arXiv:2404.03429*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- 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*.
- Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. [Math-tutorbench: A benchmark for measuring open-ended pedagogical capabilities of llm tutors](#). *arXiv preprint arXiv:2405.12240*.
- Mehak Maniktala and 1 others. 2020. [Avoiding help avoidance: Using interface design changes to promote unsolicited hint usage in an intelligent tutor](#). *arXiv preprint arXiv:2009.13371*.
- 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.
- Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. 2015. [Flexible domain adaptation for automated essay scoring using correlated linear regression](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Lisbon, Portugal. Association for Computational Linguistics.
- 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.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Deliang Wang, Dapeng Shan, Ran Ju, Ben Kao, Chenwei Zhang, and Gaowei Chen. 2024. Investigating dialogic interaction in k12 online one-on-one mathematics tutoring using ai and sequence mining techniques. *Education and Information Technologies*, pages 1–26.
- Yibo Yan, Shen Wang, Jiahao Huo, Philip S. Yu, Xuming Hu, and Qingsong Wen. 2024. [Mathagent: Leveraging a mixture-of-math-agent framework for real-world multimodal mathematical error detection](#). *arXiv preprint arXiv:2405.12284*.

RETUYT-INCO at BEA 2025 Shared Task: How Far Can Lightweight Models Go in AI-powered Tutor Evaluation?

Santiago Góngora † and Ignacio Sastre † and Santiago Robaina
Ignacio Remersaro and Luis Chiruzzo and Aiala Rosá

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 2025 shared task. Our participation was characterized by the decision of using relatively small models, with fewer than 1B parameters. This self-imposed restriction tries to represent the conditions in which many research labs or institutions are in the Global South, where computational power is not easily accessible due to its prohibitive cost. Even under this restrictive self-imposed setting, our models managed to stay competitive with the rest of teams that participated in the shared task. According to the *exact* F_1 scores published by the organizers, the performance gaps between our models and the winners were as follows: 6.46 in *Track 1*; 10.24 in *Track 2*; 7.85 in *Track 3*; 9.56 in *Track 4*; and 13.13 in *Track 5*. Considering that the minimum difference with a winner team is 6.46 points — and the maximum difference is 13.13 — according to the *exact* F_1 score, we find that models with a size smaller than 1B parameters are competitive for these tasks, all of which can be run on computers with a low-budget GPU or even without a GPU.

1 Introduction

The remarkable advances in the development of Large Language Models (LLMs) in recent years have turned Natural Language Processing into a discipline with great potential for application in different domains, and *Education* is not the exception (Ignat et al., 2024). However, these technological advances are not affordable to everyone. The cost of closed models — which are the most powerful and are typically considered the State-of-the-art in NLP — and the expensive infrastructure required to use large open models, coupled with

negative effects on the environment, make research on other methods still essential.

Our RETUYT-INCO team, as a research lab from South America, is no exception to this reality. Naturally, we are concerned about these issues and, consequently, we have focused on experimenting with open models in recent editions of the BEA shared tasks. For the 2023 shared task, consisting in generating teacher responses in educational dialogues (Tack et al., 2023), we participated using open models, obtaining competitive results (Baladón et al., 2023). One of the highlights of our participation was the “*Hello*” baseline, a simple strategy we followed which achieved remarkable results, unveiling the fragility of BERTScore (Zhang et al., 2020). More recently, for the 2024 BEA shared task, consisting in performing simplification experiments for different languages (Shardlow et al., 2024), we mainly focused on fine-tuning BERT and Mistral models (i.e., open models), even using synthetic data in some cases (Sastre et al., 2024).

In this paper, we present the RETUYT-INCO participation in the **five tracks** of the BEA 2025 Shared Task: *Pedagogical Ability Assessment of AI-powered Tutors* (Kochmar et al., 2025). This year, in addition to maintaining our restriction of working with open models, we challenged ourselves with an extra restriction: to experiment only with language models of fewer than a billion parameters and classical machine learning (ML) approaches. We will call these *lightweight* models, as they have to be small enough to run on a low-end GPU or with no GPU at all. This restriction is related to the situation many research labs face every day in the Global South: the lack of minimum resources to run what other regions consider *small* models (7B parameters or more). In our case, we have limited access to a national computing cluster, which we can use to fine-tune LLMs up to 7B parameters, but we do not have resources to host the fine-tuned

† These (corresponding) authors contributed equally to this work: {sgongora, isastre}@fing.edu.uy

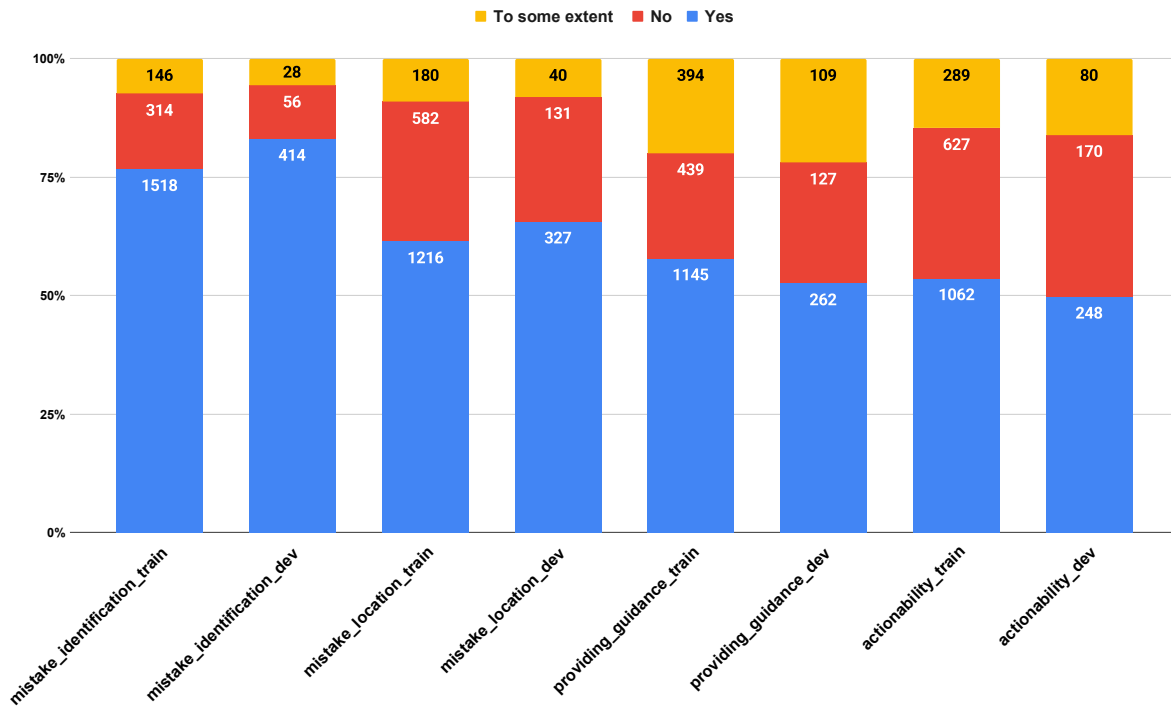


Figure 1: Class balance in our train and dev sets. The columns are coupled according to classes.

LLMs and use them in real applications.

Moreover, this is not the only motivation for this self-imposed restriction, as one of the research lines of our lab is the application of NLP tools to aid teachers in rural areas (Chiruzzo et al., 2022; Rosá et al., 2025). In such contexts it is very unlikely to use state-of-the-art LLMs, due to the impossibility of using them through APIs (since children’s privacy is key, sending private data to third-party servers is not an option), and the prohibitive cost of installing capable GPUs in trustworthy servers.

Overall, throughout this paper, we will try to answer the general research question that motivated our participation: *What is the performance gap between lightweight models and those state-of-the-art models, which would naturally have a better chance of winning the competition?*.

2 Dataset

For this edition, the dataset consists of 300 conversations (Maurya et al., 2025). Each dialogue is composed of interactions between a teacher and a math student. In the final turn of each dialogue the student shows clear confusion about a concept, and the dataset includes potential tutor *responses* intended to help the student. These responses — some of them generated by seven LLM-based tutors and others written by human tutors — are also evaluated by human evaluators (using **Yes**, **No** or

To some extent) according to four dimensions of interest that coincide with the four proposed tracks in the shared-task: *mistake identification*, *mistake location*, *providing guidance* and *actionability*.

Due to the lack of a specific *development* set, during the first month we split the official dataset published by the organizers into two parts: 80% We decided to do this split focusing on the conversations — and not on the responses — trying to ensure that each conversation and all its responses stayed either in the *train* or the *dev* set. As a consequence, our *train-dev* split may not preserve the class balance of the original set. Figure 1 shows the class balance for each dimension in our *train* and *dev* partition.

3 Considered approaches

For our experiments we considered classical ML classification algorithms, BERT-based approaches and fine-tuning a small autoregressive language model. Since all the tracks in the shared task are classification problems, many of the models we considered were used in more than one track. All of them were trained (or fine-tuned) using our *train* set, running on GoogleColab¹ or a national computational cluster (see Section 3.4.1). At the end of this section — in Subsection 3.5 — we will show

¹<https://colab.research.google.com/>

the results we obtained when evaluating them on the dev set.

3.1 Preliminary experiments

To gain a greater understanding of how challenging the *tracks* were, we performed four preliminary experiments with increasing degree of complexity.

The first and most basic one consisted in answering always **yes**, what naturally degraded the F_1 macro score, since the **No** and **To some extent** classes were never chosen. Then, we tried using a random classifier, which consistently yielded accuracy values around 33%

Additionally, before imposing ourselves the constraint of using *lightweight* models only, we wanted to have an informed perception of how well bigger LLMs could perform in these tracks. Therefore, we explored both prompting a closed model via an API and fine-tuning an open model. For prompting, the model we chose was Gemini Flash 2.0 Lite², using the prompt reported in the paper that presented the dataset (Maurya et al., 2025). For fine-tuning, we chose Llama 3.1 8B Instruct (AI@Meta, 2024), and used Low Rank Adaptation (LoRA) (Hu et al., 2022). This experiment follows the same methodology explained in Section 3.4.

3.2 Classical Machine Learning approaches

Among all the available algorithms in Sklearn³ we tried, those that had the best performance on the dev set were *Random Forest*, *SVC* (Support-vector classifier) and *k-NN* (k-Nearest Neighbors). To represent the input texts we experimented with Bag of Words and TF-IDF, trying different n-gram ranges from $n = 1$ to $n = 8$.

Since all these algorithms have problems capturing the complexities of long-context dependencies, after some experimentation we decided to train the models using the *response* text only, i.e. without taking into consideration the full interaction between the student and the tutor. Those preliminary experiments we did with the full interaction (i.e. concatenating the response of the tutor to the conversation history) had a notably lower performance.

3.3 BERT-based approaches

We also tried BERT-based approaches. We experimented with fine-tuning them, combining them

²<https://deepmind.google/technologies/gemini/flash-lite/>

³https://scikit-learn.org/stable/supervised_learning.html

with classification algorithms and also with some rules.

3.3.1 BERT for Tracks 1–4

We implemented a simple method by fine-tuning a simple BERT model for tracks 1 through 4. Specifically, we finetuned the DistilBERT `distilbert-base-uncased` variant (Sanh et al., 2020), a compact and computationally efficient distillation of BERT with approximately 66 million parameters.

For this experiment, we only considered the response text as input data, without the conversation history. We fine-tuned the model to predict each of the target variables (`mistake_identification`, `mistake_location`, `providing_guidance`, `actionability`). We initially tried to fine-tune the model in a three-class configuration, but our experiments were unable to predict any value of the class **To some extent** whatsoever, so we changed the approach. We ended up training two-class models, joining **No** and **To some extent** as the negative class. After fine-tuning, we analyzed the logit of the positive class and observed that even if both classes were lumped together during training, the **No** values actually got lower logit than the **To some extent** values, which allowed us to define thresholds to separate the three classes.

The hyperparameters in these experiments were the number of training epochs (from 1 to 3) and two thresholds to distinguish the frontier between **No** and **To some extent**, and between **To some extent** and **Yes**, which depending on the target output could vary between -1 and +1. In this round of experiments, we used Adam optimization with a learning rate of 5×10^{-6} .

3.3.2 BERT for Track 5

In our approach to track 5, the objective was to classify the tutor identity based once again solely on the provided response text. For fine-tuning, the following parameters were used: a learning rate of 2×10^{-5} , a weight decay of 0.01, a training duration of 4 epochs, and batch sizes set to 16.

3.3.3 Sentence Embeddings

In addition to fine-tuning, we explored the use of BERT-like models to generate sentence embeddings (Reimers and Gurevych, 2019), which were then combined with classical ML methods for classification.

For tracks 1–4, we used the

multilingual-e5-large-instruct⁴ model (Wang et al., 2024), a multilingual encoder initialized from xlm-roberta-large (Conneau et al., 2019) (561M parameters). We generated a sentence embedding for each example in our training partition and then used those embeddings as input to classical classifiers: k-NN and multilayer perceptron (MLP). For this approach we explored three different input configurations:

- **Response-only:** The input consists solely of the embedding corresponding to the response to be evaluated.
- **Response + conversation history:** The input is formed by concatenating the embedding of the response with the embedding of the full conversation history.
- **Response + conversation history + LLM probabilities:** The input extends the previous configuration by appending the probabilities assigned to the three class labels by the fine-tuned LLM (see Section 3.4).

For the k-NN classifier, we chose $k = 9$ based on the performance on our dev set prior to submitting predictions for the competition’s test set. For the MLP, we used a simple model with no hidden layer and trained it until convergence, defined as no improvement greater than a tolerance of 1×10^{-4} for 10 consecutive iterations.

For the mistake identification dimension, due to the high class imbalance in the training data, we applied under-sampling by fitting the k-NN classifier on a perfectly balanced subset. This subset contained an equal number of examples for each class, matching the count of the least frequent class. As shown in Section 4, this strategy led to improved performance. For this classifier, different values of k for each track were chosen (mistake identification: 415; mistake location: 540; providing guidance: 125; actionability: 96).

For track 5, we explored leveraging the DistilBERT model that was previously fine-tuned for direct sequence classification (as described in the previous section). In this setup, the core transformer layers (the base model, without the classification head) of this fine-tuned DistilBERT were employed as a feature extractor. Embeddings were generated for the "response" texts. These DistilBERT-derived

embeddings were used as input features for an XGBoost classifier (Chen and Guestrin, 2016), which was configured for multiclass classification corresponding to the number of tutor labels.

3.3.4 BERT approach + Educated guess

Another experimental approach we tried for track 5 was to take the predictions of the BERT + XGBoost model and, based on the distribution of the predicted labels, guess some tutor identities. Under the premise that if the model predicted correctly the majority of the time the correct tutor for a certain label, then taking the same prediction for a label might improve the performance of the model. Therefore, for this approach we modified the predictions of BERT + XGBoost forcing to always classify Tutor9 as “Novice”, Tutor2 as “Mistral” and Tutor3 as “Llama31405B”.

Unfortunately, this approach turned out to perform poorly in comparison to the BERT + XGBoost one, denoting that the labels shown in the test dataset might not have a direct mapping with the actual classes.

3.4 Fine-tuning autoregressive LM

In addition to using encoder-only transformers such as BERT, we also experimented with decoder-only LMs. For these experiments, we only focused on the first four tracks. Although these tracks are framed as classification and are therefore usually better suited to encoder-only architectures, we wanted to compare BERT-style fine-tuned models with similarly sized, fine-tuned autoregressive LMs. Specifically, we used Qwen2.5-0.5B-Instruct⁵ (Team, 2024; Yang et al., 2024), which has 494 million parameters and has undergone instruction tuning.

3.4.1 Training

We performed full fine-tuning on our train partition. Each example was converted into a prompt following the prompt template adopted during the model’s instruction tuning phase. The prompt (available in Appendix A) consists of:

- **System prompt:** We used the same system prompt reported in the shared-task dataset paper (Maurya et al., 2025), which was also used for LLM-based evaluation.

⁴<https://huggingface.co/intfloat/multilingual-e5-large-instruct>

⁵<https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct>

- **User message:** This part contains the conversation history, a task-specific rubric, and the response to be evaluated. The rubrics are the same as those used in the dataset paper.
- **Assistant message:** This consists solely of the class label corresponding to the example (**Yes / No / To some extent**).

We experimented with two different training approaches:

- **Dimension-specific approach:** This involves training four separate models, each dedicated to one of the four evaluation dimensions (mistake identification, mistake location, providing guidance, and actionability). Each model is trained only on examples corresponding to its specific dimension.
- **Multi-dimension approach:** This involves training a single model using the combined training data from all four dimensions. The model is expected to infer the appropriate evaluation criteria based on the scoring rubric included in the user message.

The multi-dimension approach may help mitigate the class imbalance present in certain dimensions (particularly mistake identification), as the model is exposed to a more balanced distribution of the three class labels across different contexts.

The model was trained for three epochs with a batch size of 8 and a learning rate set to 2×10^{-4} , using a linear scheduler with a warm-up ratio of 0.03 and weight decay of 0.001. The training objective was next-token prediction, the same as in pre-training.

To train these models, we used the ClusterUY infrastructure (Nesmachnow and Iturriaga, 2019) with limited (and usually interrupted) access to NVIDIA A100 and NVIDIA A40 GPUs.

3.4.2 Inference

Once the models are fine-tuned, we perform inference using greedy decoding. The input prompt includes the system prompt and the user message, and the model is tasked with generating the assistant message. Since these are classification tasks, we perform a single forward pass and select the class label whose first token receives the highest logit value. Only the three candidate tokens (corresponding to the possible class labels) are considered, and the rest are ignored. This approach

prevents hallucinations by constraining the model to produce one of the predefined labels.

We observed that with the previous method, the **To some extent** label was often under-predicted in favor of the **Yes** or **No** labels. To address this, we introduced an alternative method using thresholds defined separately for each dimension. We retrained the multi-dimension model (i.e. a single model for the first four tracks) on a subset of the training data and used the remaining examples as a validation set to tune the thresholds. The training/validation split was 80/20.

Using the fine-tuned model’s predictions on the validation set, we computed the average probability of each class, grouped by the predicted label. Based on these statistics, we manually defined threshold rules using only the predicted probabilities for the **Yes** and **No** labels. Table 4 in Appendix B shows the threshold values we chose.

3.5 Results obtained over the dev set

We evaluated all these models on the dev set and the results are shown in Table 1.

The first observation we want to do is that even when classical ML algorithms did not manage to be the best in any track, they are still competitive. Some of them even achieved good performances, sometimes getting closer to the best model in the track. Secondly, we want to highlight that some sentence embeddings approaches performed better than using the fine-tuned Llama3.1 8B that we considered in the preliminary experiments. Finally, as expected, neural models performed the best.

Another interesting observation is that the Llama 3.1 8B LoRA fine-tuning, a model 16 times larger than Qwen and BERT, did not achieve significantly better results. In some dimensions, such as providing guidance and actionability, it even performed worse than the fine-tuned Qwen.

Overall, the best models in each track were as follows:

- Track 1 (Mistake Identification) – Sentence Embeddings and k-NN, using the balanced dataset
- Track 2 (Mistake Location) – Fine-tuning DistilBERT with thresholds
- Track 3 (Providing Guidance) – Fine-tuning Qwen using the multi-dimension approach and thresholds

Approach	Track 1		Track 2		Track 3		Track 4		Track 5	
	Mistake identification		Mistake location		Guidance		Actionability		Tutor identity	
	F1-macro	Accuracy	F1-macro	Accuracy	F1-macro	Accuracy	F1-macro	Accuracy	F1-macro	Accuracy
Preliminary experiments										
Always Yes	30.26	83.13	26.42	65.66	22.98	52.61	22.16	49.80	–	–
Random	23.95	33.73	26.94	31.93	30.02	31.33	28.99	30.72	11.75	12.05
Gemini Flash 2.0 Lite	50.74	76.71	48.49	62.25	46.54	55.62	–	–	–	–
Llama 3.1 8B (LoRA)	74.41	92.37	50.68	76.71	51.83	63.25	55.90	72.09	–	–
Classical Machine Learning										
RandomForest + TF-IDF (1-5)grams	71.46	90.76	47.24	74.10	44.08	60.64	52.00	64.26	70.25	69.68
RandomForest + TF-IDF (1-7)grams	60.14	87.55	40.17	60.64	40.93	52.21	42.00	58.23	69.66	68.47
RandomForest + TF-IDF (1-8)grams	71.04	90.56	46.92	74.30	44.37	60.44	50.19	62.25	68.13	66.87
SVC + TF-IDF (1-2)grams	71.82	91.37	49.26	75.50	43.10	61.04	48.42	65.06	79.16	77.71
SVC + TF-IDF (2-5)grams	60.47	88.96	43.74	73.96	40.68	60.64	40.92	60.04	74.76	73.69
k-NN ($k = 7$) + TF-IDF (1-2)grams	73.24	91.16	46.82	71.49	49.52	60.64	50.91	59.84	60.11	59.24
Fine-tuning DistilBERT										
DistilBERT	58.37	90.76	50.30	76.91	42.24	61.24	57.01	68.47	86.51	85.94
DistilBERT (thresholds)	63.04	88.35	56.30	67.47	52.22	55.22	53.02	66.67	–	–
BERT Embeddings + XGBoost	–	–	–	–	–	–	–	–	87.74	87.14
Sentence Embeddings										
e5 (response) k-NN ($k = 9$)	74.93	91.77	48.06	76.69	47.40	58.84	48.55	58.84	–	–
e5 (resp+hist) MLP	72.82	90.96	54.17	75.30	52.51	60.44	56.42	65.06	–	–
e5 (resp+hist+llm) MLP	73.54	91.16	54.36	75.30	52.10	59.84	56.51	65.46	–	–
e5 (resp) k-NN (balanced)	79.16	92.37	47.82	71.08	52.85	56.83	49.08	50.00	–	–
Fine-tuning Qwen										
Qwen (dimension-specific)	74.73	92.17	48.30	74.70	52.71	63.05	61.20	73.29	–	–
Qwen (multi-dimension)	73.04	91.37	51.96	74.50	53.26	61.45	59.57	68.47	–	–
Qwen (thresholds)	65.58	81.33	54.92	64.06	54.18	55.82	55.82	58.84	–	–

Table 1: Results for the five tracks over our dev set. In bold, the best results according to each metric for each track.

- Track 4 (Actionability) – Fine-tuning Qwen using the multi-dimension approach
- Track 5 (Tutor identification) – BERT embeddings and XGBoost

4 Final submissions and experimental analysis

After evaluating all the previously described models on our dev set, we chose those which had the best performance, trying to ensure that at least one model of each category (*Classical machine learning*, *Fine-tuning DistilBERT*, *SentenceEmbeddings* and *Fine-tuning Qwen*) was used to predict the test instances in most of the tracks. The classical ML models were trained from scratch using both our train and dev set, while the neural models were only fine-tuned using the train set.

Table 2 shows the performance of our models in each track, the results we obtained and the resulting ranking position (#). To better understand the performance of our systems, we also considered quartiles for each track, and they are included in the table under the “Q” column. Taking a first look at the quartiles, we can see that none of our models was competitive enough to climb the rankings and finish in the first quartile. However, we want to highlight that in three out of the five tracks (Track1, Track3 and Track4) our models managed to finish in Q_2 .

Overall, as when evaluating on the dev set, this

Submission	F1-macro	Accuracy	#	Q
Track 1 - Mistake identification				
e5 (resp.) k-NN (balanced)	65.35	84.49	56/153	Q_2
Qwen (dimension-specific)	64.94	86.68	62/153	Q_2
DistilBERT (thresholds)	64.30	85.20	64/153	Q_2
SVC + TF-IDF	59.11	84.81	104/153	Q_3
e5 (response) k-NN ($k = 9$)	58.39	84.36	110/153	Q_3
Track 2 - Mistake location				
DistilBERT (thresholds)	49.58	58.63	47/86	Q_3
Qwen (multi-dimension)	49.52	70.78	49/86	Q_3
e5 (resp+hist) MLP	49.40	67.36	51/86	Q_3
Qwen (thresholds)	49.13	55.20	54/86	Q_3
SVC + TF-IDF	45.85	70.39	72/86	Q_4
Track 3 - Providing guidance				
Qwen (multi-dimension)	50.49	59.47	36/105	Q_2
DistilBERT (thresholds)	49.19	53.85	48/105	Q_2
Qwen (thresholds)	47.53	50.36	64/105	Q_3
k-NN + TF-IDF	47.41	59.21	66/105	Q_3
e5 (resp+hist) MLP	47.14	57.85	71/105	Q_3
Track 4 - Actionability				
Qwen (dimension-specific)	61.28	70.33	42/87	Q_2
Qwen (multi-dimension)	60.54	68.00	46/87	Q_3
e5 (resp+hist) MLP	56.37	63.22	60/87	Q_3
DistilBERT (thresholds)	52.61	64.12	68/87	Q_4
RandomForest + TF-IDF	51.91	62.64	70/87	Q_4
Track 5 - Tutor identification				
BERT + XGBoost	83.85	84.74	27/54	Q_3
DistilBERT	83.85	84.74	28/54	Q_3
SVC + TF-IDF	80.44	80.22	39/54	Q_3
BERT + Educated guess	68.16	68.65	42/54	Q_4

Table 2: Results for the five tracks over the competition’s test data. The “#” column indicates the position the system got in the rankings, and the “Q” column indicates the quartile related to that position (splitting in 4 buckets the number of participants in each track).

time the neural models again achieved the best performance among our models. Moreover, something interesting to observe is that the fine-tuned Qwen

Track	Rank / Total	Q	Δ Exact F_1	Δ Exact Accuracy	Δ Lenient F_1	Δ Lenient Accuracy
Track 1	23 / 44	Q_3	71.81 – 65.35 = 06.46	86.23 – 84.49 = 01.74	89.57 – 83.95 = 05.62	94.57 – 91.92 = 02.65
Track 2	21 / 32	Q_3	59.83 – 49.59 = 10.24	76.79 – 58.63 = 18.16	83.86 – 72.00 = 11.86	86.30 – 76.08 = 10.22
Track 3	17 / 35	Q_3	58.34 – 50.49 = 07.85	66.13 – 59.47 = 06.66	77.98 – 70.57 = 07.41	81.90 – 77.51 = 04.39
Track 4	17 / 29	Q_3	70.85 – 61.29 = 09.56	72.98 – 70.33 = 02.65	85.27 – 82.72 = 02.55	88.37 – 85.59 = 02.78
Track 5	12 / 20	Q_3	96.98 – 83.85 = 13.13	96.64 – 84.75 = 11.89	N/A	N/A

Table 3: Performance difference between our best submissions and the winners, for each task. This table was built based on the *team results*, so the total number of submissions for each track is always fewer than those considered in Table 2.

models and the BERT models got similar performance. This seems to indicate that the generative capabilities of Qwen are good enough to also work as an emergent classifier.

Most of the models we used are not well-suited for handling long contexts. This led us to question how essential the *conversation history* truly is for assessing the four evaluation dimensions, or whether the model’s response alone is enough to obtain good results. Therefore, we tested training some models both with and without including the conversation history as input. In this regard, the most significant experiments were those using sentence embeddings. These experiments show that nearly every dimension benefits from the inclusion of history, except for the mistake identification dimension, which performs notably better without it (i.e. using only the tutor’s response). More broadly, Qwen-based methods (which incorporate the full conversation history) achieve the best results in providing guidance and actionability, and, in contrast, BERT-based methods (which do not use the conversation history) perform better on mistake location. This pattern suggests that more subtle dimensions like guidance and actionability benefit more from access to the full conversational context. Further experimentation is required to validate all these preliminary observations.

Finally, while the DistilBERT with XGBoost approach seemed to have a good performance on our dev set, its final performance (on the test set) was identical to that of the DistilBERT fine-tuning model (without XGBoost). This was not the only difference we had between our dev set and the test set. As can be seen by comparing Tables 1 and 2, most methods performed noticeably better on our internal dev set than on the test set. We believe this performance gap may be due to differences in the class distributions between the two sets.

Furthermore, the experiment using under-sampling to balance the classes showed a significant improvement on the test set, going from

being the worst-performing submission to being the best one. This further highlights the impact of class imbalance on model performance.

4.1 How far were these lightweight models from winning?

Finally, to answer our *research question*, we wanted to check how far our lightweight models went in the shared task. Beyond the ranking positions, we wanted to focus on *how big* (according to the official scores) was the gap between these models and those that settled the state of the art, winning the competition. In Table 3 we show the difference (Δ) — for each metric — of our best predictions with the winner team in each track⁶. As a reference, we also include our team’s position in that track and the correspondent *quartile* (this time, based on the number of teams, and not on the number of submitted systems).

Taking a look at the table, we can see that, according to Δ Exact F_1 , the closest gap between our performance and the winning team was 06.46 (in Track 1), while the biggest gap was 13.13 (in Track 5). We think this difference in performance is very small considering the restrictions we had.

5 Conclusions

In this paper we presented the RETUYT-INCO participation at the 2025 BEA shared task, characterized by our self-imposed restriction of only using models under 1B parameters. Although our research lab have access to cheap API LLMs and very limited access to run 7B LLMs on clusters, we are conscious that this is not the case for other research labs in the Global South, that usually work

⁶Since the organizers considered the *Exact F_1* metric as the main one, we considered as *winning teams* those which got the highest score according to that metric. Therefore, for all metrics, we calculated the Δ according to the score achieved by the winning team in that track. This way, even if a different team got a better result according to other metric, we still calculated the Δ according to the winning team.

in even deeper under-resourced scenarios. Our self-imposed restriction tries to represent this scenario.

Overall, we used classical machine learning models, BERT-based models, and a QWEN 0.5B LLM. Despite their (very small) size we finished in mid-ranking positions. Beyond the results in the rankings, the result we want to highlight is that the gaps in performance we had with the winning teams were between 6.46 and 13.13 F_1 *exact* points.

We find that gap surprisingly small, taking into account that we did not use LLMs bigger than 1B, nor paid for API access, nor paid for premium cloud computing, nor needed top-tier resources to run our experiments. Additionally, following the environmental concerns that surround the carbon footprint of state-of-the-art LLMs (Luccioni et al., 2023; Faiz et al., 2024; Liu and Yin, 2024), we consider this an interesting tradeoff: to sacrifice some performance, in order to have models that do not need extensive training/inference time or power, but that are still competent. Based on all of the above, we think research on models that run on low-cost GPUs — or need no GPU at all — should definitely go on.

6 Limitations

Throughout the paper we have outlined several limitations we have to run experiments with large models. These constraints led to our self-imposed restriction of using only neural models with fewer than 1B parameters. Naturally, our work does not present state-of-the-art results, nor does it intend to. Furthermore, we prioritized breadth (i.e. trying many model types) over depth (i.e. optimizing a single approach or architecture extensively). While this gives a broader perspective on the diverse possibilities that lightweight models have to offer, it may have limited the performance ceiling of individual models.

Regarding our methodology, we made the decision of splitting the full set into two subsets (train and dev) considering as a priority to keep the conversations and their responses in the same subset. This decision may have introduced some noise and class imbalance, since we found remarkable differences in the performance of our models over the dev set and the final test set (after submission). Since the fine-tuned models and the thresholds used were adjusted specifically to our dev set, they may not generalize well to other similar corpora.

Finally, and related to the previous considerations, we did not systematically perform hyperparameter tuning due to both hardware and time limitations. Additionally, prior to our final submissions, we only trained (from scratch) the classical ML models on the full set (our train + dev sets). Since the neural models were our best approaches, searching better hyperparameters and training them with more data could have made the performance gaps a bit smaller.

Acknowledgments

This paper has been funded by ANII (Uruguayan Innovation and Research National Agency), Grant No. *FMV_1_2023_1_176581*.

References

- AI@Meta. 2024. [Llama 3 model card](#).
- 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.
- 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.
- 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.
- 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.
- Ahmad Faiz, Sotaro Kaneda, Ruhan Wang, Rita Chukwunyerere Osi, Prateek Sharma, Fan Chen, and Lei Jiang. 2024. [Llmcarbon: Modeling the end-to-end carbon footprint of large language models](#). In *The Twelfth 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*.
- Oana Ignat, Zhijing Jin, Artem Abzaliev, Laura Bester, Santiago Castro, Naihao Deng, Xinyi Gao, Aylin Ece Gunal, Jacky He, Ashkan Kazemi, Muhammad Khalifa, Namho Koh, Andrew Lee, Siyang Liu, Do June Min, Shinka Mori, Joan C. Nwatu, Veronica Perez-Rosas, Siqi Shen, Zekun Wang, Winston Wu, and Rada Mihalcea. 2024. [Has it all been solved? open NLP research questions not solved by large language models](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8050–8094, Torino, Italia. ELRA and ICCL.
- Ekaterina Kochmar, Kaushal Kumar 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*.
- Vivian Liu and Yiqiao Yin. 2024. Green ai: exploring carbon footprints, mitigation strategies, and trade offs in large language model training. *Discover Artificial Intelligence*, 4(1):49.
- Alexandra Sasha Luccioni, Sylvain Viguiet, and Anne-Laure Ligozat. 2023. Estimating the carbon footprint of bloom, a 176b parameter language model. *Journal of Machine Learning Research*, 24(253):1–15.
- 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.
- Sergio Nesmachnow and Santiago Iturriaga. 2019. Cluster-uy: Collaborative scientific high performance computing in uruguay. In *Supercomputing*, pages 188–202, Cham. Springer International Publishing.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- 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.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. [Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter](#). *Preprint*, arXiv:1910.01108.
- 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.
- 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 Sag-gion. 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.
- 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.
- Qwen Team. 2024. [Qwen2.5: A party of foundation models](#).
- 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*.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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

A Prompts

This Appendix presents the prompts used for fine-tuning the decoder-only language models, as explained in Section 3.4.

System prompt

You are a critic evaluating a tutor's interaction with a student, responsible for providing a clear and objective single evaluation based on specific criteria. Each assessment must accurately reflect the absolute performance standards.

User message

```
# Previous Conversation between Tutor and Student:
{history}
```

```
# Scoring Rubric:
{rubric}
```

```
# Tutor Response:
{response}
```

Mistake identification rubric

[Has the tutor identified a mistake in a student's response?]

- 1) Yes
- 2) To some extent
- 3) No

Mistake location rubric

[Does the tutor's response accurately point to a genuine mistake and its location?]

- 1) Yes
- 2) To some extent
- 3) No

Providing guidance rubric

[Does the tutor offer correct and relevant guidance, such as an explanation, elaboration, hint, examples, and so on?]

- 1) Yes (guidance is correct and relevant to the mistake)
- 2) To some extent (guidance is provided but it is fully or partially incorrect or incomplete)
- 3) No

Actionability rubric

[Is it clear from the tutor's feedback what the student should do next?]

- 1) Yes
- 2) To some extent
- 3) No

B Qwen Thresholds

Table 4 presents the thresholds used with the Qwen model, as explained in Section 3.4.

Dimension	Yes condition	No condition	TSE condition
Mistake Identification	Yes > 0.90 & No < 0.05	Yes < 0.40 & No > 0.50	Otherwise
Mistake Location	Yes > 0.75 & No < 0.15	Yes < 0.42 & No > 0.50	Otherwise
Providing Guidance	Yes > 0.65 & No < 0.12	Yes < 0.35 & No > 0.45	Otherwise
Actionability	Yes > 0.70 & No < 0.14	Yes < 0.25 & No > 0.65	Otherwise

Table 4: Threshold-based classification rules for each evaluation dimension using Qwen. TSE = "To some extent".

K-NLPers at BEA 2025 Shared Task: Evaluating the Quality of AI Tutor Responses with GPT-4.1

Geon Park^{*1}, Jiwoo Song^{*2}, Gihyeon Choi^{*2}, Juoh Sun^{*2} and Harksoo Kim^{1,2},

¹Department of Computer Science and Engineering, Konkuk University

²Department of Artificial Intelligence, Konkuk University

Correspondence: nldrkim@konkuk.ac.kr

Abstract

This paper presents automatic evaluation systems for assessing the pedagogical capabilities of LLM-based AI tutors. Drawing from a shared task, our systems specifically target four key dimensions of tutor responses: Mistake Identification, Mistake Location, Providing Guidance, and Actionability. These dimensions capture the educational quality of responses from multiple perspectives, including the ability to detect student mistakes, accurately identify error locations, provide effective instructional guidance, and offer actionable feedback. We propose GPT-4.1-based automatic evaluation systems, leveraging their strong capabilities in comprehending diverse linguistic expressions and complex conversational contexts to address the detailed evaluation criteria across these dimensions. Our systems were quantitatively evaluated based on the official criteria of each track. In the Mistake Location track, our evaluation systems achieved an Exact macro F1 score of 58.80% (ranked in the top 3), and in the Providing Guidance track, they achieved 56.06% (ranked in the top 5). While the systems showed mid-range performance in the remaining tracks, the overall results demonstrate that our proposed automatic evaluation systems can effectively assess the quality of tutor responses, highlighting their potential for evaluating AI tutor effectiveness.

1 Introduction

Recent advancements in Large Language Models (LLMs) have significantly enhanced performance across various tasks in natural language processing and artificial intelligence (Kim et al., 2025, 2024; Das et al., 2025). These developments have spurred interest in applying LLMs within educational settings, aiming to leverage their capabilities for personalized learning, intelligent tutoring, and educational assessment (Macina et al., 2023; Chevalier et al., 2024; Wang et al., 2024b; Gan et al.,

2023). However, despite these promising developments, how well LLMs can provide educational feedback and guidance in authentic tutoring scenarios remains underexplored. To address this gap, there is a growing need for systematic evaluation methods that can rigorously assess the pedagogical quality of LLM-generated tutor responses. To address this need, we participated in the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI Tutors (Kochmar et al., 2025), which aims to systematically evaluate the educational quality of AI tutor responses across multiple dimensions. Our system was submitted to four subtasks (Tracks 1–4), each corresponding to the pedagogical evaluation dimensions defined in the shared task: Mistake Identification, Mistake Location, Providing Guidance, and Actionability.

For this study, we employed prompting techniques using GPT-4.1¹ model to complete each evaluation task. GPT-4.1 is well-known for its superior ability in instruction-following, handling complex contexts, and performing multi-step reasoning (OpenAI, 2025). These capabilities, combined with our prompting strategies, enabled effective evaluation of tutoring performance. This paper details the prompting strategies and methodologies utilized in the evaluation tracks in which we participated. Additionally, we analyze and discuss the performance of our proposed systems, aiming to provide practical insights for future development of LLM-based tutoring systems.

2 Methodology

This section introduces the three prompting-based evaluation systems we developed for the BEA 2025 Shared Task. Each system is designed to align with the pedagogical goals of different evaluation tracks, while sharing a common objective of simulating human-like reasoning in tutoring scenarios.

* These authors contributed equally to this work.

¹[gpt-4.1-2025-04-14](https://openai.com/index/gpt-4-1/)

Section 2.1 presents the Chain-of-Thought-based Pedagogical Evaluation system with Reasoning Layers. This approach models the step-by-step reasoning process of a human tutor, who first analyzes the student’s thinking, constructs a correct solution, and then evaluates the tutor’s response in context. This system was applied to **Track 1**, which focuses on evaluating whether the tutor successfully identifies student mistakes, and **Track 4**, which assesses the actionability of the feedback provided.

Section 2.2 describes the Multi-Perspective Reflective Evaluation, developed for **Track 2**. Inspired by the reflective feedback behavior of human tutors, this method simulates internal deliberation among distinct reasoning perspectives to assess whether the tutor accurately identifies the location of a student’s mistake.

Section 2.3 details the Rubric-Based Evaluation Method, which targets **Track 3**. This approach decomposes the Providing Guidance criterion into multiple rubric-based sub-questions. It extracts structured features from LLM-generated probability distributions and enhances scoring consistency by using a downstream classifier trained to align model judgments with human evaluation patterns.

These methodologies establish a comprehensive framework for evaluating tutor responses, enhancing interpretability, pedagogical alignment, and educational validity in open-ended dialogue settings.

2.1 Chain-of-Thought-based Pedagogical Evaluation with Reasoning layers

This system is designed to automatically evaluate the pedagogical appropriateness of a tutor’s final utterance in a math lesson dialogue. Instead of using a single prompt or a simple classification-based approach, we adopted a step-by-step processing structure that emulates how human tutors interpret student solutions and determine appropriate feedback. The design of this structure is based on two key observations: First, large language models (LLMs) show improved performance on complex problems when explicitly guided through intermediate reasoning steps, a technique known as chain-of-thought prompting (Wei et al., 2022). Second, LLMs tend to exhibit conformity bias—favoring only a single "standard" solution path and struggling to respond appropriately to diverse or alternative reasoning strategies (Li et al., 2024).

To address these issues, we designed the flow so that the model first analyzes the student’s reason-

ing process, then generates a correct solution path based on that reasoning, and finally evaluates the tutor’s utterance in light of the student’s thinking. All stages are implemented using the GPT-4.1 model, which was selected for its strong performance in instruction following, conversational context retention, and multi-step reasoning. Our proposed system is composed of the following four stages:

1. **Problem Extraction:** Extract the math problem from the dialogue. The extracted problem serves as the foundation for all subsequent reasoning, and functions as a critical preprocessing step for maintaining contextual coherence and semantic consistency.
2. **Student Reasoning Process Reconstruction:** Based on the student’s response and the flow of the conversation, reconstruct the reasoning path that the student followed to solve the problem. Even in the absence of explicit explanations, infer a plausible line of reasoning. This mirrors how a human tutor might infer a student’s thought process in real instructional settings to provide targeted feedback.
3. **Correct Reasoning Process Generation:** Using the reconstructed student reasoning as a foundation, generate a correct solution path. If the student’s approach is partially valid, it is preserved and only the errors are corrected. If the approach is fundamentally flawed, a new solution is generated. This stage serves both as a reference point for comparison and as a mechanism to mitigate the conformity bias described earlier.
4. **Tutor Response Evaluation:** Finally, the tutor’s final utterance is evaluated using the following four criteria:
 - Mistake Identification
 - Mistake Location
 - Providing Guidance
 - Actionability

These criteria, while based on the definitions provided by the task organizers, are redefined in our approach to focus on how utterances actually function within the student’s learning process, moving beyond simple sentence-level evaluation. **Mistake Identification** is judged not merely on whether a mistake was mentioned, but on whether this recognition was perceptible and significant to the student. **Mistake Location** is assessed not by whether the error’s position is explicitly stated, but by whether

the student can reasonably infer where the mistake occurred based on the tutor’s response. **Providing Guidance** is assessed not by the mere provision of a correct answer, but by whether it was a method that stimulated and broadened student thinking. **Actionability** uses as its criterion whether the student can actually understand and follow the guidance, rather than the mere presence or absence of a suggested action.

These redefinitions allow the LLM to evaluate the tutor not as a mere provider of correct answers (tutor-as-answerer), but as a facilitator of reasoning and learning (tutor-as-guide). To ensure consistent scoring across levels, especially for nuanced categories like "To some extent", concrete judgment criteria were clearly designed. The specific prompts corresponding to each criterion, along with detailed evaluation guidelines, are provided in detail in the Appendix A.

2.2 Multi-Perspective Reflective Evaluation Method

To accurately determine whether a tutor’s response correctly identifies the location of a mistake in a student’s solution, our system proposes a multi-perspective reasoning process inspired by how human tutors approach student feedback. Rather than relying on static classification, this system simulates a dynamic reasoning process, decomposing the evaluation into distinct functional perspectives such as recalling relevant context, analyzing logic, assessing clarity, and monitoring emotional tone.

2.2.1 Human-Like Multi-Step Reasoning

When human tutors assess a student’s response, they typically do not evaluate it in a single step. Instead, they engage in a layered cognitive process: understanding the problem, reconstructing the student’s reasoning, identifying discrepancies, and delivering feedback that balances correctness and pedagogical clarity. One recent attempt to emulate this human-like multi-step reasoning within a single LLM is **Solo Performance Prompting (SPP)**, which activates diverse personas to facilitate self-collaboration and mimic human reasoning (Wang et al., 2024c). Building on this idea, our system mirrors such behavior by simulating a group of internal “reasoning participants”, each representing a specific evaluative function. These participants collaborate iteratively to reach a decision regarding the quality of the tutor’s feedback.

2.2.2 Reasoning Process

Given a conversation history, including the original question, the student’s response, and the tutor’s follow-up, the system performs the following steps:

- **Perspective Initialization:** Depending on the complexity of the student’s reasoning and the characteristics of the tutor’s feedback, a set of internal perspectives is dynamically activated. These perspectives represent distinct reasoning roles (e.g., logical analysis, memory retrieval, contextual interpretation).
- **Independent Assessment:** Each perspective independently analyzes whether the tutor’s response points to the specific step where the mistake occurred. The analysis includes not only factual correctness but also the interpretability and relevance of the feedback.
- **Collaborative Deliberation:** After the initial assessments, the perspectives engage in a multi-turn collaborative discussion. They provide critical feedback on one another’s reasoning, refine interpretations, and critique or support conclusions.
- **Final Decision:** Based on this internal collaboration, the system synthesizes a final judgment: "Yes", "To some extent", or "No", depending on how clearly and precisely the tutor’s response identifies the location of the student’s mistake.

2.2.3 Prompting Strategy

We implement the above reasoning process through a carefully designed prompt that guides the language model to simulate human-like evaluation. Rather than instructing the model to directly respond to a tutor’s utterance, the prompt breaks the evaluation into distinct reasoning roles. It encourages the model to adopt multiple perspectives. The prompt explicitly instructs the model to initiate internal reflection by assigning roles such as logical analysis, memory recall, and clarity evaluation. It then simulates a collaborative discussion where these roles critique and refine one another’s views before converging on a final judgment. This structured interaction is carried out entirely within a single language model, enabling it to reason through the task in a self-contained yet multi-faceted manner. By prompting the model to consider both explicit and implicit forms of feedback, as well as emotional tone and pedagogical clarity, this design elicits more interpretable and human-aligned judgments. It ensures the model reflects on why a tutor

response is effective or not, rather than simply what label to assign.

2.3 Rubric Based Evaluation Method

In this section, we aim to evaluate whether tutor LLMs provide correct and relevant guidance within the context of tutoring dialogue. We apply a method that predicts high-dimensional judgments through item-specific probability distributions, such as those used in LLM-Rubric (Hashemi et al., 2024), to assess the educational validity of tutor LLM responses. Specifically, for the prediction of Providing Guidance, we designed five detailed questions (Q_{rubric}) and a single comprehensive question ($Q_{overall}$) utilizing statistical information. For each item, we constructed prompts such that the LLM outputs the probabilities of "Yes", "To some extent", "No". However, the evaluation labels generated by the LLM may not completely align with the labels of human evaluators. Therefore, we use the item-wise probability distributions as input features for a subsequent classifier, aiming to calibrate the LLM's judgments to be more consistent with human evaluation.

2.3.1 Feature Extraction via Structured Prompting

The feature extraction step based on structured prompting consists of two components: rubric-based evaluation criteria and statistical information. The prompts used for each task are presented in Appendix B, and the responses were generated using the GPT-4.1 model.

Feature Extraction from Rubric-Based Evaluation Criteria The prompt for feature extraction based on rubric-defined evaluation items consists of role specification, presentation of dialogue context, definition of label criteria and output format, and a list of evaluation questions.

- **Role specification:** By assigning the expert role of "expert evaluator analyzing a tutor's response in a learning dialogue," the model is encouraged to think critically from the perspective of an evaluator rather than as a simple generator.
- **Presentation of dialogue context:** The dialogue context is presented sequentially and consists of the entire conversation between the tutor and student, the student's last utterance, and the tutor's response to that utterance. This allows the LLM

to conduct evaluations based on a sufficient understanding of the context.

- **Definition of label criteria and output format:** For each item, the judgment consists of three options: Yes, To some extent, and No. The definitions of these labels are based on criteria defined by the annotator. For each item, the model outputs the probability value for each of the three labels in decimal form, based on the rationale for its judgment. The sum of all probability values is designed to be 1.0, and these values are used as input features for the subsequent classifier.
- **List of questions:** The prompt includes five questions designed to capture various aspects of the Providing Guidance criterion. Each question is constructed to evaluate specific elements of detailed feedback, as follows.
 - Q1 Did the tutor attempt to provide any explanation, hint, or example?
 - Q2 Was the guidance factually correct and appropriate given the student's error?
 - Q3 Did the tutor's response directly address the student's specific mistake?
 - Q4 Did the guidance help the student figure out what to do next, without directly giving the final answer?
 - Q5 Was the tutor's response clear and unlikely to confuse the student?

This prompt design enables the LLM to consistently perform structured evaluations. The extracted features serve as inputs for subsequent classifiers, thereby enhancing the precision and reliability of the automated evaluation framework.

Feature Extraction Using Statistical Information

Antecedent	Consequent	Support	Confidence	Lift
ML = Yes	MI = Yes	0.6163	0.9889	1.2674
MI = Yes	ML = Yes	0.6163	0.7898	1.2674
PG = Yes	MI = Yes	0.5399	0.9502	1.2178

Table 1: Results of Association Rule Analysis among Mistake Identification, Mistake Location, and Providing Guidance

In the feature extraction step utilizing statistical information, features were constructed based on association rules among items analyzed from the development dataset. To this end, the Apriori algorithm (Agrawal and Srikant, 1994) was applied

using the label information of the three items: Mistake Identification, Mistake Location, and Providing Guidance. Based on the calculated support and confidence values, the three most reliable association rules were extracted. The main association rules derived from this analysis are shown in Table 1. The top three association rules all exhibit high confidence values, suggesting that the relationships between items possess meaningful associations beyond mere coincidence. Among these, the relationship between "Mistake Location = Yes" and "Mistake Identification = Yes" demonstrates particularly high confidence, confirming a strong association between the two items.

To reflect these statistically significant associations, the following elements were added within the same prompt structure used in previous tasks:

- **Insertion of prior prediction results:** Predictions from previous Tracks (Mistake Identification and Mistake Location) were included in the prompt, allowing the LLM to perform evaluations based on this prior knowledge.
- **Provision of statistical associations:** Confidence values derived from association analysis were explicitly presented in the prompt to numerically illustrate conditional relationships among the three items. This allows the model to reference the likelihood of specific judgments influencing others during response evaluation.
- **Presentation of a single comprehensive question:** A question designed to elicit an overall assessment of "Providing Guidance" was included. The model was prompted to holistically assess whether the response attempted meaningful guidance, based on the given content, prior Track predictions, and statistical information.

Through this prompt, the model can extract comprehensive judgment features for Providing Guidance by simultaneously considering both existing prediction results and quantitative association information.

2.3.2 Improving Consistency in LLM Evaluation Using Classifiers

To calibrate the evaluation results of LLM responses with human assessors' judgments, we adopted an approach utilizing a subsequent classifier and conducted experiments to select an optimal classification model. The features extracted

in Section 2.3.1 consist of probability values for the three categories—"Yes," "To some extent," and "No"—for each of the six sub-questions (Q_{rubric} and $Q_{overall}$). Each sub-question is represented as a three-dimensional vector (i.e., three probabilities), and concatenating these yields an 18-dimensional real-valued vector (6 questions \times 3 classes = 18), which serves as the input feature for the response quality classifier. We compared the performance of three classification models — Random Forest (Breiman, 2001), Logistic Regression (Cox, 1958), and XGBoost (Chen and Guestrin, 2016) — using 5-fold cross-validation on the development dataset.

Classifier	Gold		Pred		w/o $Q_{overall}$	
	F1	Acc	F1	Acc	F1	Acc
Logistic Regression	0.61	0.71	0.49	0.66	0.49	0.66
Random Forest	0.63	0.70	0.55	0.61	0.54	0.60
XGBoost	0.61	0.69	0.52	0.66	0.50	0.64

Table 2: Comparison of Classifier Performance According to the Use of $Q_{overall}$. **Gold** denotes that ground-truth labels for $Q_{overall}$ were supplied, whereas **Pred** uses the values predicted by the methods in Sections 2.1 and 2.2.

Table 2 compares classifier performance across three experimental conditions. The first condition inputs gold labels for Mistake Identification and Mistake Location into $Q_{overall}$. The second condition uses predicted values from previous Tracks for $Q_{overall}$, while the third entirely excludes $Q_{overall}$ from input features to analyze its performance impact. Under the gold-label configuration for $Q_{overall}$, Random Forest achieved the highest Macro F1 score of 0.63. This indicates strong alignment between $Q_{overall}$ and the final Providing Guidance label, representing the upper performance bound of the proposed methodology. In the simulated test environment using predicted values for $Q_{overall}$, Random Forest maintained superior performance with a Macro F1-score of 0.55, though lower than the gold-label scenario. This performance gap underscores the influence of prediction uncertainty in $Q_{overall}$ and highlights its critical role in overall accuracy. Experiments excluding $Q_{overall}$ resulted in performance degradation across all models, demonstrating that $Q_{overall}$ facilitates comprehensive judgment rather than isolated item assessment, thereby making substantial contributions to classifier efficacy.

Based on these findings, the study implemented a system incorporating all Q_{rubric} and $Q_{overall}$ items

as input features, with Random Forest selected as the final classifier for Predicting Providing Guidance labels. This configuration optimizes robustness while maintaining practical applicability in automated feedback evaluation.

3 Evaluation

In this section, we report and discuss the evaluation results obtained from each of the prompting methodologies applied to Tracks 1 through 4. The performance of each proposed method is presented briefly, highlighting their strengths and identifying areas for improvement.

3.1 Evaluation Metrics

Our evaluation followed the same metrics defined by the shared task organizers. Specifically, accuracy and macro F1 scores were utilized as the primary metrics for evaluating performance across Tracks 1 through 4. These metrics were computed under two distinct settings:

- **Exact evaluation (Ex.):** Predictions were assessed based on the precise classification into three distinct categories ("Yes," "To some extent," and "No").
- **Lenient evaluation (Len.):** Considering the qualitative similarities between responses annotated as "Yes" and "To some extent," these two classes were combined into a single category ("Yes + To some extent"), resulting in a simplified binary classification ("Yes + To some extent" vs. "No") for performance evaluation.

3.2 Dataset

We conducted our experiments using the dataset provided by the shared task organizers. The dataset consists of 300 dialogues extracted from the MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024a) datasets, and includes a total of 2,476 tutor responses annotated for four pedagogical aspects based on the scheme proposed by Maurya et al. (2025). These annotated responses were used as the development set. An additional 1,547 tutor responses, constructed in the same manner, were used as the test set.

3.3 Chain-of-Thought-based Pedagogical Evaluation System with Reasoning layers

To evaluate the effectiveness of the proposed assessment system in section 2.1, we conducted experi-

ments on two models: GPT-4.1 and GPT-4.1-mini². The experiments were performed on the entire development set. Since the proposed system does not require a separate training phase, all examples in the dataset were directly used for evaluation. To facilitate comparative analysis of the proposed system’s performance, we also conducted experiments using an alternative baseline prompt (see Appendix A for details of the baseline prompt), defined by the following conditions:

- The input consists only of the dialogue history and the tutor’s final utterance.
- For each evaluation criterion, the original definitions provided by the task organizers were used, rather than the redefined versions proposed in this study.

This setup allows us to directly compare how variations in prompt design and evaluation criteria definitions affect final performance, under identical language model and dataset conditions.

Task	Prompt	Model	Ex. F1	Ex. Acc	Len. F1	Len. Acc
MI	Base	GPT 4.1 mini	0.5566	0.6975	0.8037	0.8958
		GPT 4.1	<u>0.5850</u>	<u>0.7383</u>	<u>0.8107</u>	0.9055
	Ours	GPT 4.1 mini	0.5699	0.7282	0.7965	<u>0.9079</u>
		GPT 4.1	0.6225	0.7993	0.8371	0.9204
ML	Base	GPT 4.1 mini	<u>0.5037</u>	<u>0.5856</u>	0.7447	0.7928
		GPT 4.1	0.4642	0.4851	0.7361	0.8029
	Ours	GPT 4.1 mini	0.4885	0.5166	0.7581	<u>0.8146</u>
		GPT 4.1	0.5238	0.5969	<u>0.7564</u>	0.8154
PG	Base	GPT 4.1 mini	<u>0.5286</u>	0.5428	0.7347	0.8247
		GPT 4.1	0.4905	0.4758	0.7374	0.8320
	Ours	GPT 4.1 mini	0.5117	0.4956	<u>0.7506</u>	0.8389
		GPT 4.1	0.5398	<u>0.5355</u>	0.7583	0.8384
ACT	Base	GPT 4.1 mini	0.4934	0.5141	0.6889	0.7597
		GPT 4.1	0.4487	0.4378	0.6975	0.7815
	Ours	GPT 4.1 mini	<u>0.5045</u>	<u>0.5250</u>	<u>0.7129</u>	<u>0.7851</u>
		GPT 4.1	0.5210	0.5384	0.7253	0.7948

Table 3: Performance comparison across tasks, prompts, and models. **Bold** indicates the best performance within each task, and underline indicates the second-best.

Table 3 presents a performance comparison between the proposed evaluation system and the baseline prompt. The proposed approach demonstrates overall superior results across all four evaluation criteria compared to the base prompt. Notably, when using the GPT-4.1 model, improvements in response quality were observed under both exact and lenient evaluation metrics.

For Mistake Identification, which measures the model’s ability to recognize student errors, the proposed system proved more effective in producing

²gpt-4.1-mini-2025-04-14

clear and convincing judgments.

For Mistake Location, which assesses how well the tutor’s response pinpoints where the student made a mistake, the proposed system also showed better performance when using GPT-4.1. Although the performance gains were more limited with the smaller model (GPT-4.1-mini), the proposed system helped generate responses with more consistent error localization patterns.

For Providing Guidance, which evaluates whether the tutor’s response offers not only correct answers but also instructional support — such as explanations, hints, or examples — the proposed system was more effective in assessing responses using this criterion, as it successfully identified a variety of instructional strategies, including explanations, hints, and guiding questions. This indicates that the redefined evaluation criteria were more closely aligned with authentic pedagogical practices.

For Actionability, which assesses whether the student can clearly understand what to do next based on the tutor’s feedback, the proposed system demonstrated consistently high performance in evaluating responses that effectively prompted concrete next steps. This result likely stems from the prompt structure and evaluation criteria, which were explicitly designed to reflect a student-centered communicative framework.

Taken together, these results demonstrate that even without fine-tuning, the combination of a structured prompt chain, evaluation criteria redefined from the student’s perspective, and a reasoning-guided process can enhance both the reliability and pedagogical validity of tutor response evaluation. However, the experimental findings also imply that distinguishing fine-grained judgment boundaries—such as between “Yes”, “To some extent”, and “No”—remains a challenge. This highlights the limitation of relying solely on prompt-based inference, as the model may still struggle to fully grasp the nuanced intent behind each evaluation category without task-specific training.

Despite these limitations, we applied the proposed system to the official evaluations of Track 1 and Track 4 without any additional training, in order to see whether it would perform reliably in a real evaluation setting. As a result, the system maintained stable performance on the test set, achieving Exact macro F1 scores of 0.6669 and 0.5664 for Track 1 and Track 4, respectively, thereby demonstrating that the performance observed on the development set was consistently replicated in the

official evaluation.

3.4 Multi-Perspective Reflective Evaluation System

We submitted our system, developed under the team name K-NLPers, to the Mistake Location track of the BEA Shared Task. It was built upon our proposed multi-perspective reasoning framework and evaluated using the GPT-4.1 model.

Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
BLCU-ICALL	0.5983	0.7679	0.8386	0.8630
BJTU	<u>0.5940</u>	0.7330	0.7848	0.8261
K-NLPers	0.5880	<u>0.7641</u>	0.8404	<u>0.8610</u>
MSA	0.5743	0.6975	0.7848	0.8209
SG	0.5692	0.7602	0.8118	0.8416

Table 4: Evaluation Results on the Mistake Location Track under Multi-Perspective Reflective Evaluation. **Bold** indicates the best performance and underline indicates the second-best.

As shown in Table 4, our system achieved competitive results, ranking 3rd overall among participating teams. In particular, it showed strong performance in Exact Accuracy (0.7641), Lenient macro F1 (0.8404), and Lenient Accuracy (0.8610), with scores closely comparable to those of the top two teams. These results suggest that our system produces predictions with consistent structure and high lexical accuracy, demonstrating that the proposed approach can effectively compete with state-of-the-art systems. However, the Exact macro F1 score (0.5880) was slightly lower than that of the top-ranked teams, primarily due to difficulty in distinguishing responses labeled as “To Some Extent”. Despite this, the results confirm that our system is robust and generalizable, yielding strong overall performance across evaluation metrics in a competitive setting.

3.5 Rubric Based Evaluation System

This section evaluated the Providing Guidance dimension (Track 3) using the rubric-based system proposed in Section 2.3.

Team	Ex. F1	Ex. Acc	Len. F1	Len. Acc
BLCU-ICALL	0.5741	0.6716	0.7487	0.8061
BJTU	0.5725	0.6490	0.7445	0.8100
K-NLPers	0.5606	0.6270	0.7446	0.8003
bea-jh	0.5451	0.6387	0.7253	0.7977

Table 5: Performance and ranking of our models in predicting "Providing Guidance" on the test set.

Table 5 presents the leaderboard results on the test set for the final system proposed in Section 2.3. Our system achieved an Exact macro F1 score of 0.5606 and a Lenient macro F1 score of 0.7446 on the test set. Despite employing a straightforward approach that relies solely on prompt-based probability distribution outputs and a post-processing classifier, the system demonstrates the capability to secure a satisfactory level of precision and consistency in real-world settings. Notably, attaining an Exact macro F1 score of 0.5606—a stringent evaluation criterion—indicates that the structured multi-dimensional features derived from the rubric-based items Q_{rubric} and $Q_{overall}$ effectively capture the educational validity of tutor responses.

These findings suggest that the prompt-based multi-dimensional judgment methodology not only generates responses but also effectively aligns the evaluation and classification of responses with human raters. Furthermore, the methodology maintains a certain degree of generalization performance even on inputs that were not seen during training, thereby illustrating that evaluations leveraging large language models can function as assessments with genuine educational validity.

4 Conclusion

This study proposed a set of prompting-based automatic evaluation methods to assess the pedagogical quality of AI tutor responses across four key dimensions: Mistake Identification, Mistake Location, Providing Guidance, and Actionability. Leveraging the capabilities of GPT-4.1, the methods were designed to emulate human-like reasoning through chain-of-thought prompting, multi-perspective reflection, and rubric-based probability estimation, aligning large language model outputs with authentic educational feedback standards.

Our approaches demonstrated competitive performance in the BEA 2025 Shared Task across multiple evaluation tracks. The Multi-Perspective Reflective Evaluation showed strong performance in Mistake Location, while the Rubric-Based Evaluation validated the effectiveness of structured feature extraction and post-classification for nuanced feedback analysis in Providing Guidance. These findings confirm that prompt engineering—when guided by educational theory and structured evaluation logic—can significantly improve the interpretability and reliability of LLM-based tutor assessments. Although fine-grained distinctions be-

tween evaluation categories remain challenging, the results underscore the feasibility of using large language models for scalable, pedagogically sound evaluation in open-ended educational dialogues.

Future work may explore integrating these methods into real-time tutoring systems, applying task-specific fine-tuning to improve classification sensitivity, and extending the framework to multimodal or domain-specific educational contexts. Ultimately, this line of research contributes to developing AI systems that are not only linguistically fluent but also aligned with human learning objectives.

Limitations

As the proposed methods relies on the model’s internal reasoning to perform evaluations, it may yield interpret evaluation criteria differently depending on the model. This is especially true for intermediate categories such as “To some extent”, where subjective interpretation can lead to ambiguity, indicating a limitation in ensuring the reliability of automatic assessment.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2023-00216011, Development of artificial complex intelligence for conceptually understanding and inferring like human). This work also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00553041, Enhancement of Rational and Emotional Intelligence of Large Language Models for Implementing Dependable Conversational Agents).

References

- Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94*, page 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Leo Breiman. 2001. Random Forests. *Machine learning*, 45:5–32.
- 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 Chevalier, Jiayi Geng, Alexander Wettig, Howard Chen, Sebastian Mizera, Toni Annala, Max Jameson Aragon, Arturo Rodríguez Fanlo, Simon Frieder, Simon Machado, Akshara Prabhakar, Ellie Thieu, Jia-chen T. Wang, Zirui Wang, Xindi Wu, Mengzhou Xia, Wenhan Xia, Jiatong Yu, Jun-Jie Zhu, and 3 others. 2024. Language Models as Science Tutors. In *Proceedings of the 41st International Conference on Machine Learning*, ICML’24. JMLR.org.
- D. R. Cox. 1958. [The Regression Analysis of Binary Sequences](#). *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242.
- Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. 2025. Security and Privacy Challenges of Large Language Models: A Survey. *ACM Computing Surveys*, 57(6):1–39.
- 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, Los Alamitos, CA, USA. IEEE Computer Society.
- 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)*, pages 13806–13834, Bangkok, Thailand. Association for Computational Linguistics.
- Hongjin Kim, Jeonghyun Kang, and Harksoo Kim. 2025. [Can Large Language Models Differentiate Harmful from Argumentative Essays? Steps Toward Ethical Essay Scoring](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8121–8147, Abu Dhabi, UAE. Association for Computational Linguistics.
- Hongjin Kim, Jai-Eun Kim, and Harksoo Kim. 2024. [Exploring Nested Named Entity Recognition with Large Language Models: Methods, Challenges, and Insights](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8653–8670, Miami, Florida, USA. Association for Computational Linguistics.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
2024. [Ask-Before-Detection: Identifying and Mitigating Conformity Bias in LLM-Powered Error Detector for Math Word Problem Solutions](#). *arXiv preprint arXiv:2412.16838*.
- 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.
- 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.
- OpenAI. 2025. [Introducing GPT-4.1 in the API](#). <https://openai.com/index/gpt-4-1/>. Accessed: 2025-05-20.
- Rose E. Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024a. [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*. Association for Computational Linguistics.
- Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024b. [Large Language Models for Education: A Survey and Outlook](#). *arXiv preprint arXiv:2403.18105*.
- Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024c. [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.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. [Chain-of-Thought Prompting Elicits Reasoning in Large Language Models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Hang Li, Tianlong Xu, Kaiqi Yang, Yucheng Chu, Yanling Chen, Yichi Song, Qingsong Wen, and Hui Liu.

A Chain-of-Thought-based Pedagogical Evaluation with Reasoning layers Prompts

This section presents the detailed prompt used in the Chain-of-Thought-based Pedagogical Evaluation with Reasoning Layers methodology described in Section 2.1. The prompt serves as a core component of the proposed step-by-step structure, which emulates how human tutors interpret students' reasoning and determine appropriate feedback. It guides the model to first reconstruct the student's reasoning, then generate a correct solution path, and finally evaluate the pedagogical appropriateness of the tutor's response in light of the student's thinking process.

Problem Extraction

Identity

You are an expert in analyzing conversations and extracting specific information precisely from textual inputs.

Your task is to read through a dialogue transcript carefully and extract a math problem as-is, without modifying any part of it. The conversation always contains exactly one math problem.

Instructions

- * Read the conversation carefully and identify the one math problem embedded within.
- * Copy the entire text of that math problem verbatim, exactly as it appears in the dialogue.
- * Do not add any explanation, paraphrasing, or interpretation.
- * Output only the extracted math problem and nothing else.

Student Reasoning Process Reconstruction

Identity

You are an expert in analyzing educational conversations to reconstruct the reasoning processes behind students' mathematical answers.

Your task is to read a conversation between a tutor and a student, along with the math problem discussed. Then, from the student's point of view, explain the reasoning process the student might have used to arrive at their final answer.

Your goal is to reconstruct the student's reasoning path — whether correct or incorrect — as faithfully and coherently as possible based on the conversation and the problem given.

Instructions

1. Do not modify, correct, or reinterpret the student's final answer — even if the answer is incorrect.
2. Base your reasoning entirely on what was stated in the conversation.
3. If no reasoning was explicitly given by the student, infer a likely and plausible thought process they could have followed to reach their answer.
4. Ensure that your explanation is logically consistent with the content of the conversation. Avoid introducing contradictions.
5. Write your reasoning as a clear, step-by-step explanation, emulating how the student may have thought through the problem.

Correct Reasoning Process Generation

Identity

You are a logical reasoning assistant specialized in mathematical thinking and student misconception analysis.

You are given:

1. A transcript of a conversation between a student and a tutor
2. The math problem discussed in the conversation
3. The student's reasoning process, which has been reconstructed based on the conversation, not written directly by the student.
 - Parts explicitly mentioned in the dialogue can be trusted as the student's actual reasoning.
 - Parts not mentioned directly have been inferred based on the dialogue and should be treated as plausible, but not definitive.

Your task is to carefully analyze the student's reasoning process and perform the following instructions:

Instructions

Step 1: Identify Reasoning Errors Review the student's reasoning. Clearly point out any logical, mathematical, or conceptual errors. If there are no errors, state that explicitly.

Step 2: Reconstruct the Correct Reasoning (Based on Student's Thought Process) If the student made partial progress or had a valid approach but made an error along the way, retain and respect their original reasoning path. Correct the specific mistakes and continue the reasoning from where they deviated. If the student's reasoning is fundamentally flawed from the beginning or completely irrelevant to the problem, it is acceptable to construct a new, correct reasoning path.

Step 3: Solve the Problem Using the corrected reasoning (rooted in the student's approach if applicable), solve the math problem and provide the correct final answer.

Step 4: Output Format Provide your response in the following structure:

- Student Reasoning Error(s): [List and explain]
- Corrected Reasoning (Respecting Student's Logic): [Step-by-step, rooted in their original path]
- Final Answer: [Answer]

Tutor Response Evaluation

Identity

You are a senior math tutor and tutor coach with expertise in evaluating instructional quality. You will receive the following inputs:

1. A dialogue between a student and a novice tutor.
2. The math problem discussed in the dialogue.
3. A senior tutor's analysis of the student's likely reasoning and a revised correct solution.
4. The final utterance made by the novice tutor.

Instructions

Evaluation Criteria

- Evaluate the novice tutor's final utterance, using the following four criteria:
- broader dialogue context, including the student's previous responses and the progression of the conversation.
- In other words, evaluate how well the tutor's final utterance functions as a response within the instructional flow and in light of the student's reasoning process.

1. Mistake Identification

- Did the novice tutor demonstrate awareness of a mistake in the student's reasoning?
 - Yes: The tutor reasonably indicates awareness of the student's mistake or explicitly suggests the possibility of an error, even if somewhat general.
 - To some extent: The tutor vaguely hints at a mistake, but the suggestion is overly ambiguous or uncertain.
 - No: The tutor does not identify or suggest any mistake in the student's reasoning.

2. Mistake Location

- Did the novice tutor pinpoint where the mistake occurred in the student's process?
 - Yes: The tutor appropriately identifies or indicates the step or area where the student's mistake occurred. Exact pinpointing is not required, as long as the general location or nature of the error is clear.
 - To some extent: The tutor provides only a vague or unclear indication of the error's location, potentially leading to student confusion.
 - No: The tutor does not specify or reference where the student's error occurred.

3. Pedagogical Guidance

- Did the novice tutor provide helpful explanations, hints, or examples to support student learning?
 - Yes: The tutor provides explanations, hints, or examples that meaningfully support student understanding. Slight inaccuracies or imperfections are acceptable as long as the guidance is helpful from the student's perspective.
 - To some extent: The tutor offers guidance, but it contains significant ambiguities, inaccuracies, or potential misconceptions.
 - No: The tutor provides no useful explanation or hints, or the provided guidance is clearly incorrect or misleading.

4. Actionability

- Can the student clearly understand what to do next based on the tutor's response?
 - Yes: The tutor reasonably suggests a clear next step or strategy that the student can readily understand and follow. Explicit instructions are not required as long as the suggested action is practical and clear enough.
 - To some extent: The tutor suggests a next step, but the recommendation is unclear, confusing, or insufficiently specific.
 - No: The tutor does not suggest any actionable next step or strategy for the student.

Output Format (MANDATORY)

- Respond in exactly the following format. Do not change the structure, headings, or indentation.

"Mistake Identification: [Yes / To some extent / No]

Explanation: [...]

Mistake Location: [Yes / To some extent / No]

Explanation: [...]

Pedagogical Guidance: [Yes / To some extent / No]

Explanation: [...]

Actionability: [Yes / To some extent / No]

Explanation: [...]"

You must follow this format strictly. Any deviation will be considered incorrect.

Now, evaluate the novice tutor's final utterance.

Basic Prompt

Identity

You are a senior math tutor and tutor coach with expertise in evaluating instructional quality. You will receive the following inputs:

1. A dialogue between a student and a novice tutor.
2. The final utterance made by the novice tutor.

Instructions:

Evaluation Criteria

- Evaluate the novice tutor's final utterance, using the following four criteria.

1. Mistake Identification

- Detect whether tutors' responses recognize mistakes in students' responses. The following categories are included:
 - Yes: The mistake is clearly identified/recognized in the tutor's response.
 - To some extent: The tutor's response suggests that there may be a mistake, but it sounds as if the tutor is not certain.
 - No: The tutor does not recognize the mistake (e.g., they proceed to simply provide the answer to the asked question).

2. Mistake Location

- Assess whether tutors' responses accurately point to a genuine mistake and its location in the students' responses. The following categories are included:
 - Yes: The tutor clearly points to the exact location of a genuine mistake in the student's solution.
 - To some extent: The response demonstrates some awareness of the exact mistake, but is vague, unclear, or easy to misunderstand.
 - No: The response does not provide any details related to the mistake.

3. Pedagogical Guidance

- Evaluate whether tutors' responses offer correct and relevant guidance, such as an explanation, elaboration, hint, examples, and so on. The following categories are included:
 - Yes: The tutor provides guidance that is correct and relevant to the student's mistake.
 - To some extent: Guidance is provided but it is fully or partially incorrect, incomplete, or somewhat misleading.
 - No: The tutor's response does not include any guidance, or the guidance provided is irrelevant to the question or factually incorrect.

4. Actionability

- Assess whether tutors' feedback is actionable, i.e., it makes it clear what the student should do next. The following categories are included:
 - Yes: The response provides clear suggestions on what the student should do next.
 - To some extent: The response indicates that something needs to be done, but it is not clear what exactly that is.
 - No: The response does not suggest any action on the part of the student (e.g., it simply reveals the final answer).

Output Format (MANDATORY)

- Respond in exactly the following format. Do not change the structure, headings, or indentation.

Mistake Identification: [Yes / To some extent / No]

Explanation: [...]

Mistake Location: [Yes / To some extent / No]

Explanation: [...]

Pedagogical Guidance: [Yes / To some extent / No]

Explanation: [...]

Actionability: [Yes / To some extent / No]

Explanation: [...]

You must follow this format strictly. Any deviation will be considered incorrect.

Now, evaluate the novice tutor's final utterance.

B Rubric Based Evaluation Method prompt

To evaluate the Providing Guidance dimension, we designed a structured prompt that guides the language model to simulate expert judgment across six sub-criteria. The prompt first provides the full dialogue context, the student's final utterance, and the tutor's response. It also includes prediction results from other evaluation dimensions (Mistake Identification and Mistake Location), as well as statistical correlations observed between them. The model is instructed to assess the tutor's response based on six specific questions, each targeting a pedagogically meaningful aspect of guidance. For each item, the model outputs a brief rationale and assigns probabilities to three labels: Yes, To some extent, and No. The final output consists of both the explanation and a normalized probability distribution that sums to 1.0. The sixth question is designed to produce an overall judgment by incorporating both model predictions and prior statistical informations, providing a holistic measure of guidance quality.

Prompt for Rubric-Based Multidimensional Evaluation

You are an expert evaluator analyzing a tutor's response in a learning dialogue.

Below is a conversation between a student and a tutor.

[Conversation]

<Full conversation history, if any>

[STUDENT_UTTERANCE]

<Student's latest input>

[TUTOR_RESPONSE]

<Tutor's response to be evaluated>

[Prediction Results]

- Mistake Identification: <Predicted label>
- Mistake Location: <Predicted label>

Note: Based on statistical analysis of past data, the following association rules are observed:

- If Providing Guidance is "Yes", then Mistake Identification is also "Yes" with confidence 0.950.
- If Mistake Location is "Yes", then Mistake Identification is "Yes" with confidence 0.989.
- If Mistake Identification is "Yes", then Mistake Location is "Yes" with confidence 0.790.

Use the following definitions when choosing a label:

- Yes: The tutor's response fully satisfies the criterion. It is accurate, relevant, and helpful.
- To some extent: The response attempts to satisfy the criterion but is partially incomplete, inaccurate, vague, or not directly useful.
- No: The response does not satisfy the criterion at all, or it is misleading, unrelated, or entirely incorrect.

Please answer the following six questions. For each question, first provide a brief explanation for your judgment. Then, give the probability (in float format) that the response is: Yes, To some extent, or No. Ensure all three values sum to exactly 1.0.

Output format:

- Q1: <brief explanation>
- Yes: <float>
 - To some extent: <float>

- No: <float>

Questions:

- Q1. Did the tutor attempt to provide any explanation, hint, or example?
- Q2. Was the guidance factually correct and appropriate given the student's error?
- Q3. Did the tutor's response directly address the student's specific mistake?
- Q4. Did the guidance help the student figure out what to do next, without directly giving the final answer?
- Q5. Was the tutor's response clear and unlikely to confuse the student?
- Q6. Based on the tutor's response, the model's predictions, and the above statistical information, how likely is it that the tutor attempted to provide meaningful guidance?

C Analysis of Prediction Results on the Development Set

This section presents the classification results on the development set for each of our proposed systems. Although the overall metrics provide a broad overview of performance, they do not sufficiently capture the models’ ability to discriminate fine-grained categories—particularly ambiguous ones such as To some extent. Therefore, we provide a detailed analysis of each system’s predictions according to the evaluation track.

C.1 Chain-of-Thought-based Evaluation System

As shown in Section 3.3, our proposed Chain-of-Thought-based evaluation system demonstrated effectiveness in generating evaluations that are both consistent and pedagogically valid. However, as previously noted, the model still exhibits limitations in accurately distinguishing between semantically adjacent evaluation categories. To further investigate this issue, we conducted an analysis of how such difficulties manifest in actual prediction outcomes.

Actual / Predict	Yes	To some extent	No	Total
Yes	1,657	237	38	1,932
To some extent	63	68	43	174
No	60	56	254	370

Table 6: Confusion matrix of the CoT-based Evaluation System for the Mistake Identification track.

Table 6 presents the prediction results for the Mistake Identification track in the form of a confusion matrix. In this track, the model accurately classified the majority of "Yes" instances (1,657 out of 1,932), but struggled to distinguish the "To some extent" category. Specifically, only 68 out of 174 instances were correctly identified, while the remaining were misclassified as "Yes" (63 instances) or "No" (43 instances), indicating persistent challenges in delineating fine-grained judgment boundaries.

Actual / Predict	Yes	To some extent	No	Total
Yes	727	547	36	1,310
To some extent	88	245	36	369
No	253	183	361	797

Table 7: Confusion matrix of the CoT-based Evaluation System for the Actionability track.

A similar pattern is observed in the Actionabil-

ity track, as shown in Table 7. While the model achieved relatively high true positive counts for the "Yes" (727 instances) and "No" (361 instances) categories, “To some extent” cases were frequently misclassified—most notably, among the actual “No” instances, 547 were predicted as "Yes" and 183 as "To some extent".

These results indicate that while the proposed Chain-of-Thought-based evaluation system is effective in producing consistent judgments based on explicit criteria, it still faces limitations in clearly distinguishing semantically adjacent categories. In particular, the frequent misclassification of ambiguous labels such as To some extent highlights the difficulty of inducing fine-grained reasoning solely through prompts without task-specific training. This observation suggests the potential need for improved prompt engineering or subsequent fine-tuning to enhance the model’s discriminative precision.

C.2 Multi-Perspective Reflective Evaluation System

This section presents an analysis of the Multi-Perspective Reflective Evaluation System’s performance on the Mistake Location track. The goal is to understand how effectively the system distinguishes between clearly defined and semantically adjacent categories within its reflective reasoning framework.

Actual \ Predicted	Yes	To some extent	No	Total
Yes	1,219	140	184	1,543
To some extent	99	33	88	220
No	117	79	517	713

Table 8: Confusion matrix of the Multi-Perspective Reflective Evaluation System for the Mistake Location track.

The analysis of Mistake Location is presented in Table 8. The results show that although the system accurately identifies many instances of "Yes" (1,219 correct predictions), it struggles to distinguish "To some extent" from adjacent categories. Specifically, only 33 out of 220 "To some extent" cases were correctly classified, while the majority were misclassified as either "Yes" (99 instances) or "No" (88 instances). This analysis supports our observations in Sections 3.3 and 3.4 that prompt-based reasoning approaches still face challenges in making fine-grained categorical distinctions.

C.3 Rubric Based Evaluation System

To assess the performance of the Rubric Based Evaluation System in identifying pedagogically meaningful distinctions, we examine its predictions on the Providing Guidance track. This allows us to evaluate the effectiveness of rubric-derived features in capturing subtle differences between response categories.

Actual / Predict	Yes	To some extent	No	Total
Yes	1,034	266	107	1,407
To some extent	250	179	74	503
No	178	83	305	566

Table 9: Confusion matrix of the Rubric Based Evaluation System for the Providing Guidance track.

Table 9 presents the prediction results of our proposed Rubric Based Evaluation System. The system consists of a Random Forest classifier trained using the Q_{rubric} items and the predicted values of $Q_{overall}$ as input features. For the "Yes" class, 1,034 out of 1,407 instances were correctly classified. For the "No" class, 305 out of 566 instances were correctly classified. In contrast, for the "To some extent" class, only 179 out of 503 instances were correctly classified, with 250 instances misclassified as "Yes" and 74 as "No." These results indicate that the classifier struggled to clearly distinguish the "To some extent" class from "Yes" and "No." This suggests that, even when combining information from Q_{rubric} and $Q_{overall}$, additional feature engineering or refinement may be necessary to more precisely delineate the boundaries among the three classes.

Henry at BEA 2025 Shared Task: Improving AI Tutor’s Guidance Evaluation Through Context-Aware Distillation

Pagnarith Pit

University of Melbourne

ppit@student.unimelb.edu.au

Abstract

Effective AI tutoring hinges on guiding learners with the right balance of support. In this work, we introduce **CODE** (**C**ontextually-aware **D**istilled **E**valuator), a framework that harnesses advanced large language models (i.e., GPT-4o and Claude-2.7) to generate synthetic, context-aware justifications for human-annotated tutor responses in the BEA 2025 Shared Task. By distilling these justifications into a smaller open-source model (i.e., Phi-3.5-mini-instruct) via initial supervised finetuning and then Group Relative Policy Optimization, we achieve substantial gains in label prediction over direct prompting of proprietary LLMs. Our experiments show that **CODE** reliably identifies strong positive and negative guidance, but like prior work, struggles to distinguish nuanced “middle-ground” cases where partial hints blur with vagueness. We argue that overcoming this limitation will require the development of explicit, feature-based evaluation metrics that systematically map latent pedagogical qualities to model outputs, enabling more transparent and robust assessment of AI-driven tutoring.

1 Introduction

Large language models (LLMs) have opened a promising frontier for education, enabling conversational agents that deliver personalized and adaptive guidance calibrated to a learner’s current knowledge state and pace (Tack et al., 2023). Indeed, the main goal of dialectic teaching is to provoke exploration through carefully timed questions, hints, or explanations (Clark and Egan, 2015). If the guidance provided is too little, it frustrates students, while too much erodes learning opportunities and fosters over-reliance (Le, 2019). Although striking this balance is central to effective tutoring, the field still lacks precise operational definitions and automatic metrics for “optimal guidance”, making systematic eval-

uation, and therefore progress, very challenging (Kochmar et al., 2025).

In light of this missing definition, existing assessments rely heavily on individual annotation by human experts (Maurya et al., 2025). However, crafting high-quality, question-specific explanations at the scale needed to train or benchmark modern transformer models is prohibitively expensive. To address this bottleneck, we explore *reasoning distillation*: using stronger LLMs to generate reasoning about a tutor’s utterance as to why it matches the gold label. Our study investigates (i) whether synthetically contexts capture meaningful signals of pedagogical quality, and (ii) how well these signals transfer when smaller, student models are trained on them.

As such, our contributions from team Henry are as follows:

- We propose **C**ontextually-aware **D**istilled **E**valuator (**CODE**) framework, a multi-step finetuning process that distills reasoning from larger LLMs to train smaller open-sourced models to better detect what “good guidance” is. Our method consistently outperforms state-of-the-art (SOTA) proprietary models and aligns reasonably well with expert human judgements.
- We release an enriched dataset with synthetically generated reasoning based on their gold labels for each of the tutor’s last utterance across the entire human-annotated set from (Maurya et al., 2025).

2 Related Work

2.1 AI Tutor’s Guidance Evaluation

This feature of AI tutor currently lacks a unified definition, but there has been efforts in this area to explore it through various perspectives. Tack

and Piech (2022) in their work evaluates performances of AI tutor based on how much they “help the student” using human participants and expert annotators. While they don’t provide a formal definition, their approach to evaluation is closely resembled by Daheim et al. (2024)’s “actionability” where the AI tutor’s utterance provides sufficient information for the student to progress the conversation and move closer to the correct answer.

In another work by Wang et al. (2024), this feature is referred to as “usefulness”, the degree to which the responses are productive at advancing the student’s understanding and helping them learn from their errors, also evaluated through human judgments. These concepts are also reflected in the work of Al-Hossami et al. (2023), where they defined “indirectness”, where an effective tutor asks questions that induce critical thinking and not reveal the answer.

2.2 Learning via Distillation

Knowledge distillation transfers the knowledge embedded in large, high-capacity “teacher” models into smaller, more efficient “student” models by having the student match the teacher’s softened probability distributions, known as “soft targets”, rather than relying solely on hard labels. First introduced by Hinton et al. (2015), this technique has enabled compact language models to approach the performance of much larger LLMs while using reduced architectures and training data (Hsieh et al., 2023).

More recently, distillation has been extended to complex reasoning tasks, spawning the field of reasoning distillation. For example, Li et al. (2025) present Fault-Aware Distillation via Peer-Review (FAIR), in which multiple teacher models critique each other’s reasoning chains to improve fidelity. Likewise, Dai et al. (2024) propose training student models on key reasoning steps extracted from dual chain-of-thought explanations. These innovations not only enhance model interpretability but also substantially boost conceptual understanding in educational applications.

3 Methods

3.1 Synthetic Context Generation

Data Preprocessing We focus exclusively on Task 3 of the BEA Shared Task (Kochmar et al., 2025), and so we process the original dataset from Maurya et al. (2025) accordingly. From

the provided validation set, we construct a filtered dataset:

$$\mathcal{D} = \{(C_i, R_i, L_i)\}_{i=1}^N,$$

such that for each sample i :

- C_i : conversation history of each original element,
- R_i : each tutor’s response,
- $L_i \in \{Yes, To\ Some\ Extent, No\}$: the gold label provided by “Providing Guidance”

In total, we have $N = 3,589$.

Generating Reasoning with Labels To enrich each response label with contextual justification, we leverage two state-of-the-art models, namely GPT-4o (OpenAI et al., 2024) and Claude-2.7 Sonnet (Anthropic, 2024). For each model, we process batches of 10 examples from our original dataset \mathcal{D}^1 alongside the system prompt in Appendix A. Each model then generates a justification J_i for sample i , drawing on the provided label L_i , the conversation history C_i , and the latest tutor response R_i . This yields an expanded dataset

$$\mathcal{D}' = \{(C_i, R_i, L_i, J_i)\}_{i=1}^N,$$

where J_i is the synthetic justification associated with the i th response.

Selection of Justifications To ensure the quality and utility of the synthetically generated justifications, we conduct a manual selection process to identify the most suitable responses produced by the two models. The selection criteria are as follows:

- **Non-repetition:** Justifications that are repeated within the same batch are excluded to prevent redundant signals, which could lead to overfitting during downstream model training.
- **Linguistic diversity and specificity:** Selected justifications exhibit varied and distinctive vocabulary, reflecting the natural diversity found in human tutor responses. This diversity will then enhance the generalizability of models trained on the data.

¹Batch size selected after varying from 1 to 50. We find that beyond 10 samples, both models tend to hallucinate or become overly generic and provide low-quality responses.

- **Adequate length and contextual richness:** Justifications are required to provide sufficient explanatory detail to offer meaningful context for the corresponding labeled response.

Extracting Critical Tokens For each synthetic justification J_i , we perform the following preprocessing steps:

1. Convert to lowercase and strip leading/trailing whitespace.
2. Remove all stopwords.
3. Tokenize the resulting string.
4. Apply stemming and lemmatization to each token.

Let

$$\mathcal{T}_i = \{t_{i1}, t_{i2}, \dots, t_{iK_i}\}$$

be the set of remaining tokens for sample i . We refer to \mathcal{T}_i as the *context-critical* token set, and we use these tokens as our reward signals. With this, we now proceed to training our student model.

3.2 Expert Alignment Through Reinforcement Learning

To best align the student model’s outputs with those of advanced LLMs, and potentially a human tutor expert, we introduce the **CO**ntextually-aware **D**istilled **E**valuator (**CODE**) framework. In **CODE**, reasoning is distilled through a multi-step transfer process, with tailored reward signals that guide the model to generate contextually relevant tokens for downstream classification.

3.2.1 Initial Supervised Learning

We begin by performing supervised fine-tuning (SFT) to teach the student model to generate J_i given the conversation history C_i and last response R_i . This initial stage aims to instill the desired format, tone, and length characteristic of expert-generated justifications. At this point, emphasis is placed not on the semantic quality or reasoning depth of the model’s outputs, but rather on aligning the stylistic aspects of the responses to facilitate more efficient convergence during subsequent training phases. We have:

$$J_i = (j_{i,1}, j_{i,2}, \dots, j_{i,T_i}),$$

where each justification is represented as a token sequence. Under a standard cross-entropy objective, the per-sample loss is

$$\ell_i(\theta) = -\frac{1}{T_i} \sum_{t=1}^{T_i} \log p_\theta(j_{i,t} | C_i, R_i, j_{i,<t}),$$

where $p_\theta(\cdot)$ is the student model’s predicted probability and $j_{i,<t} = (j_{i,1}, \dots, j_{i,t-1})$. Averaging over all N samples gives the final SFT loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = \frac{1}{N} \sum_{i=1}^N \ell_i(\theta)$$

The system prompt used as part of this instruction tuning is provided in Appendix C.

3.2.2 Applying GRPO with Semantic Rewards

After supervised fine-tuning, we now refine the student model’s output quality via an online reinforcement-learning algorithm known as *Group Relative Policy Optimization* (GRPO) (Shao et al., 2024). We denote the student’s policy by

$$\pi_\theta(J | C_i, R_i),$$

parameterized by θ , which we adapt efficiently using low-rank adaptation (LoRA) (Hu et al., 2021) updates to the transformer weights.

Group sampling and baseline For each training example (C_i, R_i) , the model samples a *group* of M candidate justifications:

$$\{J_i^1, J_i^2, \dots, J_i^M\} \sim \prod_{j=1}^M \pi_\theta(\cdot | C_i, R_i).$$

Each candidate J_i^j is scored by a programmable reward function r_i^j . We then compute the group baseline as the mean reward:

$$b_i = \frac{1}{M} \sum_{j=1}^M r_i^j.$$

Reward design The total reward r_i^j is a weighted sum of three components:

$$r_i^j = w_{\text{tok}} r_i^{\text{tok}}(J_i^j) + w_{\text{sent}} r_i^{\text{sent}}(J_i^j) + w_{\text{ppl}} r_i^{\text{ppl}}(J_i^j),$$

where:

$$r_i^{\text{tok}}(J) = \sum_{t \in \mathcal{T}_i} \mathbf{1}\{t \text{ appears in } J\},$$

$$r_i^{\text{sent}}(J) = \begin{cases} 1 & \text{if the sentiment of } J \text{ matches the gold label,} \\ 0 & \text{otherwise,} \end{cases}$$

$$r_i^{\text{ppl}}(J) = -\frac{1}{|J|} \sum_{t=1}^{|J|} \log p_{\theta_0}(j_t | C_i, R_i, j_{<t}).$$

Here \mathcal{T}_i is the context-critical token set for example i . In this reward scheme, we do not punish the student model arbitrarily for not generating trivial tokens. Likewise, it is only rewarded if it can generate the critical tokens that would be informative for the response’s label based on the context provided.

Next, the sentiment score is given by a finetuned transformer based on DistilBERT (Sanh et al., 2020) (i.e., DistilBERT-based SST-2 classifier). While sentiment alone is not a reliable indicator of guidance quality (Wang et al., 2024), it provides a concrete and readily interpretable signal that can guide model generation. The inclusion of this sentiment-based reward facilitates faster convergence of the student model by offering an easier-to-learn proxy objective compared to directly optimizing for alignment with complex gold labels. Crucially, sentiment is not intended as a hard classification signal but rather a soft reward, encouraging the generation of justifications whose affective tone is consistent with the associated label. This approach helps steer the model’s learning trajectory in a meaningful direction in light of GRPO’s multiple response generation.

Finally, p_{θ_0} denotes the frozen base model (i.e., untrained student model) used to compute perplexity. Importantly, the perplexity score is added such that the trained model does not exploit the other reward signals by randomly inserting tokens as part of their outputs. This ensures that the final responses produced by the trained student model is still cohesive and human understandable.

Before performing the policy gradient update, these scores are then normalised to zero mean and unit variance to prevent their magnitude from dominating other scores, with the weights ($w_{\text{tok}}, w_{\text{sent}}, w_{\text{ppl}}$) balancing these signals.²

3.3 Final Classification

To produce the final label predictions, we append a trainable classification head atop the trained student model. The primary objective of this step is feature selection. That is, the student model has been previously trained to generate justifications containing critical tokens, and as such, this classification head aims to capture and interpret these contextual cues, mapping them effectively to the

²We experimented with several weighting schemes but observed only minor, non-meaningful variations. As such, for our final implementation we adopted uniform weights, assigning a value of 1 to each.

target label space. In this final training stage, the model is trained to associate its own generated output with the corresponding gold label.

We first map each gold label:

$$L_i \in \{\text{Yes, To Some Extent, No}\}$$

to a categorical target $y_i \in \{1, 2, 3\}$. Let θ^* denote the student model parameters after merging the LoRA adapters. Here, we freeze all θ^* and add a dense feedforward layer with parameters $\phi = (W, b)$, where

$$W \in R^{3 \times d}, \quad b \in R^3.$$

However, instead of training it on one forward pass on each example (C_i, R_i) , we first generate the student model’s full response by

$$\hat{J}_i = \arg \max_J \pi_{\theta^*}(J | C_i, R_i).$$

We encode this output using the student model’s tokenizer, and train ϕ using standard cross-entropy loss on the last token hidden state:

$$\mathcal{L}_{\text{CE}}(\phi) = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^3 \mathbf{1}[y_i = c] \log \hat{p}_{i,c}.$$

where $\hat{p}_{i,c}$ is the softmax of the classification head’s predicted labels.

This pooling design choice to use the last token hidden state as the representation is particularly motivated by the architecture of decoder-only transformers, which lack a dedicated classification token such as [CLS] found in encoder-based models (Fu et al., 2023). The last token in our generated outputs (i.e., each justification) functions as a natural summary or conclusion to the token sequence, providing a meaningful contextual embedding that reflects the entire output. This approach balances computational efficiency, avoiding the increased complexity of attention-based pooling, and mitigates the potential noise or dilution of critical token signals that may arise with mean pooling strategies (Suganthan et al., 2025).

External Benchmarking In addition to comparing against gold labels and the CodaBench leaderboard submission, we also evaluate whether our method could outperform direct prompt-tuning of the proprietary models. For each example, we prompt GPT-4o and Claude-2.7 Sonnet to predict the guidance label without revealing the true label and recorded their accuracy on the validation set. The exact prompts used for this experiment are provided in the Appendix B.

Model	Validation set				CodaBench set			
	Ex. F1	Ex. Acc	Len. F1	Len. Acc	Ex. F1	Ex. Acc	Len. F1	Len. Acc
GPT-4o	56%	69%	75%	83%	49%	58%	70%	75%
Claude-2.7 Sonnet	61%	70%	73%	81%	—	—	—	—
CoDE	64%	74%	83%	89%	53%	63%	72%	78%

Table 1: Evaluation of all models across both validation set and the CodaBench test set. Due to the limited nature of submission on CodaBench set, results from Claude-2.7 Sonnet were not submitted in the competition. All result reported has been rounded to the nearest percent. The final reported score on the official leaderboard for CoDE is slightly higher than the reported value in this table, but because the baseline score of GPT-4o is not reported there, we report the values of CoDE from the unofficial table for consistency.

3.4 Experimental Setup

We used the Unsloth-provided “Phi-3.5-mini-instruct” (Daniel Han and team, 2023; Abdin et al., 2024) as our student model. The original validation set was further split 80/20 into training and test subsets. All data preprocessing ran on an NVIDIA L40 GPGPU, with model training and evaluation performed on an NVIDIA A100 GPU. In total, preprocessing, training, and evaluation consumed over 70 GPU-hours. Complete details on training hyperparameters, such as GRPO and LoRA parameters, are detailed in the Appendix.

4 Results

As shown in Table 1, **CODE** consistently outperforms state-of-the-art baselines, achieving the tenth position in the final CodaBench ranking. We attribute this improvement both to the quality of the synthetic data and to the student model’s ability to capture hidden features from the extended context. Our results suggest that existing SOTA models possess an implicit notion of “good guidance”, and their generated outputs can be effectively transferred to smaller models. This observation corroborates prior work demonstrating that large language models can serve as an effective tutors, offering substantial instructional value, albeit not at expert-level proficiency (Wollny et al., 2021).

Notably, on both the validation set and, to a lesser extent, the CodaBench benchmark, **CODE** exhibits a larger gain when evaluated with lenient F1 compared to exact F1, with improvements under strict scoring criteria remain modest. This pattern indicates that fine-tuning renders **CODE** less sensitive to the ambiguous label that is “To some extent”. The strong labels, “Yes” or “No”, are much easier to deduce, with clearer human definition, but this “middle-ground” is much more nu-

anced, and since finetuning is known to reduce LLMs’ general reasoning (Luo et al., 2025), this drop may be inevitable. When pedagogical value differs only slightly, we see that even among human experts, these are difficult to discern (Macina et al., 2023).

5 Conclusion

In this paper, we have investigated the potential of modern large language models to both generate and train on synthetic data that emulate expert human reasoning in educational guidance through our **CODE** framework. Across both our validation and CodaBench test sets, our approach consistently outperforms SOTA baselines and aligns reasonably well with human judgments. However, our findings also underscore the persistent challenge of the absence of a formal, operational definition of this pedagogical quality. In particular, nuances embodied by the “middle-ground” label appear too subtle or demand too much data for current LLMs to learn reliably.

As future direction, we advocate for the continued development of explicit metrics that systematically map these latent pedagogical features to models’ outputs. By grounding fine-tuning in a well-defined, feature-based evaluation framework, we can move beyond black-box learning of hidden signals and instead foster more robust, transparent, and interpretable AI tutoring systems.

Limitations

5.1 Models Faithfulness and Prompt Sensitivity

The dataset created is not guaranteed to match reasoning provided by expert tutors. While we have conducted manual inspections on samples in the synthetic data to ensure some level of consis-

tency between reasoning and the label provided, this cannot be assured.

Furthermore, the SFT prompt used for training may not be optimal. This was chosen only after a few iterations of prompt tuning on a sample of the synthetic data.

5.2 Distillation Cost

Both models used are not open source nor free. Generating these takes extensive time on paid models, limiting the number of reasoning responses to one per sample. Ideally, we would like to expand this dataset further by generating multiple responses under various prompts to better simulate the diversity in thinking among real human tutors.

Ethical Consideration

Our research adheres strictly to ethical standards, using publicly available datasets as well as following distillation restriction carefully. We uphold the principles of fairness, accountability, and academic integrity throughout the research process.

References

- Marah Abidin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benham, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. [Phi-3 technical report: A highly capable language model locally on your phone](#). *Preprint*, arXiv:2404.14219.
- Erfan Al-Hossami, Razvan Bunescu, Ryan Teehan, Laurel Powell, Khyati Mahajan, and Mohsen Dordchi. 2023. [Socratic questioning of novice debuggers: A benchmark dataset and preliminary evaluations](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 709–726, Toronto, Canada. Association for Computational Linguistics.
- Anthropic. 2024. [Claude 2.7 sonnet](#). [Large language model].
- Gavin Clark and Sarah Egan. 2015. [The socratic method in cognitive behavioural therapy: A narrative review](#). *Cognitive Therapy and Research*, pages 1–17.
- 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.
- Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. 2024. [Beyond imitation: Learning key reasoning steps from dual chain-of-thoughts in reasoning distillation](#). *Preprint*, arXiv:2405.19737.
- Michael Han Daniel Han and Unsloth team. 2023. [Unsloth](#).
- Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, Shengding Hu, Zhiyuan Liu, and Nigel Collier. 2023. [Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder](#). *Preprint*, arXiv:2304.04052.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#). *Preprint*, arXiv:1503.02531.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander 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](#). *Preprint*, arXiv:2305.02301.
- 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.
- Ekaterina Kochmar, Kaushal Kumar 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)*.
- Nguyen-Thanh Le. 2019. [How do technology-enhanced learning tools support critical thinking?](#) *Frontiers in Education*, 4.
- Zhuochun Li, Yuelu Ji, Rui Meng, and Daqing He. 2025. [Learning from committee: Reasoning distillation from a mixture of teachers with peer-review](#). *Preprint*, arXiv:2410.03663.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. [An empirical study of catastrophic forgetting in large language models during continual fine-tuning](#). *Preprint*, arXiv:2308.08747.
- 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.

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.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. [Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter](#). *Preprint*, arXiv:1910.01108.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.

Paul Suganthan, Fedor Moiseev, Le Yan, Junru Wu, Jianmo Ni, Jay Han, Imed Zitouni, Enrique Alfonseca, Xuanhui Wang, and Zhe Dong. 2025. [Adapting decoder-based language models for diverse encoder downstream tasks](#). *Preprint*, arXiv:2503.02656.

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.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. [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.

Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik

Drachslar. 2021. [Are we there yet? - a systematic literature review on chatbots in education](#). *Frontiers in Artificial Intelligence*, 4:654924.

A Synthetic Data Generation Prompt

For both GPT-4o and Claude, we used the following prompt to create our dataset:

Data Creation Prompt

You are an expert evaluator of Socratic-style tutoring dialogs in programming education. Your task is to justify the human-supplied quality label for the tutor’s last response You will receive: `conversation_history` (full transcript up to, but NOT including, the tutor’s latest reply, and the speaker turns are prefixed with “Student:” or “Tutor:”, `last_response` (the tutor’s latest reply, to be evaluated), and `label` (one of Yes, To some extent, No indicating whether the reply provides adequate, partial, or no helpful guidance to the student. These labels correspond to:

- Yes: The reply gives sufficient, specific, actionable guidance or hints that directly help the student correct their error or deepen understanding.
- To some extent: Contains some guidance, but it is vague, incomplete, or only tangentially helpful. Student would likely still struggle.
- No: Gives the answer outright without guidance, or offers no meaningful help, such as generic reassurance, topic change, or silence.

Return your result explaining why the provided label is appropriate as structured JSON with these keys: `{“label_justification”: string}`

B Proprietary Model Label Prompt

To produce labels from both GPT-4o and Claude as our external baselines, we used the following prompt:

Data Label Prompt

You are an expert evaluator of Socratic-style tutoring dialogs in programming education. You will receive: `conversation_history` (full transcript up to, but NOT including, the tutor's latest reply, and the speaker turns are prefixed with "Student:" or "Tutor:"), and `last_response` (the tutor's latest reply, to be evaluated). Your job is to provide a label (one of Yes, To some extent, No indicating whether the reply provides adequate, partial, or no helpful guidance to the student. These labels correspond to:

- **Yes:** The reply gives sufficient, specific, actionable guidance or hints that directly help the student correct their error or deepen understanding.
- **To some extent:** Contains some guidance, but it is vague, incomplete, or only tangentially helpful. Student would likely still struggle.
- **No:** Gives the answer outright without guidance, or offers no meaningful help, such as generic reassurance, topic change, or silence.

Return your result explaining why the provided label is appropriate as structured JSON with these keys: { "label": string }

C SFT Training System Prompt

The system prompt used to align the model's behaviour to that of a professional tutor is as follows:

System Prompt

You are a professional tutor. Your goal is to focus on whether the Last Response from the example is providing enough guidance (i.e, explanation, hints, guidance) to the student to act upon, progressing the conversation based on the conversation history. DO NOT continue the conversation, and you MUST use the Last Response provided. Focus on these characteristics:

1. If the Last Response provides specific, actionable guidance that identifies exactly where errors occur and

offers clear steps forward, balancing encouragement with targeted correction while addressing misconceptions without giving away complete answers.

2. If the Last Response acknowledges problems but offer incomplete guidance—they might identify errors without explaining how to fix them, use ambiguous language, or address only part of the misconception, leaving students without clear direction on how to proceed.
3. If the Last Response fails to provide meaningful guidance by offering empty praise without addressing errors, changing the subject, reinforcing incorrect understanding, giving answers without explanation, or presenting completely irrelevant information that leaves students with no actionable path forward in solving their problem.

D LoRA Training Arguments

This details the full LoRA training parameters:

- `max_seq_length`: 2048
- `dtype`: cuda
- `load_in_4bit`: False
- `device`: cuda
- `device_map`: cuda:0
- `r`: 64
- `target_modules`: {q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj}
- `lora_alpha`: 64
- `lora_dropout`: 0
- `bias`: none
- `use_gradient_checkpointing`: unsloth
- `random_state`: 42
- `use_rslora`: True
- `loftq_config`: None

E GRPO Training Arguments

This details the full GRPO training arguments:

- use_vllm: **True**
- learning_rate: 5×10^{-6}
- adam_beta1: 0.9
- adam_beta2: 0.99
- weight_decay: 0.1
- warmup_ratio: 0.1
- lr_scheduler_type: cosine
- optim: paged_adamw_8bit
- logging_steps: 10
- bf16: **True**
- per_device_train_batch_size: 1
- gradient_accumulation_steps: 1
- num_generations: 6
- max_prompt_length: 2048
- max_completion_length: 256
- num_train_epochs: 5
- max_steps: -1
- save_strategy: steps
- save_steps: 250
- max_grad_norm: 0.1

TBA at BEA 2025 Shared Task: Transfer-Learning from DARE-TIES Merged Models for the Pedagogical Ability Assessment of LLM-Powered Math Tutors

Sebastian Gombert¹, Fabian Zehner^{1,2}, and Hendrik Drachsler^{1,3,4,5}

¹DIPF | Leibniz Institute for Research and Information in Education

²Centre for International Student Assessment (ZIB)

³studiumdigitale & ⁴Computer Science Department, Goethe University Frankfurt

⁵Department of Online Learning and Instruction, Open University NL Heerlen
{s.gombert, f.zehner, h.drachsler}@dipf.de

Abstract

This paper presents our contribution to the *BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-Powered Tutors*. The objective of this shared task was to assess the quality of conversational feedback provided by LLM-based math tutors to students regarding four facets: whether the tutors 1) identified mistakes, 2) identified the mistake’s location, 3) provided guidance, and whether they 4) provided actionable feedback. To leverage information across all four labels, we approached the problem with *FLAN-T5* models, which we fit for this task using a multi-step pipeline involving regular fine-tuning as well as model merging using the *DARE-TIES* algorithm. We can demonstrate that our pipeline is beneficial to overall model performance compared to regular fine-tuning. With results on the test set ranging from 52.1 to 68.6 in F1 scores and 62.2% to 87.4% in accuracy, our best models placed 11th of 44 teams in Track 1, 8th of 31 teams in Track 2, 11th of 35 teams in Track 3, and 9th of 30 teams in Track 4. Notably, the classifiers’ recall was relatively poor for underrepresented classes, indicating even greater potential for the employed methodology.

1 Introduction

Large language models, such as the ones from the *GPT* (Radford et al., 2018) or *Llama* (Grattafiori et al., 2024) families, have demonstrated remarkable capabilities in generating a wide range of textual content. This has resulted in their quick adoption in the educational space, where they are used for diverse purposes, such as assessing student-generated content, providing feedback and guidance, or generating exercise questions, among others (Wang et al., 2024). They have also been incorporated into intelligent tutoring systems, combining multiple of these features and capabilities into a single application (Wang et al., 2025). However, a core problem with these models is that they do

not guarantee accurate, practical, or focused output (Xu et al., 2025). As generation is handled through a combination of autoregression and probabilistic sampling, it cannot be guaranteed that each production of a given model is purposeful and correct.

Importantly, this can be a severe problem in educational settings. In the European Union, the EU AI Act (European Parliament and Council of the European Union, 2024) classifies AI-based systems in an educational context as high risk. What if a tutor provides a learner with incorrect feedback because of a chain of unfortunate random sampling during the corresponding generation process? What if specific prompt characteristics affect output quality systematically, disadvantaging certain learner groups (Hofmann et al., 2024; Salikutluk et al., 2024)? What if a given feedback text is not actionable, and a learner is left with more questions? One possibility to address a few, albeit not all, such problems is to deploy models tailored explicitly for policing the output of a given model. What is already an established practice with commercial models, where, for example, the generation of toxic content is policed, also has enormous potential for the educational sector, where policing by educational criteria is required.

The *BEA 2025 Shared Task on the Pedagogical Ability Assessment of AI-powered Tutors* (Kochmar et al., 2025) explores this idea for a narrow use case where the output of LLM tutors when assisting students with simple arithmetic problems is assessed. In particular, the goal is to assess communication records between students attempting to solve simple math problems and LLMs that assist them as tutors. The communication records are classified according to whether the LLM tutor identified student mistakes, recognised the mistake location, provided guidance, and whether the provided guidance is actionable. As highlighted by Holmes et al. (2022), ethical considerations in AI in education, despite their crucial impact, are of-

ten not prioritized. The present shared task, therefore, offers the opportunity to address a subset of vulnerabilities that could otherwise lead to ethical breaches.

Our submissions to this shared task are based on variants of *FLAN-T5-xl* (Chung et al., 2024) that underwent multiple steps of task-wise fine-tuning and model merging via *DARE-TIES* (Yu et al., 2024). On the shared task leaderboard, based on macro F1, our systems rank 11th out of 44 teams in Track 1, 8th out of 31 teams in Track 2, 11th out of 35 teams in Track 3, and 9th out of 30 teams in Track 4.

2 Background

2.1 Pedagogical Ability Assessment and Pedagogical Alignment of LLMs

Using conversational agents in education is not a novel idea; it has been explored for several years, e.g., in the form of tutors or assistants (Wollny et al., 2021). However, following the release of ChatGPT in 2022 and the resulting surge in research on conversational large language models, interest in this topic has increased (e.g., Pal Chowdhury et al. 2024). Although large language models have demonstrated remarkable capabilities and possess significant potential for educational use cases, their probabilistic nature also presents challenges that must be addressed before these models can be safely deployed in pedagogical contexts. Older conversational agents are often based on rules, fuzzy matching against a search space of expected inputs, information retrieval, and pre-defined answers and dialogue scripts (Wollny et al., 2021). This makes it easy to pedagogically align them since all output they can generate is pre-defined to a certain degree, or can, in the case of information retrieval, at least be curated.

For LLMs, this is not the case. While they can answer and react more dynamically and are better suited to providing deeply individualised feedback since they can deal with unforeseen inputs posing problems to more traditional chatbot designs, achieving alignment with pedagogical criteria is harder for these models. On the one hand, this is due to the well-known hallucination problem (Xu et al., 2024). On the other hand, even when a model does not hallucinate and generates correct output, this does not necessarily imply that what is generated follows good pedagogical practice¹, since

¹<https://benchmarks.ai-for-education.org/>; ac-

these models were never trained with the same in mind.

For this reason, there has been increased interest in studying and improving pedagogical alignment for large language models (LLMs). Sonkar et al. (2024) compared *supervised fine-tuning* (SFT) and *learning from human preference* (LHP; Christiano et al., 2017) as training approaches for achieving pedagogical alignment for LLMs, with the latter approach achieving overall better downstream results. Dai et al. (2023) assessed feedback generated by *ChatGPT* using the well-known Hattie framework (Hattie and Timperley, 2007) and concluded that feedback generated by the model was overall more detailed compared to a human gold standard with an overall high agreement in terms of what exact elements from Hattie’s framework were represented in the feedback texts. Meyer et al. (2024) found increased motivation and performance on a revision task as well as more positive feelings through LLM-generated feedback compared to no feedback. Tack et al. (2023) hosted a shared task that benchmarked the overall ability of LLMs to act as pedagogically sound tutors when fine-tuned or prompt-tuned for the same purpose. Maurya et al. (2025) introduced a framework to rate the qualities of LLM-based tutors using eight different dimensions, each rated on a three-level scale. Four of these dimensions form the basis for the dataset used in this shared task.

2.2 Model Merging

Model merging refers to a growing set of recently developed methods that combine multiple fine-tuned models into a single one, sharing all their strengths. The core idea behind model merging lies in what is called *task arithmetics* (Ilharco et al., 2023). If we interpret the set of all parameters of a given LLM as one long vector, we can define such vectors for both a pre-trained model (θ_0) as well as task-specific fine-tuned versions of the same (θ_t). By subtracting the initial vector θ_0 from the fine-tuned vector θ_t , we gain the so-called task vector θ'_t representing the knowledge a model acquired during a specific fine-tuning instance t . We can then create *merges* by combining the resulting task vectors in various ways and adding the resulting vector to the original pre-trained model.

A naive approach to recombining task vectors is to calculate a weighted mean of them. How-

cessed on 2025-05-21

ever, this comes with several problems that mainly stem from the nature of stochastic gradient descent, which can lead to different fine-tuned models converging to distinct local minima in the parameter space. While two datasets a given model might be fine-tuned with might be highly related, implying that the respective fine-tuned models will have learned similar underlying functions by having adjusted the weights of a given model similarly, it is by no means specific that these learned representations will be localized in the identical or corresponding parameters (e.g., polysemanticity). Colloquially speaking, two different fine-tuning instances might store different knowledge in the same parameters, resulting in parameter interference and decreased downstream performance.

For this reason, algorithms such as TIES (Yadav et al., 2023) and DARE-TIES (Yu et al., 2024), which improves on the previous, have been developed. While TIES aims at fitting a transformation matrix that acts as a translation layer between two different fine-tuning instances and strives to identify correspondences between the internal representations of both task vectors, *DARE-TIES* combines this with directional averaging and heavy pruning of the individual task vectors to minimize interference during merging. The method can be denoted in the following way, with t being a given task from the set of all tasks used for a particular merge T :

$$\theta_{\text{DARE}}^t = \text{DARE}(\theta_t, \theta_0), \text{ for } t \in T \quad (1)$$

$$\theta_M = \theta_0 + \lambda \sum_t^T \text{TIES}(\theta_{\text{DARE}}^t, \theta_0) \quad (2)$$

For the exact implementation of *DARE* and *TIES*, see Yu et al. (2024) respectively Yadav et al. (2023).

3 Method

3.1 Dataset

The dataset used in the BEA 2025 shared task contains conversation histories between an LLM, functioning as a tutor, and a corresponding human student. Each conversation history involves a simple math problem and reflects a corresponding conversation between a student and an LLM. While the training set includes 300 such conversation histories, the test set contains 191. For each conversation, there are up to seven different final responses that were each generated by a different model, such as *GPT-4* or *Mistral* (Jiang et al., 2023)

in response to the provided history. Moreover, human responses from both expert and novice tutors are provided for each conversation. For each of the responses, four of the overall eight dimensions from the framework introduced by Maurya et al. (2025) are annotated using a three-level scale (no, to some extent, yes). The dimensions are:

- **Mistake Identification:** Is the LLM able to identify the learner mistake in its response?
- **Mistake location:** Is the location of a given mistake provided in a response?
- **Providing guidance:** Does the model provide appropriate guidance on how to solve the mistake?
- **Actionability:** Is what the model answers actionable?

Each of the four dimensions corresponds to an individual evaluation track of the shared task. Due to time constraints and its conceptually distinct goal, we disregarded the fifth track that was concerned with identifying the generating LLM.

3.2 System Development

Our system uses *FLAN-T5* (Chung et al., 2024) models to model classification as a sequence-to-sequence task, where the model is trained to generate an output sequence containing the correct label for a given input, which includes the full conversation context, including all utterances of both student and tutor in a given conversation. Concretely, a model receives the following input for a given datapoint x and assessment dimension d (*mistake location, mistake identification, ...*), with h_x denoting the provided conversation history and r_x the provided tutor response:

$$I(x, d) = d : \text{history} : h_x \text{ response} : r_x \quad (3)$$

We did not make any structural modifications to the models themselves and used the standard implementations provided by the *Huggingface Transformers* framework (Wolf et al., 2020). The procedure we used to fit these models, however, distinguishes this work from other use cases of *FLAN-T5* for classification.

It involves three steps, as depicted in Figure 1. In a first step, the given *FLAN-T5* models were fine-tuned for three epochs, one model for each of the

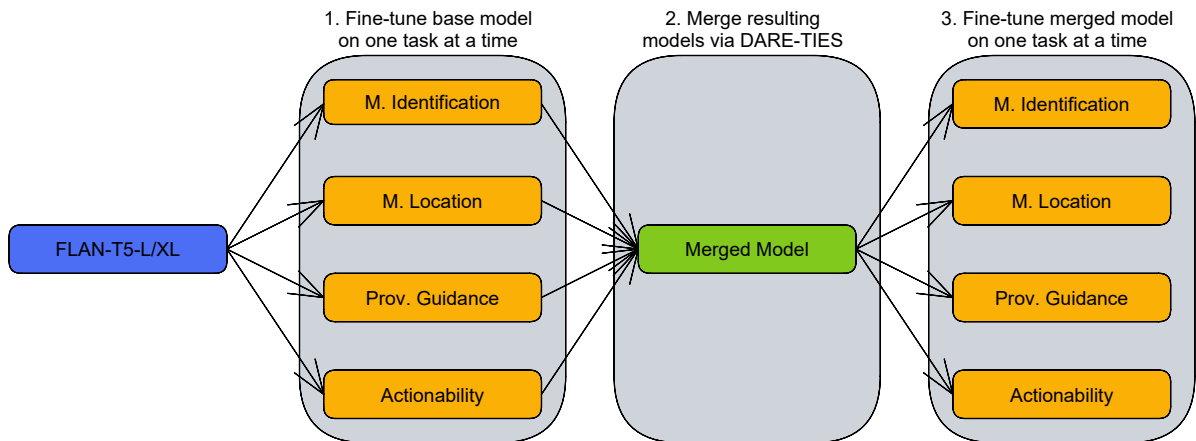


Figure 1: This figure depicts the overall training process we used during both our pre-experiments as well as for the final submission. First, *FLAN-T5* models are fine-tuned for three epochs for one dimension at a time. The resulting models are then merged using *DARE-TIES*. Lastly, the merged model serves as the basis for another round of task-specific fine-tuning, yielding the four final models.

four assessment dimensions, resulting in four individual models for *mistake identification*, *mistake location*, *provision of guidance*, and *actionability*. Fine-tuning was conducted using *Adagrad* as optimiser, a learning rate of $3e-4$, and a batch size of 4. These four models were then merged using the *DARE-TIES* (Yu et al., 2024) algorithm implementation provided by *Mergekit* (Goddard et al., 2024), with each model being uniformly weighted ($\lambda = 0.25$). The resulting model was then used as a basis for another round of fine-tuning, where we fine-tuned the merged model for each task individually again, resulting in another quartet of task-specific models.

The rationale behind this approach is the inherent interconnectedness between the four individual dimensions. We assumed that, for example, a mistake location can only reasonably be provided if a mistake is identified. Moreover, appropriate guidance can also be provided only if a mistake is identified. Then, only if guidance was provided at all can this guidance be actionable. Consequently, we assume that some of the parameters within models fine-tuned for one of these specific tasks likely encode information beneficial to the others.

DARE-TIES (Yu et al., 2024) as an algorithm enables us to exploit this property by merging multiple fine-tuned models into a single one that inherits the capabilities of all the used base models, with the possibility of even improving performance in some cases where the individual tasks are complementary to each other. This is achieved through the alignment and directional merging of the specific

Variant	MI	ML	PG	AC
Pre-merge	89.20	69.95	71.97	81.34
Merged	84.46	77.17	77.42	73.23
Post-merge	88.48	82.15	82.85	88.49

Table 1: Macro F1 scores for the three model stages in our pre-experiments. MI = Mistake Identification. ML = Mistake Location. PG = Providing Guidance. AC = Actionability.

model parameters.

Initially, we had assumed that the model resulting from the *DARE-TIES* merge would already be slightly stronger for each assessment dimension than the dimension-specific models. However, in our pre-experiments, we could not completely confirm this hypothesis. Using a 5x5 cross-validation setup with the complete training set, we fine-tuned and then merged *FLAN-T5-base* (Chung et al., 2024) models, with the result that the merged models showed a weaker performance for *mistake identification* and *actionability* than the dimension-specific models from which they were created (see Table 1), with an improved performance for *mistake location* and *providing guidance*.

For this reason, as a next step, we explored whether the resulting merged model would at least function as a reasonable basis for fine-tuning a next generation of dimension-specific models. As Table 1 shows, this was indeed the case, and the resulting dimension-specific models showed an improved performance over the merged variants as well as the dimension-specific models fine-tuned

Metric	MI	ML	PG	AC
Macro F1	68.58	54.90	52.12	66.71
Rank	11/44	8/31	11/35	9/30
Accuracy	87.40	73.24	66.52	73.24
Rank	5/44	6/31	5/35	4/30

Table 2: Results from the official shared task leaderboard. Rank indicates the rank our submissions achieved for the specific dimension and metric. MI = Mistake Identification. ML = Mistake Location. PG = Providing Guidance. AC = Actionability.

from *FLAN-T5-base*, except for *mistake location*. For this reason, we went with this procedure for our final submissions.

With the post-merge fine-tuning stage adding an epoch of training, performance gains may also have resulted from improved task-specific fitting rather than the merging process itself. While tentative experiments did not provide evidence for this, we did not rule this out through a systematic experiment.

4 Shared Task Submission and Evaluation

Following the intuition behind the scaling law that, on average, larger models show an improved downstream performance compared to smaller models when trained on the same data (Kaplan et al., 2020), we replicated our setup with *FLAN-T5-xl* (Chung et al., 2024) for the shared task submission. Again, we first fine-tuned dimension-specific models for all four dimensions for three epochs each, then merged them using *DARE-TIES* (Yu et al., 2024), and then used the resulting model as a basis for fine-tuning for another epoch to acquire again dimension-specific models (as depicted in Figure 1. Since, in our pre-experiments, the post-merge models for *mistake identification* were slightly outperformed by the pre-merge ones, we submitted results from both for the final task (since up to five submissions were allowed per dimension). Here, contrary to our pre-experiments, the post-merged version came out on top.

In the context of the shared task, the resulting models could all achieve upper mid-table results, going by *Macro F1*. For *Mistake Identification*, we placed 11th of 44 teams. For *Mistake Location*, we placed 8th of 31 teams. For *Providing Guidance*, we placed 11th of 35 teams. For *Actionability*, we placed 9th of 30 teams. For *Accuracy*, which served as a secondary evaluation metric, our models were among the best submissions in the shared

task. Here, we placed 5th, 6th, 5th and 4th for the respective dimensions.

These results suggest that our approach was overall highly successful in modelling the different dimensions, but, in particular, fell short for the *No* category, which was comparably underrepresented in the data. We assume that techniques such as *paraphrased oversampling* (Patil et al., 2022) would likely have helped combat that overall behaviour, but were not considered by us since we implemented our solution within one week under heavy time pressure. Table 2 shows the corresponding results. Overall, the results suggest that our approach is reasonable and the use of *DARE-TIES* merging allowed us to achieve upper mid-table results, although our placements suggest, that there are certainly better solutions for the problem than what we propose in this paper.

5 Conclusion

In this paper, we presented our submission to the *BEA 2025 Shared Task on the Pedagogical Ability Assessment of AI-powered Tutors*. Our submission combines fine-tuning and *DARE-TIES* merging *FLAN-T5-xl* models. In terms of macro F1, the primary evaluation metric used for the shared task, our models could only achieve upper mid table results, which is likely due to the underrepresentation of *No* and *to some extent* cases within the training set. In terms of overall accuracy, our submissions achieve more competitive results. Our general results show that combining *DARE-TIES* merging with fine-tuning can have beneficial results on downstream performance.

Limitations

Focus on *FLAN-T5*: In this paper, we focused solely on *FLAN-T5* models while not considering other models such as *Mistral-7b* (Jiang et al., 2023). The reason behind this was mainly that our contribution was created under heavy time pressure, so we wanted to focus on making our approach work for one model family as best as we could, instead of comparing a larger range of models.

No data augmentation used: Since the provided dataset is highly imbalanced, with the *no* and *to some extent* cases being underrepresented for all four dimensions, we assume that data augmentation could have likely benefited our systems, e.g., in the form of techniques such as *paraphrased oversampling* (Patil et al., 2022). However, due to the

heavy time pressure, we decided against exploring data augmentation.

No hyperparameter search: We did not conduct a hyperparameter search but instead stuck to the standard training hyperparameters used to pre-train the *FLAN-T5* models, except for the batch size, which we reduced from the original 64 to 4 due to limited computational resources. Similarly, it is possible that performance gains in the task-specific post-merge fine-tuning stem at least partly from an additional epoch of training.

References

- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- 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.
- 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.
- 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.
- Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. *Arcee’s MergeKit: A toolkit for merging large language models*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. 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.
- John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research*, 77(1):81–112.
- Valentin Hofmann, Pratyusha Ria Kalluri, Dan Jurafsky, and Sharese King. 2024. AI generates covertly racist decisions about people based on their dialect. *Nature*, 633(8028):147–154.
- Wayne Holmes, Kaska Porayska-Pomsta, Ken Holstein, Emma Sutherland, Toby Baker, Simon Buckingham Shum, Olga C. Santos, Mercedes T. Rodrigo, Mutlu Cukurova, Ig Ibert Bittencourt, and Kenneth R. Koedinger. 2022. *Ethics of AI in education: towards a community-wide framework*. *International Journal of Artificial Intelligence in Education*, 32(3):504–526.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. *Editing models with task arithmetic*. *arXiv preprint*. ArXiv:2212.04089 [cs].
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. *Mistral 7b*. *Preprint*, arXiv:2310.06825.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- 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.
- Jennifer Meyer, Thorben Jansen, Ronja Schiller, Lucas W. Liebenow, Marlene Steinbach, Andrea Horbach, and Johanna Fleckenstein. 2024. *Using LLMs to bring evidence-based feedback into the classroom: AI-generated feedback increases secondary*

- students' text revision, motivation, and positive emotions. *Computers and Education: Artificial Intelligence*, 6:100199.
- 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, L@S '24*, page 5–15, New York, NY, USA. Association for Computing Machinery.
- Annapurna P Patil, Shreekanth Jere, Reshma Ram, and Shruthi Srinarasi. 2022. T5w: A paraphrasing approach to oversampling for imbalanced text classification. In *2022 IEEE international conference on electronics, computing and communication technologies (CONECCT)*, pages 1–6. IEEE.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.
- Vildan Salikutluk, Elifnur Doğan, Isabelle Clev, and Frank Jäkel. 2024. Involving affected communities and their knowledge for bias evaluation in large language models. In *1st HEAL Workshop at CHI Conference on Human Factors in Computing Systems*, Honolulu, Hawaii, USA.
- 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, Miami, Florida, USA. Association for Computational Linguistics.
- Anais Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 785–795, Toronto, Canada. Association for Computational Linguistics.
- 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*.
- Tianfu Wang, Yi Zhan, Jianxun Lian, Zhengyu Hu, Nicholas Jing Yuan, Qi Zhang, Xing Xie, and Hui Xiong. 2025. Llm-powered multi-agent framework for goal-oriented learning in intelligent tutoring system. In *Companion Proceedings of the ACM on Web Conference 2025, WWW '25*, page 510–519, New York, NY, USA. Association for Computing Machinery.
- 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, 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.
- Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachslar. 2021. Are we there yet?-a systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2025. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.
- Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115.
- Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning, ICML'24*. JMLR.org.

LexiLogic at BEA 2025 Shared Task: Fine-tuning Transformer Language Models for the Pedagogical Skill Evaluation of LLM-based tutors

Souvik Bhattacharyya, Billodal Roy, Niranjana Kumar M, Pranav Gupta
Lowe’s

Correspondence: {souvik.bhattacharyya, billodal.roy, niranjan.k.m, pranav.gupta}@lowes.com

Abstract

While large language models show promise as AI tutors, evaluating their pedagogical capabilities remains challenging. In this paper, we, team LexiLogic presents our participation in the BEA 2025 shared task on evaluating AI tutors across five dimensions: Mistake Identification, Mistake Location, Providing Guidance, Actionability, and Tutor Identification. We approach all tracks as classification tasks using fine-tuned transformer models on a dataset of 300 educational dialogues between a student and a tutor in the mathematical domain. Our results show varying performance across tracks, with macro average F1 scores ranging from 0.47 to 0.82, achieving rankings between 4th and 31st place. Such models have the potential to be used in developing automated scoring metrics for assessing the pedagogical skills of AI math tutors.

1 Introduction

While significant progress has been made in making today’s large language models helpful, aligned, and responsible (Tan et al., 2023; Ji et al., 2023; Feng et al., 2024), their full potential in academic settings remains underutilized. Despite growing interest in using LLM-based AI tutors for academic support, traditional evaluation benchmarks tend to focus more on knowledge, factual accuracy, and reasoning (DeepSeek-AI et al., 2025; Abdin et al., 2025) rather than on the ability of these dialogue systems to function effectively in the role of a tutor. In educational contexts, there is a pressing need for systems and evaluation metrics specifically designed to assess complex pedagogical qualities. Therefore, it is essential to not only develop intelligent tutoring systems but also to evaluate them in terms of their ability to provide sufficient, helpful, and factually accurate guidance.

The shared task organized as part of the BEA workshop (Kochmar et al., 2025) focuses on educational dialogues between a student and a tutor in

the mathematical domain, specifically addressing student mistakes or confusion. The goal of the AI tutor is to help remediate these issues. The tutor responses, generated by the task organizers, come from a range of state-of-the-art LLMs with varying sizes and capabilities, including GPT-4 (OpenAI et al., 2023), Gemini (Reid et al., 2024), Sonnet (Anthropic, 2025), Mistral (Jiang et al., 2023), Llama 3.1 (Grattafiori et al., 2024) and Phi-3 (Abdin et al., 2024a). In addition to the generated responses, the development set includes annotations evaluating their quality across several pedagogically motivated dimensions: Mistake Identification, Mistake Location, Providing Guidance, Actionability, and Tutor Identification.

Across all tracks of the shared task, we approached the problems as classification tasks and followed a fine-tuning approach using several transformer-based encoder and decoder models. Table 1 summarizes the performance of our submitted models compared to the top-performing entries in terms of macro average F1 score in each task.¹

Track	Our Score	Best Score
Track 1	0.65	0.72
Track 2	0.48	0.60
Track 3	0.47	0.58
Track 4	0.69	0.71
Track 5	0.82	0.95

Table 1: Performance of our models compared to the best scores in each track.

2 Related Work

With the widespread use of large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Team et al., 2025) as conversational systems in educational contexts, several studies have evaluated

¹The code for this work is available at <https://github.com/prannerta100/acl-bea2025-workshop-st>

their pedagogical capabilities. There are numerous LLM evaluation metrics such as BLEU (Papineni et al., 2002), BERTScore (Zhang et al., 2020), ROUGE (Lin, 2004), DialogRPT (Gao et al., 2020), etc., which are not necessarily designed to assess an LLM’s educational or pedagogy-related capabilities (Jurenka et al., 2024) and shown to have relatively low correlation with human judgments (Liu et al., 2023). This highlights the need for alternative methods to evaluate LLM performance in educational settings. One such approach is to use human annotators to rate LLM responses based on various criteria (Collins et al., 2023; Shen and Wu, 2023; Lee et al., 2024). While human evaluators can consider context, tone, and pedagogical effectiveness, offering qualitative insights that go beyond quantitative metrics, they are also prone to bias, and the process tends to be time-consuming and relatively expensive.

At the other end of the spectrum, there is growing interest in automated evaluation systems and LLM-as-a-judge approaches (Jurenka et al., 2024). Chen et al. (2023)’s experimental results show that ChatGPT is capable of evaluating text quality effectively from various perspectives without reference, and it demonstrates superior performance compared to most existing automatic metrics. Macina et al. (2025) developed MATHTUTORBENCH to score the pedagogical quality of open-ended teacher responses and also trained several LLM-based reward models, showing that these models can distinguish expert from novice teacher responses with high accuracy. TUTUREVAL, a diverse question-answering benchmark, was released by Chevalier et al. (2024), who evaluated the capabilities of several open-weight and proprietary LLMs using GPT-4 as the evaluator. Maurya et al. (2025a) introduced MRBench, which includes a large set of student–tutor conversations from seven state-of-the-art LLM-based and human tutors, and evaluated them across various dimensions using a different set of LLMs. Jurenka et al. (2024) also introduced LearnLM-Tutor, a fine-tuned model that was consistently preferred over base models for various academic tasks as judged by LLM-based critics.

3 Task Description and Methodology

The dataset provided for the shared task (Maurya et al., 2025b) consisted of conversation history between a tutor and a student along with a final response from the tutor based on which the vari-

ous pedagogical capability label is to be predicted. There were a total of 300 distinct conversations out of which we chose 50 to include in our test set, which resulted in 2067 training data points and 409 test data points. The same train-test split is used in all our experiments.

3.1 Track 1 - Mistake Identification

Track 1 of the shared task aims to develop systems that can identify whether a tutor’s response acknowledges mistakes in a student’s answer. The distribution of three categories in this track is detailed in Table 2. Each data point consists of a conversation history between a tutor and a student, along with a final response from the tutor. Participants are required to assess whether the tutor’s reply explicitly recognizes the student’s mistake within the conversation.

Tutor	Yes	No	To some extent
GPT4	234	15	1
Gemini	215	21	14
Sonnet	212	20	18
Phi-3	68	176	6
Mistral	223	10	17
Llama318B	202	31	17
Llama31405B	239	7	4
Expert	188	15	47
Novice	28	11	28
Total	1609	306	152

Table 2: Distribution of instances across categories for each tutor in the dataset in Track 1

For this task, our experiments involved fine-tuning various encoder and decoder models. The input sequence was formed by concatenating the conversation history with the final response, and we replaced the model’s un-embedding layer with a classification head for the three target classes. The models we used included FLAN-T5 (Chung et al., 2022), ModernBert (Warner et al., 2024), Llama 3.2 (Grattafiori et al., 2024), Phi-4 (Abdin et al., 2024b), and Qwen-2.5 (Qwen et al., 2025). All models were trained for 10–15 epochs with an initial learning rate between $5e-5$ and $1e-4$, using an exponential learning rate scheduler, a batch size of 8–10, and a gradient accumulation step of 2. On the test set, FLAN-T5-large performed the best, achieving a macro average F1 score of 0.65 and placing us 22nd among 44 submissions on the official leaderboard. The training and test set performance of all models is presented in Table 3 (with

the train set F1 scores corresponding to the epoch with the highest test set performance).

Model	Train F1	Test F1
FLAN-T5-large	0.94	0.65
ModernBERT-large	0.98	0.61
Llama-3.2-3B	0.99	0.62
Phi-4-mini-instruct	1.0	0.63
Qwen2.5-7B-Instruct	0.73	0.55

Table 3: Strict macro average F1 scores of different models on training and test datasets of Track 1

3.2 Track 2 - Mistake Location

In subtask 2, the objective is to develop a system capable of identifying whether a tutor’s response effectively locates the mistake in the student’s answer and provides a clear explanation of the error. This includes assessing whether the tutors’ responses accurately point to a genuine mistake and its location in the students’ responses. The distribution of each labels across different categories for each tutor in the training dataset is shown in Table 4.

Tutor	Yes	No	To some extent
GPT4	242	37	13
Gemini	176	93	31
Sonnet	207	60	33
Phi-3	73	223	4
Mistral	216	52	35
Llama318B	161	108	31
Llama31405B	252	33	15
Expert	197	58	45
Novice	19	60	2
Total	1543	724	209

Table 4: Distribution of instances across categories for each tutor in the dataset for Track 2

The final response is concatenated with the conversation history and fed as input into our model. Our experimental setup predominantly focused on transformer-based encoder and decoder models. In both encoder-decoder and large language model (LLM) configurations, we modify the original models by removing the final un-embedding layer and replacing it with a classification head. Among the encoder-based models, we evaluated ModernBERT (Warner et al., 2024), and MathBERT (Peng et al., 2021). For large language models, we conducted experiments with Llama 3.2 (Grattafiori et al., 2024), Phi-4 (Abdin et al., 2024b), and Qwen-

2.5 (Qwen et al., 2025).

We fine-tuned all models for a maximum of 10 epochs, with an initial learning rate in the range of $2e-2$ to $5e-5$, an exponential learning rate scheduler with gamma set between 0.9 and 0.9375 with a batch size between 4 and 12, with gradient accumulation steps set to 2. During training we minimized the categorical cross-entropy loss. In Table 5, we report the strict Macro average F1 scores of various models. The reported training set F1 scores correspond to the epoch with the highest F1 score on the test set. On the held-out test set, our submission based on Phi-4-mini-instruct achieved an F1 score of 0.48 on the unseen test dataset placing us at the 23rd position out of total 31 submissions.

Model	Train F1	Test F1
MathBERT	0.67	0.5
ModernBERT-large	0.72	0.52
Llama-3.2-3B	0.73	0.55
Llama-3-8B	0.71	0.53
Phi-4-mini-instruct	0.78	0.68
Qwen2.5-7B-Instruct	0.67	0.55

Table 5: Strict Macro average F1 scores of different models on training and test datasets

3.3 Track 3 - Providing Guidance

Track 3 focuses on evaluating whether a tutor’s response provides effective guidance to help students understand and correct their mistakes. This task goes beyond simply identifying and locating errors to assess the pedagogical quality of the tutoring response. The system must determine if the tutor offers constructive feedback, explanations, or suggestions that would help the student learn from their mistakes. Similar to the previous tracks, the task includes three categories: ‘Yes’, ‘No’, and ‘To some extent’, with their distribution across different tutors shown in Table 6.

Our approach for this track followed a similar methodology to the previous tasks, where we concatenated the conversation history with the final tutor response and fed it as input to our classification models. The experimental setup involved fine-tuning various transformer-based models to classify the quality of guidance provided in tutor responses.

We evaluated several model architectures including both encoder-only and decoder-only models. Among the encoder-based models, we experimented with ModernBERT (Warner et al., 2024),

Tutor	Yes	No	To some extent
GPT4	228	41	31
Gemini	168	47	85
Sonnet	184	52	64
Phi-3	51	189	60
Mistral	189	47	64
Llama318B	134	65	101
Llama31405B	238	16	46
Expert	205	47	48
Novice	10	62	4
Total	1407	566	503

Table 6: Distribution of instances across categories for each tutor in the dataset for Track 3

while for large language models, Phi-4 (Abdin et al., 2024b), and FLAN-T5 (Chung et al., 2022). All models were modified by replacing the final un-embedding layer with a three-way classification head corresponding to our target categories. The train and test F1 values are in Table 7.

The training configuration involved fine-tuning for 8-12 epochs with learning rates ranging from $1e-5$ to $8e-5$, using an exponential learning rate scheduler with gamma values between 0.85 and 0.95. We employed batch sizes of 6-14 with gradient accumulation steps of 2, and optimized using categorical cross-entropy loss. The performance of different models on both training and test sets is presented in Table 7, where the training F1 scores correspond to the epoch achieving the highest test set performance.

Model	Train F1	Test F1
FLAN-T5-large	0.92	0.36
ModernBERT-large	0.89	0.39
Phi-4-mini-instruct	0.97	0.45

Table 7: Strict Macro average F1 scores of different models on training and test datasets for Track 3

Our best performing model, Phi-4-mini-instruct, achieved a macro average F1 score of 0.47 on the test set, securing the 31st position out of 35 total submissions on the official leaderboard. The relatively lower performance across all models suggests that evaluating the quality of pedagogical guidance is inherently more challenging than simple mistake identification, as it requires understanding the nuanced aspects of effective tutoring strategies and educational support.

3.4 Track 4 - Actionability

In Track 4, the goal is to develop system to identify whether the tutor’s response is clear in regards to what the student should do next, i.e., whether or not the tutor response was vague, unclear or a conversation stopper. Table 8 shows the distribution of instances across different categories for each tutor in the training dataset provided.

Tutor	Yes	No	To some extent
GPT4	116	125	9
Gemini	142	52	56
Sonnet	141	74	35
Phi-3	27	215	8
Mistral	168	43	39
Llama318B	106	93	51
Llama31405B	182	40	28
Expert	200	18	32
Novice	3	52	12
Total	1085	673	309

Table 8: Distribution of instances across categories for each tutor in the dataset in Track 4

We use as an input the sequence of tokens after the final response from the tutor is appended with the original conversation. We experimented with multiple transformer based encoder and decoder models in this task as well. In all the experiments, we remove the final un-embedding layer from the original models and replace it with a classification head producing three dimensional logits corresponding to the three available classes in this task. Among the encoder models we have experimented with FLAN-T5 (Chung et al., 2022), ModernBert (Warner et al., 2024) and MathBERT (Peng et al., 2021) and among the LLMs we tried Llama 3.2 (Grattafiori et al., 2024), Phi-4 (Abdin et al., 2024b) and Qwen-2.5 (Qwen et al., 2025).

We fine-tune all the models for 15–20 epochs, using an initial learning rate in the range of $5e-5$ to $1e-4$, with an exponential learning rate scheduler (gamma set to 0.9). We use a batch size between 8 and 12, gradient accumulation steps of 2, and minimize the categorical cross-entropy loss. In Table 9, we report the Strict macro average F1 scores of various models. Note that the reported training set F1 scores correspond to the epoch with the highest test set F1 score. In the held-out test set, our submission based on Phi-4-mini-instruct scored an F1 score of 0.69 securing us the 4-th place among 29 submissions in the official leaderboard.

Model	Train F1	Test F1
FLAN-T5-base	0.76	0.59
MathBERT	0.98	0.58
ModernBERT-large	1.0	0.67
Llama-3.2-3B	0.97	0.61
Llama-3-8B	0.74	0.55
Phi-4-mini-instruct	1.0	0.71
Qwen2.5-7B-Instruct	1.0	0.65

Table 9: Strict macro average F1 scores of different models on training and test datasets of Track 4

3.5 Track 5 - Tutor Identification

The goal of track 5 was to predict the identity of the tutor for a given response, from a set of 9 identities, such as Sonnet, Llama3.1 8B, Llama 3.1 405B, GPT4 to name a few. We mainly fine-tuned various transformer models for this task with a similar setup to the previous tasks, and have reported our scores in Table 10. We observed that for many models the per-class F1 score for Novice, Llama 3.1 405B and 8B was lower than other classes. For the Novice class, a possible cause could be the lack of enough Novice examples in the dataset. We did not investigate the cause for the low performance for Llama 3.1 8B and 405B in detail, but when we looked at the test set confusion matrix for one of the models, we found that there was significance confusion between Llama 3.1 8B and 405B. It would be interesting to investigate how much of these similarities are task-specific and how much are specific to the base model. A recent preprint (Smith et al., 2025) suggests similar patterns in cosine similarities between the outputs of various LLMs. Note that these metrics are reported on our hold out sets and not the leaderboard test sets. Our best leaderboard test set performance was 0.82, and our final leaderboard position was 16th according to the macro average F1 metric.

4 Conclusion

In this work, we presented our experiments using a fine-tuning-based approach with several encoder-based and large language models to evaluate the pedagogical capabilities of AI tutors. We observed that different LLMs yield varying performance levels, highlighting model-specific behavior. Some class labels in the training data had very few examples, which may have impacted performance. Future work could explore data augmentation and sampling techniques to address this imbalance and

Model	Train F1	Test F1
FLAN-T5-base	0.76	0.59
ModernBERT-large	0.99	0.84
Phi-4-mini-instruct	1.0	0.78
Llama-3.2-3B	1.0	0.85
Longformer	-	0.83*
BigBird Roberta Large	-	0.79*
MathBERT	-	0.79*

Table 10: Macro average F1 scores of different models on training and test datasets of Track 5

*: test set drawn from the same distribution but might differ from the other models

potentially improve results. It would also be worthwhile to investigate prompt-based classification methods for evaluating tutor responses in zero-shot or few-shot settings, and explore the use of the models reported in this paper as reward models for post-training or performing test-time scaling on LLMs for improving their pedagogical skills. Additionally, future research could examine the potential of using the same set of AI tutors to reflect on and revise their responses to better align with the goals of effective and helpful AI tutoring systems.

5 Limitations

Automated scoring metrics for evaluating the pedagogy of AI math tutors and AI tutors in general come with their own limitations. Bias introduced by the finetuned model and the underlying pre-trained model can lead certain behaviors to be reinforced and certain demographics to be highlighted over other demographics. Cultural considerations also play an important part in pedagogy. A lack of rigorous theoretical guarantees on the mathematical and conceptual accuracy of LLM models can propagate incorrect concepts among students and lead to unwanted friction with instructors. Accessibility of AI tutoring tools could be a barrier for some students with limited resources and internet access, given the resource-expensive nature of LLMs. Moreover, AI tutoring tools typically require students to access internet on their phone or computer, enhancing their risk of being exposed to other websites and social media, causing the risks to outweigh the benefits.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen,

- Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, Piero Kauffmann, Yash Lara, Caio César Teodoro Mendes, Arindam Mitra, Bismira Nushi, Dimitris Papailiopoulos, Olli Saarikivi, Shital Shah, Vaishnavi Shrivastava, and 4 others. 2025. [Phi-4-reasoning technical report](#). *Preprint*, arXiv:2504.21318.
- Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024a. [Phi-3 technical report: A highly capable language model locally on your phone](#). *Preprint*, arXiv:2404.14219.
- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024b. [Phi-4 technical report](#). *Preprint*, arXiv:2412.08905.
- Anthropic. 2025. Claude 3.7 sonnet. Available at <https://www.anthropic.com/claude/sonnet>.
- 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](#). *Preprint*, arXiv:2005.14165.
- Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. [Exploring the use of large language models for reference-free text quality evaluation: An empirical study](#). *Preprint*, arXiv:2304.00723.
- Alexis Chevalier, Jiayi Geng, Alexander Wettig, Howard Chen, Sebastian Mizera, Toni Annala, Max Jameson Aragon, Arturo Rodríguez Fanlo, Simon Frieder, Simon Machado, Akshara Prabhakar, Ellie Thieu, Jiachen T. Wang, Zirui Wang, Xindi Wu, Mengzhou Xia, Wenhan Xia, Jiatong Yu, Jun-Jie Zhu, and 3 others. 2024. [Language models as science tutors](#). *Preprint*, arXiv:2402.11111.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2022. [Scaling instruction-finetuned language models](#). *Preprint*, arXiv:2210.11416.
- Katherine M. Collins, Albert Q. Jiang, Simon Frieder, Lionel Wong, Miri Zilka, Umang Bhatt, Thomas Lukasiewicz, Yuhuai Wu, Joshua B. Tenenbaum, William Hart, Timothy Gowers, Wenda Li, Adrian Weller, and Mateja Jamnik. 2023. [Evaluating language models for mathematics through interactions](#). *Preprint*, arXiv:2306.01694.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. [Don't hallucinate, abstain: Identifying llm knowledge gaps via multi-llm collaboration](#). *Preprint*, arXiv:2402.00367.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. [Dialogue response ranking training with large-scale human feedback data](#). *Preprint*, arXiv:2009.06978.
- 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.
- Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. [Beavertails: Towards improved safety alignment of llm via a human-preference dataset](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 24678–24704. Curran Associates, Inc.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, and 1 others. 2023. [Mistral 7b](#). *arXiv preprint arXiv:2310.06825*.
- 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.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.

- Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, Rose E. Wang, Minae Kwon, Joon Sung Park, Hancheng Cao, Tony Lee, Rishi Bommasani, Michael Bernstein, and Percy Liang. 2024. [Evaluating human-language model interaction](#). *Preprint*, arXiv:2212.09746.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yang Liu, Dan Iter, Yichong Xu, Shuhang Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-eval: Nlg evaluation using gpt-4 with better human alignment](#). *Preprint*, arXiv:2303.16634.
- 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](#). *Preprint*, arXiv:2502.18940.
- Kaushal Kumar Maurya, KV Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025a. [Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors](#). *Preprint*, arXiv:2412.09416.
- Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025b. [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.
- OpenAI and 1 others. 2023. [Gpt-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. [Mathbert: A pre-trained model for mathematical formula understanding](#). *Preprint*, arXiv:2105.00377.
- Qwen, :, 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, Jiayi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, and 1 others. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context](#). *arXiv preprint arXiv:2403.05530*.
- Hua Shen and Tongshuang Wu. 2023. [Parachute: Evaluating interactive human-llm co-writing systems](#). *Preprint*, arXiv:2303.06333.
- Brandon Smith, Mohamed Reda Bouadjeneq, Tahsin Alamgir Kheya, Phillip Dawson, and Sunil Aryal. 2025. [A comprehensive analysis of large language model outputs: Similarity, diversity, and bias](#). *Preprint*, arXiv:2505.09056.
- Xiaoyu Tan, Shaojie Shi, Xihe Qiu, Chao Qu, Zhenting Qi, Yinghui Xu, and Yuan Qi. 2023. [Self-criticism: Aligning large language models with their understanding of helpfulness, honesty, and harmlessness](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 650–662, Singapore. Association for Computational Linguistics.
- 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.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- 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.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). *Preprint*, arXiv:1904.09675.

IALab UC at BEA 2025 Shared Task: LLM-Powered Expert Pedagogical Feature Extraction

Sofía Correa Busquets^{1,2,3}, Valentina Córdova Véliz^{1,3}, Jorge Baier^{1,2,3},

¹Pontificia Universidad Católica de Chile, ²Millenium Institute Foundational Research on Data,

³National Center for Artificial Intelligence

{sbcorrea, avcordova, jbaier}@uc.cl

Abstract

As AI’s presence in educational environments grows, it becomes critical to evaluate how its feedback may impact students’ learning processes. Pedagogical theory, with decades of effort into understanding how human instructors give good-quality feedback to students, may provide a rich source of insight into feedback automation. In this paper, we propose a novel architecture based on pedagogical-theory feature extraction from the conversation history and tutor response to predict pedagogical guidance on MRBench. Such features are based on Brookhart’s canonical work in pedagogical theory, and extracted by prompting the language model LearnLM. The features are then used to train a random-forest classifier to predict the Track 3: Pedagogical Guidance of the BEA 2025 shared task. Our approach ranked 8th in the dimension’s leaderboard with a test Macro F1-score of ~ 0.54 . Our work provides some evidence in support of using pedagogical theory qualitative factors treated separately to provide clearer guidelines on how to improve low-scoring intelligent tutoring systems. Finally, we observed several inconsistencies between pedagogical theory and MRBench’s inherent relaxation of the tutoring problem implied in evaluating on a single-conversation basis, calling for the development of more elaborate measures which consider student profiles to serve as true heuristics of AI tutors’ usefulness.

1 Introduction

As part of the AI revolution, AI tutors will gain a growing role in education. Their use, however, should be preceded by rigorous evaluation, as omitting this step would be as unthinkable as hiring untrained teachers. To contribute to the development of evaluation standards for AI tutors, this paper describes an approach to automatically classify certain aspects of pedagogical ability on the Mistake Remediation Benchmark (MRBench) dataset of grade-school math tutoring chats (Maurya et al.,

2025a). The dataset contains annotations for the dimensions of identifying that the student has made a mistake, correctly individualizing what that mistake was, providing the student with relevant and helpful guidance, and cueing the student on how to follow the conversation. Of these, our approach attempts to classify whether feedback did, did not, or did to some extent, provide pedagogical guidance (PG) on the, Track 3: Pedagogical Guidance of the BEA 2025 shared task (Kochmar et al., 2025).

PG as an object of study is richly explored in the theory of pedagogy. For instance, the area of math didactics has studied phenomena such as students’ capacity to grasp concepts progressing from the concrete, to the pictorial, to the abstract (Bruner, 1966); how to develop an academic math discourse to support understanding (Chapin et al., 2009); and best practices for orchestrating productive student discussions (Smith and Stein, 2011). Also, assessment theory compiles frameworks on how to construct feedback as a powerful tool to improve student understanding and performance (Brookhart, 2008; Tunstall and Gipps, 1996). Our approach attempts to transfer knowledge from pedagogical theory by proposing a set of engineered features for PG classification strongly based on Brookhart’s work. With these features in hand, we propose a two-phase classification process. In the first phase, we use an LLM to query the text, which includes the conversation history between the student and the tutor, for the presence, or lack thereof, of our features in the tutor’s feedback. In the second phase, we use a random-forest classifier which is given a binary vector representing the output of the previous phase and attempts to predict the PG dimension.

2 Related Work

MRBench’s dimensions on which to assess the pedagogical ability of AI tutors result from the distil-

lation of a body of previous work in NLP addressing ITS evaluation (Tack and Piech, 2022; Macina et al., 2023; Daheim et al., 2024; Wang et al., 2024). Tack and Piech (2022), in their “AI Teacher Test” evaluated the dialogic pedagogical ability of certain LLMs in a mathematics-domain educational dialogue from the dimensions of whether they speak like a teacher, understand a student, and help a student. Specifically in math mistake remediation in the tutoring context, Macina et al. (2023) dimensions included coherence, correctness, and equitable tutoring. In the same context, Daheim et al. (2024) create the dimensions of targetedness, correctness, and actionability. Finally, and also within said context, Wang et al. (2024) put forth usefulness, care, and humanness. Maurya et al. (2025b) compile MRBench to address this need for a unified evaluation framework, and the present Shared Task is proposed as a challenge because all the aforementioned work is not, as of yet, fully independent from manual evaluation.

3 Preliminaries

To determine qualities that make feedback effective, the pedagogical perspective generally follows Brookhart’s (2008) four-dimension framework: content, specificity, timing and audience. Rather than assigning intrinsic value to hints, explanations or other information the tutor might provide, these dimensions promote that feedback’s potential depends on every point that it communicates complying with certain characteristics. For example, when amending any student misconceptions (content-focus), to unambiguously identify the misconception (specificity-clarity), feedback should explicitly distinguish it from what the student has understood correctly (content-valence). The same would be true for the offering of procedural guidance (content-focus): a hint about the right direction may confuse the student into undoing correct steps taken. Furthermore, the clarity of all the aforementioned depends on the student’s level of prior knowledge (audience-individual), which in this case we may approximate as the school year. This framework thus offers a theoretically grounded approach to tackle the interdependence of feedback dimensions in function of the ultimate goal: helping the student.

4 Methodology

To distill a set of features from pedagogical theory, we first asked the virtual assistant Claude (Anthropic, 2024) to create a feedback checklist from the key takeaways of seminal books on assessment and math didactics (Chapin et al., 2009; Smith and Stein, 2011; Brookhart, 2008; Tunstall and Gipps, 1996). Second, we merged redundant points together and discarded factors that were outside the scope of MRBench: anything that required knowing the student personally, communicating non-verbally and/or interacting in a classroom environment. The few remaining factors came chiefly from Brookhart (2008). Third, we stress-tested these for what we anticipated as possible AI tutor failures and edited accordingly by hand. For example, we added “accurately and specifically” at the beginning of Claude’s sentence “identify what was done correctly before addressing the error”. Then, each quality was separated into its own feature (identifies / identifies accurately / identifies specifically), so that binary tags on these features would be as informative as possible. Fourth, we phrased each feature as a yes-or-no question to prompt LearnLM (Team et al., 2024). Finally, we performed prompt engineering on the questions using a subset of 20 random tutor responses. The full resulting list of questions is available in Appendix A.

5 Architecture

Our proposed architecture, shown in Figure 1, is composed of two models working sequentially: a feature extractor, and a classifier.

First, features are extracted by prompting LearnLM (Team et al., 2024), a domain-specific Gemini fine-tuning, currently in experimental phase. We chose this model because of its expert training on tutoring data and pedagogical theory sources. For each feature, the conversation history is concatenated to the tutor response and a yes-or-no question representing the feature (see Appendix B), to which the model is prompted to respond with a binary 0/1 tag. Since preliminary tests yielded no relevant difference resulting from temperature variation, the model’s hyperparameters were left at their default values. The full feature extraction prompt is in Appendix B.

To accommodate the low dimensionality of the data, decision tree (DT) and random forest (RF) models were included in the trials for the final classifier. These were chosen for their structural

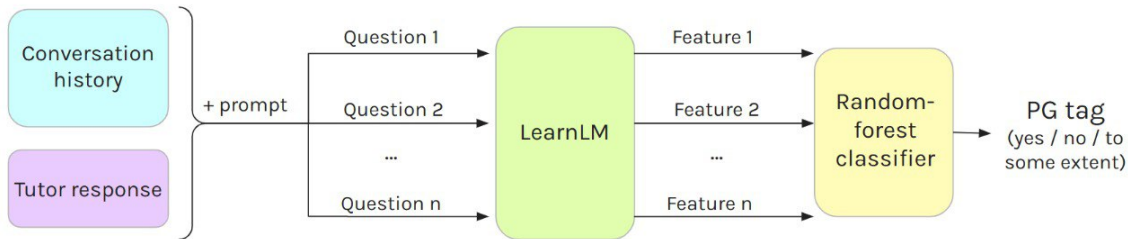


Figure 1: Proposed architecture to classify using expert pedagogical features.

mimicking of the decision process that pedagogy professionals described while annotating sample data.

6 Results

The variety of classifiers trained resulted from a different selection of extracted features as input, hyperparameter combinations, and the choice of DT versus RF model. A total 17320 DTs and 1400 RFs were trialed, each with 5-fold cross-validation, and the best candidates were then iterated using SMOTE oversampling. The highest performing model was an RF excluding some features from the input data, the hyperparameters of which we include in [Appendix C](#), with training metrics detailed in [Table 1](#).

Phase	Exact macro F1	Exact accuracy	Lenient macro F1	Lenient accuracy
Train	0.5662	0.6373	0.7529	0.8214
Test	0.5369	0.6244	0.7379	0.7822

Table 1: Performance of selected classifier model.

The final architecture using this model ranked 8th in the leaderboard for the pedagogical guidance dimension, with test metrics detailed in [Table 1](#).

7 Conclusions

We have presented an approach to PG classification that combines LLMs and traditional AI techniques with a theoretical framework on PG. The features we propose offer a perspective that considers the interdependence of the original MRBench dimensions, but puts them all in service of how well the tutor guides the student.

Our work shows the potential of using PG-theory-based features, which is a fine-grained way of assessing elements of good-quality feedback while exploiting an LLM. Future work should explore other ways in which identification of these

features may be exploited to iterate the construction of good-quality feedback via LLMs. In addition, we think that PG theory invites developers of AI tutors to take two other complementary routes for future work. The first is to design tutors aware of learning objectives, since this is fundamental to understand how to guide the student. The second is that AI tutors should build and exploit a student profile over time, considering the student’s previous knowledge, degree of metacognition, and learning strategies that have previously worked or failed. Tackling these two action points would expand the frontier of AI tutor evaluation beyond the biggest limitations of this work from the standpoint of PG theory.

Limitations

Our architecture first assumes the limitations of our theoretical alignment: following [Brookhart \(2008\)](#) may better describe certain Western learning contexts than other sociocultural realities. Then, the architecture’s reliance on LearnLM means it inherits any of the model’s possible inaccuracies and biases, and that implementation depends on proprietary API use. Finally, the classifier model’s performance should be improved with further trials using cleaned and augmented data.

Regarding the last point, the MRBench dataset carries limitations that transfer to our architecture. In terms of quality, we found conversation histories that we considered to be noisy: some lacking the original word problem being solved, with alternative tutor responses embedded within, or exchanging tutor/student speaker tags. We also did not find tagging criteria to be self-evident: the question of what constituted relevant “explanation, elaboration, hint, examples, and so on” seemed both open and necessitating at least some degree of expert pedagogical knowledge. Finally, the dataset is limited to the English language, mathematics school subject, arithmetic content and grade-school instruc-

tion level. Asymmetric advances in low-resource languages and higher influence of culture in other subjects of instruction limit the applicability of the benchmark for the range of intelligent tutoring systems currently on the market.

Finally, the strongest limitation surrounding this shared task was scarcity of context. In the pedagogical theory that we reviewed and that we believe is key to incorporate to these systems, the majority of factors contributing to PG are considered to be based on the student as a subject of learning. As such, factors that are regarded as key to PG are the student's individual previous knowledge, metacognitive ability, optimal learning strategies, personal relationship to the contents being taught, role in the classroom social dynamics, and sociocultural context (Brookhart, 2008; Smith and Stein, 2011; Chapin et al., 2009). Though we expanded as much as possible on the factors inferable from a single conversation via text, existing PG literature would suggest that an AI tutor's quality of PG can only be realistically estimated against a constructed learner profile of the student. Moreover, these considerations all defined their value only in relation to learning objectives and how they advanced the student towards them, meanwhile the context of what learning objectives were being reinforced in the tutoring sessions was not present in the MRBench dataset.

Acknowledgments

This work was supported by the National Center for Artificial Intelligence (CENIA) under Grant FB210017 Basal ANID. We would like to thank all of the following people. For their input to this work: Francisco Gazitúa and Juan Pablo Fuentes. For their support in timely experiment execution: to our colleagues at IALab, friends and family. For their contribution of expert knowledge in pedagogy: Francisca Ubilla, Edgar Valencia, Carolina Véliz, Teresita Fuentes, Emilia Deichler, Chiara Hiraizumi, Javier Riquelme, Marcelo Mena, Constanza Del Solar, Camila Sánchez, Andrea Vilca and Martín Pino. For their contribution of non-expert pedagogy knowledge: Luzhania Céspedes, Joaquín Handal, Guillermo Staudt, Raimundo Labbé, Daniel Villaseñor, Alexandre Icaza, Cristobal Soto, Vicente Muñoz, Victor Marques, José Chong, Matías Valenzuela, Maximiliano Berríos and Maximiliano Navia.

References

- Anthropic. 2024. [The Claude 3 Model Family: Opus, Sonnet, Haiku](#). Technical report, Anthropic.
- Susan M. Brookhart. 2008. *How to Give Effective Feedback to Your Students*. ASCD, Alexandria, VA.
- Jerome Seymour Bruner. 1966. *Toward a Theory of Instruction*. Belknap Press of Harvard University.
- Suzanne H. Chapin, Catherine O'Connor, and Nancy Canavan Anderson. 2009. *Classroom Discussions: Using Math Talk to Help Students Learn*. Math Solutions, California.
- 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](#). *Preprint*, arXiv:2407.09136.
- Ekaterina Kochmar, Kaushal Kumar 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*.
- 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](#). *Preprint*, arXiv:2305.14536.
- Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025a. [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.
- Kaushal Kumar Maurya, KV Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025b. [Unifying AI Tutor Evaluation: An Evaluation Taxonomy for Pedagogical Ability Assessment of LLM-Powered AI Tutors](#). *Preprint*, arXiv:2412.09416.
- Margaret Schwan Smith and Mary Kay Stein. 2011. *5 Practices for Orchestrating Productive Mathematics Discussions*. National Council of Teachers of Mathematics, Reston, VA.
- 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.
- LearnLM Team, Abhinit Modi, Aditya Srikanth Veerubhotla, Aliya Rysbek, Andrea Huber, Brett Wiltshire,

Brian Veprek, Daniel Gillick, Daniel Kasenberg, Derek Ahmed, Irina Jurenka, James Cohan, Jennifer She, Julia Wilkowski, Kaiz Alarakyia, Kevin R. McKee, Lisa Wang, Markus Kunesch, Mike Schaekermann, and 27 others. 2024. [Learnlm: Improving Gemini for Learning](#). *Preprint*, arXiv:2412.16429.

Patricia Tunstall and Caroline Gipps. 1996. Teacher Feedback to Young Children in Formative Assessment: a typology. *British Educational Research Journal*, 22(4):389–404.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. [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.

A Full list Pedagogical Features

Table 2

Scope	Criterion	Question
Transversal	Psicosocial	Does the tutor's response focus on the specific task/process rather than the student personally?
Transversal	Psicosocial	Does the tutor's response frame mistakes as learning opportunities?
Transversal	Psicosocial	Does the tutor's response begin by affirming any partial success, even if minor?
Transversal	Metacognition	Throughout the conversation history and final response, does the tutor show preference for asking, rather than stating, to the student what their error could have been and/or how to fix it?
Local	Achieved	Does the tutor's response express that the student has taken some steps correctly?
Local	Achieved	Is the tutor's final response specific about which portion of the student's messages are going in the right direction to solve the proposed problem?
Local	Achieved	Is the tutor's final response correct about which portion of the student's messages are going in the right direction to solve the proposed problem?
Local	Mistaken	Does the tutor's final response imply that the student has made a mistake of some sort?
Local	Mistaken	Is the tutor's final response fully accurate in pointing out the student's mistake(s)?
Local	Mistaken	When communicating that the student has made a mistake, is the tutor's final response specific with regards to what the alleged error was?
Local	Mistaken	Does the tutor's final response provide an explanation for why the student's approach was incorrect?
Local	Mistaken	Regarding the tutor's explanation for why the student's approach was incorrect, is it clear and understandable at a 6th grade level?
Local	Mistaken	Regarding the tutor's explanation for why the student's approach was incorrect, is it fully accurate?
Local	Remediate	Does the tutor offer the student a strategy or hint to solve the word problem?
Local	Remediate	Does the tutor offer the student a correct strategy or hint that would allow them to successfully solve the word problem?
Local	Remediate	Does the tutor offer the student a strategy to solve the word problem that is clear and understandable at the 6th-grade level?
Local	Remediate	Does the tutor offer the student an example problem or fact to correct a misinterpretation of the original problem?

B Feature Extraction Prompt

""You will be presented with the conversation history from a grade-school math tutoring session happening over computer chat, where the student makes a mistake or evidences confusion.

Your task is to evaluate the tutor's final response in terms of the question: {question}

{conversation_history}

Tutor Response: {tutor_response}

Question: {question} (0 for No, 1 for Yes)

Answer: ""

C Best-Performing Classifier Configuration

```
model_config = {
  'input_features': [
    'Throughout the conversation history and final response, does the tutor show preference for asking, rather than stating, to the student what their error could have been and/or how to fix it?',
    'Does the tutor\'s response express that the student has taken some steps correctly?',
    'Is the tutor\'s final response specific about which portion of the student's messages are going in the right direction to solve the proposed problem?',
    'Is the tutor\'s final response correct about which portion of the student's messages are going in the right direction to solve the proposed problem?',
    'Does the tutor\'s final response imply that the student has made a mistake of some sort?',
    'Is the tutor\'s final response fully accurate in pointing out the student\'s mistake(s)?',
    'When communicating that the student has made a mistake, is the tutor\'s final response specific with regards to what the alleged error was?',
    'Does the tutor\'s final response provide an explanation for why the student\'s approach was incorrect?',
    'Regarding the tutor\'s explanation for why the student\'s approach was incorrect, is it clear and understandable at a 6th grade level?',
    'Regarding the tutor\'s explanation for why the student\'s approach was incorrect, is it fully accurate?',
    'Does the tutor offer the student a strategy or hint to solve the word problem?',
    'Does the tutor offer the student an example problem or fact to correct a misinterpretation of the original problem?',
  ]
  'preprocessing': {
    'oversampling': 'SMOTE'
  },
  'rf_hyperparameters': {
    'max_depth': None,
    'max_features': 'sqrt',
    'min_samples_leaf': 4,
    'n_estimators': 500
  }
}
```


MSA at BEA 2025 Shared Task: Disagreement-Aware Instruction Tuning for Multi-Dimensional Evaluation of LLMs as Math Tutors*

Baraa Hikal, Mohamed Basem, Islam Oshallah, Ali Hamdi

Faculty of Computer Science, MSA University, Egypt

{baraa.moaweya, mohamed.basem1, islam.abdulhakeem, ahamdi}@msa.edu.eg

Abstract

We present MSA-MATHEVAL, our submission to the BEA 2025 Shared Task on evaluating AI tutor responses across four instructional dimensions: Mistake Identification, Mistake Location, Providing Guidance, and Actionability. Our approach uses a unified training pipeline to fine-tune a single instruction-tuned language model across all tracks, without any task-specific architecture modifications. To improve prediction reliability, we introduce a disagreement-aware ensemble inference strategy that enhances coverage of minority labels. Our system achieves strong performance across all tracks, ranking 1st in Providing Guidance, 3rd in Actionability, and 4th in both Mistake Identification and Mistake Location. These results demonstrate the effectiveness of scalable instruction tuning and disagreement-driven modeling for robust, multi-dimensional evaluation of LLMs as educational tutors.

1 Introduction

Large language models (LLMs) are increasingly used in educational applications, acting as AI tutors that engage students in natural language. However, effective tutoring goes beyond producing correct answers. AI tutors must recognize student mistakes, explain misconceptions, provide constructive guidance, and suggest actionable next steps. Assessing such teaching behavior remains challenging.

Prior work in intelligent tutoring systems (ITS) emphasized these goals long before the advent of LLMs. For example, AutoTutor used natural language processing (NLP) and dialogue-based feedback to improve learning outcomes across domains (Nye et al., 2014). Later, metrics such as *conversational uptake* were proposed to capture tutor responsiveness and its link to instructional quality (Demszky et al., 2021).

With the rise of instruction-tuned LLMs, new frameworks have emerged to assess their teaching abilities. Tack and Piech (Tack and Piech, 2022) introduced the AI Teacher Test for evaluating model helpfulness and student understanding, while later work proposed finer rubrics such as coherence, correctness, targetedness, and actionability (Macina et al., 2023; Daheim et al., 2024; Wang et al., 2024).

Building on these efforts, the BEA 2025 Shared Task adopts *MRBench*—a pedagogically motivated benchmark introduced by Maurya et al. (2025)—to evaluate AI-generated tutor responses in math dialogues (Kochmar et al., 2025). While BEA 2023 emphasized response generation, BEA 2025 shifts toward assessing feedback quality across four instructional dimensions derived from educational science.

In this work, we present MSA-MATHEVAL, a unified system that addresses all four tracks using a single fine-tuned model and consistent training pipeline. We fine-tune the open-weight *Mathstral-7B-v0.1*—an instruction-tuned LLM specialized for mathematical reasoning—using parameter-efficient LoRA adapters. To improve prediction reliability, we incorporate ensemble-based inference that combines model disagreement and uncertainty estimation.

Our contributions are as follows:

- We design a unified training pipeline for all four BEA 2025 tracks, using LoRA-based fine-tuning of *Mathstral-7B-v0.1* with no track-specific architecture changes.
- We propose an ensemble-based inference strategy leveraging model disagreement and uncertainty for robust prediction.
- We achieve top-tier performance across all tracks, including first place in Providing Guidance.

* <https://github.com/baraahekal/BEA-2025>

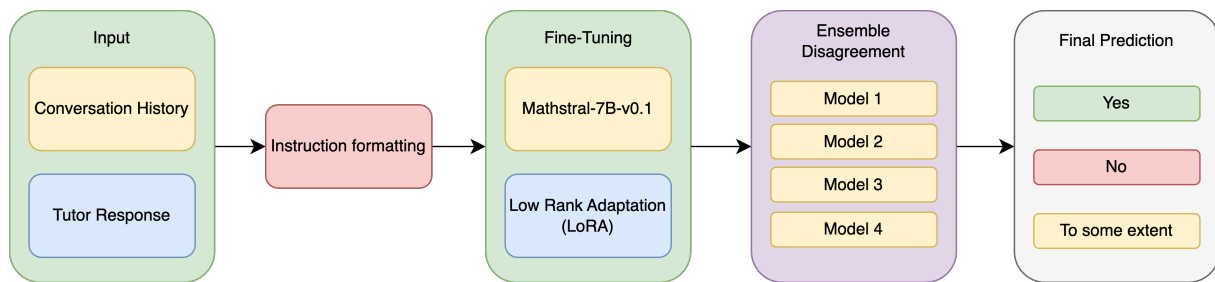


Figure 1: Overview of our unified MSA-MATHEVAL framework for the BEA 2025 Shared Task. The pipeline includes preprocessing, LoRA-based fine-tuning of Mathstral-7B-v0.1, and disagreement-aware ensemble inference.

2 Related Work

Evaluating the pedagogical capabilities of AI tutors builds upon long-standing research in intelligent tutoring systems (ITS) and more recent advances in large language models (LLMs). Early ITS such as AutoTutor emphasized the importance of natural language dialogue in promoting student learning through error remediation and scaffolding (Nye et al., 2014). These systems often relied on rule-based or statistical NLP methods to assess learner inputs and generate appropriate tutor responses.

The emergence of instruction-tuned LLMs has prompted a shift toward more scalable methods for modeling tutoring behavior. Tack and Piech (2022) proposed the AI Teacher Test to benchmark LLM outputs on criteria such as helpfulness and pedagogical appropriateness. Macina et al. (2023) and Daheim et al. (2024) introduced fine-grained rubrics for LLM tutoring quality in mathematical dialogue, including dimensions such as targetedness, coherence, and actionability.

In terms of modeling strategies, prior systems have explored both classification and ranking approaches for feedback generation. Daheim et al. (2024) used multi-aspect annotation schemes to evaluate feedback informativeness, while Wang et al. (2024) proposed a bridging rubric for LLM feedback grounded in human tutor behavior. These studies highlight the need for systems that go beyond correctness to capture richer instructional attributes.

Compared to these approaches, our work introduces a unified training and inference framework across multiple feedback dimensions, leveraging ensemble disagreement and uncertainty estimation for prediction stability. Unlike previous models with track-specific adaptations or rule-based post-processing, we apply a consistent architecture based on the Mathstral-7B-v0.1 model across all

tasks. This allows us to assess the generalizability of instruction-tuned LLMs for the mathematics domain across key dimensions of pedagogical ability.

3 Method

Our approach, MSA-MATHEVAL, applies a unified framework across all four tracks in the BEA 2025 Shared Task. We build on the instruction-tuned Mathstral-7B-v0.1 model and leverage parameter-efficient fine-tuning (LoRA) along with ensemble-based inference to enhance prediction robustness. The methodology consists of the following stages: dataset preprocessing, model selection, fine-tuning strategy, and ensemble-based inference (see Figure 1).

3.1 Preprocessing

The original dataset consists of nested JSON files, where each dialogue contains multiple tutor responses annotated across four pedagogical dimensions. To facilitate instruction-based fine-tuning, we transformed the data into four track-specific JSONL files. Each file includes a flattened dialogue, a natural language evaluation prompt, and a categorical label from three possible options: *Yes*, *To some extent*, or *No*.

Each training instance was structured as a two-turn dialogue following the chat schema used by instruction-tuned language models. Specifically:

- **user:** This field contains a complete, track-specific prompt with explicit evaluation criteria, followed by the dialogue context and tutor response to be evaluated.
- **assistant:** This field contains the gold label corresponding to the tutor response—one of "Yes", "To some extent", or "No"—as annotated in the development set.

The system role was omitted to reduce token overhead and focus the model on the input–output mapping relevant to each multi-class classification task.

Track 1 – Mistake Identification

TASK DEFINITION:

You are an expert evaluator of AI tutor responses. Your task is to determine whether the tutor’s response accurately identifies a mistake in the student’s reasoning or solution.

EVALUATION CRITERIA:

1. “Yes”– The tutor accurately identifies a mistake in the student’s response.
2. “To some extent”– The tutor shows some awareness, but it is ambiguous or uncertain.
3. “No”– The tutor fails to identify or misunderstands the mistake.

Track 2 – Mistake Location

TASK DEFINITION:

You are an expert evaluator of AI tutor responses. Your task is to determine whether the tutor’s response accurately points to a genuine mistake and its location in the student’s response.

EVALUATION CRITERIA:

1. “Yes”– The tutor clearly points to the exact location of the mistake.
2. “To some extent”– The tutor refers to a mistake but is vague or indirect.
3. “No”– The tutor provides no indication of where the mistake occurred.

Track 3 – Providing Guidance

TASK DEFINITION:

You are an expert evaluator of AI tutor responses. Your task is to determine whether the tutor’s response provides correct and relevant guidance to help the student.

EVALUATION CRITERIA:

1. “Yes”– The tutor gives helpful guidance such as a hint or explanation.
2. “To some extent”– The guidance is partially helpful, unclear, or incomplete.
3. “No”– The guidance is absent, irrelevant, or factually incorrect.

Track 4 – Actionability

TASK DEFINITION:

You are an expert evaluator of AI tutor responses. Your task is to determine whether the tutor’s feedback is actionable, i.e., it clearly suggests what the student should do next.

EVALUATION CRITERIA:

1. “Yes”– The response includes clear next steps for the student.
2. “To some extent”– Some action is implied, but it is not clearly stated.
3. “No”– No action is suggested or the feedback ends the conversation.

Each JSONL instance includes an instruction (as the user message), an input (composed of the full dialogue context and tutor response), and an output (gold label as assistant). This format enables effective supervised fine-tuning of Mathstral-7B-v0.1 on each dimension-specific classification task.

3.2 Model Selection and Architecture

Our system is built upon the Mathstral-7B-v0.1 language model, an open-source 7B-parameter transformer tailored for mathematical and scientific reasoning (Mistral AI Team, 2024). It is an instruction-tuned variant of the Mistral 7B architecture (Jiang et al., 2023), which itself builds on the transformer framework used in LLaMA (Touvron et al., 2023a,b). Mathstral uses a 32-layer transformer with 4096-dimensional hidden states and 32 attention heads (8 for keys/values), and benefits from Mistral’s sliding-window attention mechanism, enabling long-context comprehension up to 32k tokens. This makes it particularly suitable for modeling multi-turn math tutoring dialogues that require broad context retention.

Mathstral-7B-v0.1 was selected based on its strong performance in math-specific benchmarks and its open-access availability. It was instruction-tuned by Project Numina on mathematical reasoning tasks and achieves high scores on datasets such as GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), and MMLU-STEM (Hendrycks et al., 2021a). For instance, it reports 56.6% accuracy on MATH, significantly outperforming base Mistral and LLaMA models of comparable size.

Compared to alternatives, Mathstral outperforms general-purpose LLaMA 2 (Touvron et al., 2023b) and even surpasses some larger models in mathematical domains. While proprietary models like GPT-3.5 or GPT-4 (OpenAI, 2022, 2023) show impressive general capabilities, their closed nature limits fine-tuning flexibility and deployment cost-effectiveness. Mathstral, by contrast, is released under Apache 2.0, making it fine-tunable with LoRA on modest compute budgets.

We thus chose Mathstral-7B-v0.1 as the backbone of our system due to its optimal trade-off

between math reasoning accuracy, open weight availability, and instruction-following capability.

3.3 Training and Fine-Tuning

We fine-tuned Mathstral-7B-v0.1 separately for each BEA 2025 track using Low-Rank Adaptation (LoRA) (Hu et al., 2021), framing the task as three-way instruction-based classification. Each input was represented as a two-turn dialogue—comprising a prompt and a categorical label—and modeled as a supervised instruction-following task.

To enable efficient adaptation with minimal memory overhead, we used LoRA with a rank of $r = 64$, scaling factor $\alpha = 2.0$, and no dropout. Adapters were injected into the attention query and value projections in each transformer block. The low-rank update to the frozen weight matrix W is defined as:

$$\Delta W = \alpha \cdot AB \quad (1)$$

where $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$ are trainable matrices, and d is the dimension of the attention head. The final effective weight is $W + \Delta W$. Figure 2 illustrates this injection mechanism.

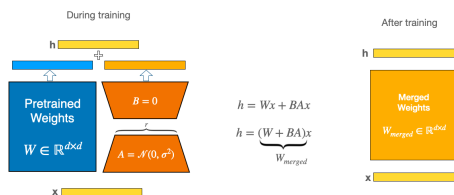


Figure 2: LoRA adaptation adds trainable low-rank matrices A and B to frozen attention weights W_0 , producing an effective weight $W = W_0 + \alpha AB$ during training. Only A and B are updated, enabling memory-efficient fine-tuning (Hu et al., 2021).

Training was capped at 500 steps with gradient norm clipping ($\|g\|_2 < 1.0$) and a maximum sequence length of 2048 tokens. We used a batch size of 2, single micro-batching, and fixed seed 42 for reproducibility. Optimization was performed using AdamW with a learning rate of 4×10^{-5} , 10% linear warmup, and weight decay of 0.05.

We evaluated model performance every 50 steps on a held-out validation set, which consisted of the last 30% of the development dataset. The development set includes 300 dialogues sourced from the MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024) datasets. Checkpoints were saved every 100 steps with a retention window of the three most

recent. Only LoRA adapter weights were saved to minimize disk usage and enable efficient inference. All training runs were conducted in a single-node setup with `world_size=1`.

This training configuration ensured stable convergence on limited supervision, while maintaining computational efficiency and reproducibility across all four pedagogical dimensions.

3.4 Inference and Ensemble Strategy

To enhance robustness and maintain generalization across all four tracks, we employed an ensemble-based inference strategy grounded in model disagreement. Rather than aggregating predictions through majority voting, we fine-tuned five independent models per track. Each model used the same base architecture Mathstral-7B-v0.1 but was trained with different random seeds and shuffled data splits to encourage diversity in learned representations. This disagreement-aware mechanism allows us to capture uncertainty and preserve minority-class predictions, especially for ambiguous cases labeled "To some extent".

Each model in the ensemble predicts a class independently using greedy decoding. During inference, we collect all five predictions for a given sample and apply a filtering policy: if the predictions exhibit full agreement, the class is retained. If the ensemble disagrees, we analyze the class distribution and prefer predictions that preserve the relative frequency of "To some extent" observed in the development set. This is crucial because "Yes" labels are dominant in both the training and validation sets, potentially leading to biased predictions under a naïve voting scheme.

Our design choice is motivated by the use of macro-F1 as the primary evaluation metric in the BEA 2025 Shared Task. Unlike accuracy or micro-F1, macro-F1 gives equal weight to all classes, making performance on minority labels such as "To some extent" especially important. By encouraging the retention of these less frequent but pedagogically relevant labels through disagreement-aware filtering, we improve per-class recall and stabilize final predictions.

This ensemble strategy is lightweight in deployment, as only LoRA adapter weights are loaded during inference. Predictions are generated sequentially and combined via a deterministic post-processing script that requires no additional training.

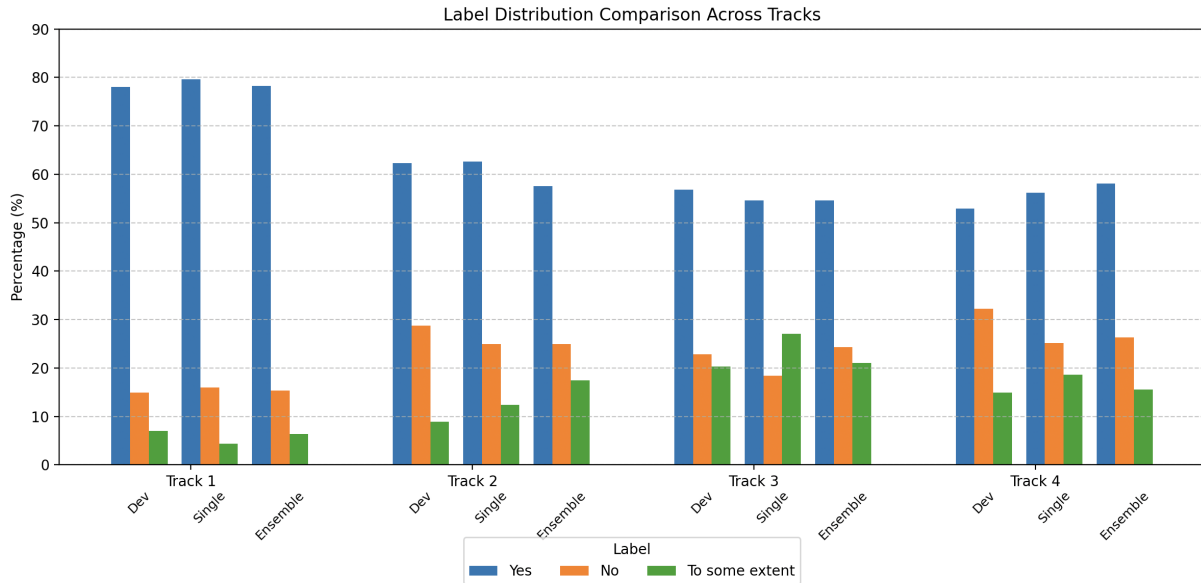


Figure 3: Label distribution comparison across four evaluation tracks. Each group of bars represents the percentage of predictions for the labels "Yes", "No", and "To some extent" for three settings: the MRBench development set (Dev), the best-performing single model on the test set (Single), and the ensemble system on the same test set (Ensemble).

4 Experiments

4.1 Dataset

The BEA 2025 Shared Task provides a benchmark for evaluating AI tutor responses across four pedagogically motivated tracks: Mistake Identification, Mistake Location, Providing Guidance, and Actionability (Kochmar et al., 2025). The dataset is based on *MRBench*, a curated collection of math-focused educational dialogues designed for evaluating feedback quality in instructional settings (Maurya et al., 2025). It includes dialogues drawn from two publicly available sources: MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024).

Each instance comprises a multi-turn conversation between a student and an AI tutor, a final student question or statement, and multiple candidate tutor responses. The task is to classify each response along the four instructional dimensions, using a three-way labeling scheme: *Yes*, *To some extent*, and *No*.

The shared task organizers provide a labeled development set with expert annotations for training and validation. The test set is blind—its labels are hidden from participants and used by the organizers to evaluate final system submissions. This setup ensures fair comparison and simulates real-world deployment where labeled data may be limited or unavailable.

MRBench Statistics:

- **192** annotated dialogues in total: **60** from Bridge and **132** from MathDial.
- **1,596** total tutor responses annotated across 7 LLMs and multiple human tutors (expert and novice).
- Each response is annotated with 8 evaluation dimensions; the shared task focuses on 4 core tracks.
- **Dialogue Length:** Bridge dialogues average 4 turns and 140 characters. MathDial averages 5.5 turns and 906 characters.

4.2 Evaluation

To evaluate the pedagogical quality of model predictions across all four tracks, the BEA 2025 Shared Task employs two complementary scoring protocols: *exact evaluation* and *lenient evaluation*. Both use macro-averaged F1 score and accuracy as core metrics.

Exact Evaluation. In the primary setting, each prediction is evaluated against a gold label using a three-way classification scheme: "Yes", "To some extent", and "No". Let C denote the set of all classes, and $F1_c$ the F1 score for class $c \in C$. The macro-F1 score is computed as the unweighted

Track	Run	Strict F1	Lenient F1	Strict Acc.	Lenient Acc.	Main Metric Rank
Mistake Identification	Run 1	71.54%	91.52%	87.59%	95.35%	4 th / 44
	Run 2	70.66%	91.42%	87.98%	95.22%	
	Run 3	56.78%	82.95%	83.65%	91.92%	
	Run 4	67.88%	90.13%	87.20%	94.76%	
	Run 5	71.34%	91.52%	87.39%	95.35%	
Mistake Location	Run 1	55.62%	77.79%	72.01%	80.93%	4 th / 31
	Run 2	56.02%	77.73%	72.01%	81.19%	
	Run 3	56.88%	78.48%	71.88%	82.09%	
	Run 4	52.79%	73.65%	63.61%	78.22%	
	Run 5	57.43%	78.48%	69.75%	82.09%	
Providing Guidance	Run 1	55.28%	76.02%	67.29%	80.35%	1 st / 35
	Run 2	53.76%	76.59%	65.09%	80.74%	
	Run 3	56.65%	74.75%	63.61%	80.61%	
	Run 4	58.33%	77.98%	66.13%	81.90%	
Actionability	Run 1	51.35%	68.81%	58.31%	76.60%	3 rd / 29
	Run 2	66.99%	84.97%	71.95%	87.91%	
	Run 3	65.90%	84.45%	71.82%	87.07%	
	Run 4	69.84%	86.59%	75.37%	89.08%	
	Run 5	65.90%	84.45%	71.82%	87.07%	

Table 1: Strict and lenient macro-F1 and accuracy across five runs per track. Bolded scores indicate per-track bests. Final column shows BEA 2025 leaderboard rank based on strict macro-F1 (main metric).

average across all classes:

$$\text{Macro-F1} = \frac{1}{|C|} \sum_{c \in C} \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (2)$$

This metric penalizes class imbalance and rewards systems that maintain recall across minority classes such as "To some extent".

Lenient Evaluation. To account for pedagogical similarity between "Yes" and "To some extent", the task also includes a two-way lenient evaluation protocol. Labels "Yes" and "To some extent" are merged into a single positive class, resulting in a binary classification task. The same macro-F1 and accuracy metrics are then applied to the collapsed label set.

Accuracy. For both settings, accuracy is defined as the proportion of correct predictions over all samples:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(\hat{y}_i = y_i) \quad (3)$$

where N is the number of samples, \hat{y}_i is the predicted label, and y_i is the gold label for instance i .

Protocol. Since the test labels were not released, we computed local metrics only on the development set. All official test results were obtained through the shared task evaluation server. Model

selection and early stopping were based on development macro-F1 under the exact evaluation setting, which served as the primary leaderboard metric.

4.3 Effect of Ensemble Disagreement on Label Distribution

To analyze the effect of our ensemble strategy on class balance, we examined the label distributions across all four tracks. The development set consistently exhibited a dominant proportion of "Yes" labels—often exceeding 55%—with "To some extent" and "No" underrepresented.

Left uncorrected, single-model predictions tended to reinforce this imbalance, frequently collapsing uncertain cases into the majority class. To mitigate this, our ensemble disagreement filtering selectively retained predictions for the minority class "To some extent" when model consensus was low. This design choice was informed by the use of macro-F1 as the shared task’s official ranking metric, which rewards balanced performance across all classes.

Figure 3 compares label distributions from the development set, single-model outputs, and ensemble predictions. The ensemble strategy improves minority-class coverage—especially for "To some extent"—by better matching the development distribution and mitigating dominant-class bias. This adjustment is particularly useful in ambiguous cases where subtle feedback is warranted.

Track	Strict Macro-F1	Lenient Macro-F1	Strict Acc.	Lenient Acc.
Mistake Identification	4 th / 44	2 nd / 44	1 st / 44	2 nd / 44
Mistake Location	4 th / 31	6 th / 31	10 th / 31	6 th / 31
Providing Guidance	1 st / 35	2 nd / 35	3 rd / 35	3 rd / 35
Actionability	3 rd / 29	1 st / 29	2 nd / 29	2 nd / 29

Table 2: Per-metric leaderboard ranks (out of all teams) for each track.

5 Results

We evaluate our system across the four BEA 2025 tracks—Mistake Identification, Mistake Location, Providing Guidance, and Actionability—using both exact (three-class) and lenient (binary) evaluation protocols, as outlined in Section 4.2. We report macro-averaged F1 and accuracy scores across five independent runs for each track and compare our best results to the official leaderboard.

5.1 Performance Across Runs

Table 1 presents detailed performance scores from five independent fine-tuning runs per track. Each run was evaluated on strict and lenient macro-F1 as well as accuracy. We observe moderate variance across runs, particularly in Tracks 2 and 4, which feature more ambiguous tutor responses.

Our best-performing models achieved:

- **Track 1:** 71.54% strict macro-F1 and 91.52% lenient macro-F1 (Run 1).
- **Track 2:** 57.43% strict macro-F1 and 78.48% lenient macro-F1 (Run 5).
- **Track 3:** 58.33% strict macro-F1 and 77.98% lenient macro-F1 (Run 4).
- **Track 4:** 69.84% strict macro-F1 and 86.59% lenient macro-F1 (Run 4).

These results highlight the robustness of our unified training pipeline and the positive impact of ensemble disagreement filtering on minority-class prediction, especially in borderline cases.

5.2 Leaderboard Rankings

Table 2 summarizes our official rankings among all participating teams. We consistently placed within the top 5 across all tracks and metrics, securing the 1st rank in Track 3 (Providing Guidance) and top-3 ranks in three other metrics.

These ranks validate the effectiveness of our approach across varied pedagogical feedback dimensions. Notably, our system generalizes well

across tasks using a unified model and minimal task-specific engineering.

6 Limitations

Despite its strong performance across BEA 2025 tracks, our approach has several limitations.

First, the specialization of `Mathstral-7B-v0.1` to mathematical reasoning may hinder generalization to non-mathematical domains. While domain-specific instruction tuning improves in-domain performance, prior work has shown that such specialization can cause *catastrophic forgetting* of general knowledge, even with parameter-efficient methods like LoRA (Dettmers et al., 2023). Moreover, although LoRA significantly reduces memory and compute costs, its low-rank decomposition can constrain the model’s expressiveness in capturing nuanced pedagogical feedback (Xu et al., 2023; Zhou et al., 2023).

Second, our ensemble disagreement strategy introduces additional inference cost. While it improves recall for minority labels such as “To some extent”, the benefit may diminish if the base models exhibit correlated predictions. Prior work shows that ensembles are most effective when model predictions are diverse and independent (Lakshminarayanan et al., 2017), which may not always hold in practice.

Finally, the reliance on macro-averaged F_1 as the primary evaluation metric, although fair for class imbalance, lacks granularity in penalizing pedagogically critical mistakes. For example, misclassifying a completely wrong tutor response as “To some extent” is penalized equally to a more plausible confusion between “Yes” and “To some extent”. While the lenient evaluation partially addresses this by collapsing similar labels, it does not fully capture the instructional severity of errors (Kochmar et al., 2025).

7 Conclusion

We presented MSA-MATHEVAL, a unified framework for evaluating AI tutor responses across

four pedagogical dimensions in the BEA 2025 Shared Task. By fine-tuning a math-specialized LLM (Mathstral-7B-v0.1) using LoRA and leveraging ensemble disagreement during inference, our system achieved top-tier results across all tracks—ranking 1st in Providing Guidance and within the top 5 in all others. Our findings highlight the effectiveness of combining domain-specific instruction tuning with disagreement-aware prediction filtering for educational feedback assessment. Future work will explore cross-domain generalization and dynamic calibration strategies to further enhance robustness.

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*.
- Nico Daheim, Jakob 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, 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.
- 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.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. [Measuring massive multitask language understanding](#). In *International Conference on Learning Representations (ICLR)*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. [Measuring mathematical problem solving with the MATH dataset](#). In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Weizhu Wang, and Zichao Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *arXiv preprint arXiv:2106.09685*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard-Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.
- Ekaterina Kochmar, Kaushal Kumar Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Ana  s 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*.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, pages 6402–6413.
- 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.
- 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.
- Mistral AI Team. 2024. Mathstral 7b v0.1: A math reasoning and scientific discovery model. <https://mistral.ai/news/mathstral>.
- 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.
- OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. <https://openai.com/blog/chatgpt>.
- OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Ana  s Tack and Chris Piech. 2022. [The AI teacher test: Measuring the pedagogical ability of blender and GPT-3 in educational dialogues](#). In *Proceedings of the 15th International Conference on Educational*

Data Mining, pages 522–529, Durham, United Kingdom. International Educational Data Mining Society.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. [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*, pages 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. 2023. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv preprint arXiv:2303.07338*.

TutorMind at BEA 2025 Shared Task: Leveraging Fine-Tuned LLMs and Data Augmentation for Mistake Identification

Fatima Dekmak

American University of Beirut
Beirut, Lebanon
fkd04@mail.aub.edu

Christian Khairallah

Aralects
Abu Dhabi, United Arab Emirates
christiank@aralects.com

Wissam Antoun

INRIA
France
wissam.antoun@gmail.com

Abstract

In light of the growing adoption of large language models (LLMs) as educational tutors, it is crucial to effectively evaluate their pedagogical capabilities across multiple dimensions. Toward this goal, we address the Mistake Identification sub-task of the BEA 2025 Shared task, aiming to assess the accuracy of tutors in detecting and identifying student errors. We experiment with several LLMs, including GPT-4o-mini, Mistral-7B, and Llama-3.1-8B, evaluating them in both zero-shot and fine-tuned settings. To address class imbalance, we augment the training data with synthetic examples, targeting underrepresented labels, generated by Command R+. Our GPT-4o model fine-tuned on the full development set achieves a strict macro-averaged F1 score of 71.63%, ranking second in the shared task. Our work highlights the effectiveness of fine-tuning on task-specific data and suggests that targeted data augmentation can further support LLM performance on nuanced pedagogical evaluation tasks.

1 Introduction

The increasing integration of large language models into educational applications has sparked significant interest in their potential as AI tutors capable of engaging students in meaningful learning dialogues. A critical component of effective tutoring lies in the ability to identify and address student misconceptions or errors. While recent studies have explored the capabilities of LLMs in simulating tutor-like behaviors, there remains a pressing need for systematic frameworks to evaluate their pedagogical effectiveness.

The BEA 2025 Shared Task (Kochmar et al., 2025) introduced a structured evaluation of AI tutors' responses, focusing on four pedagogical di-

mensions: mistake identification, mistake location, providing guidance, and actionability. In this work, we focus on the Mistake Identification sub-task, which involves determining whether a tutor's response acknowledges a student's error within a given conversational context. The task builds upon the unified evaluation taxonomy proposed by (Maurya et al., 2025), which defines key pedagogical dimensions for assessing the effectiveness of AI tutors in mistake remediation scenarios.

In our participation in this task, under the team name TutorMind, we explore the effectiveness of multiple LLMs, including GPT-4o-mini (OpenAI, 2024), Mistral-7B (Mistral-AI, 2023), and Llama-3.1-8B (Meta, 2024), in both zero-shot and fine-tuned settings. To address class imbalance in the dataset, we introduce a data augmentation strategy using the Command-R-plus model (Cohere, 2024) to generate synthetic examples targeting underrepresented classes. Our best-performing model, a fine-tuned variant of GPT-4o-mini trained on the full development dataset, achieved a strict macro-averaged F1 score of 71.63%, ranking second place in the competition.

This study contributes to the growing body of research on AI-assisted education by demonstrating how targeted fine-tuning can enhance LLMs' ability to evaluate the pedagogy of tutor LLMs. Our findings underscore the importance of aligning model training with domain-specific evaluation criteria. All fine-tuning scripts, evaluation pipelines, and data augmentation prompts, are publicly available for reproducibility and further research.¹

¹<https://github.com/fatimadekmak/TutorMind-BEA2025>

2 Related Work

LLM-Powered AI Tutors in Education: Large language models are being increasingly used as AI tutors capable of engaging students in natural dialogue and providing real-time feedback (Wang et al., 2024). In particular, domains like mathematics and programming have seen significant interest due to the structured nature of problems and the importance of identifying student misconceptions early (Daheim et al., 2024).

However, while LLMs demonstrate impressive fluency and general question-answering capabilities, their effectiveness as pedagogical models remains limited. For instance, GPT-4 often reveals answers prematurely, undermining its role as a supportive tutor. Similarly, Gemini and Phi3 struggle with coherence and actionable guidance, highlighting the need for targeted evaluation frameworks that go beyond traditional natural language generation (NLG) metrics (Jurenka et al., 2024).

Tutor LLMs Evaluation Frameworks: Traditional NLG metrics such as BLEU, ROUGE, and BERTScore are insufficient for evaluating AI tutors because they do not account for pedagogical values such as mistake identification, scaffolding, or encouraging tone. Several studies have proposed domain-specific evaluation criteria tailored to educational dialogues.

(Tack and Piech, 2022) introduced a framework assessing AI tutors based on conversational uptake, understanding, and helpfulness. (Wang et al., 2024) extended this by incorporating dimensions such as care, human-likeness, and usefulness. (Daheim et al., 2024) focused on actionability and correctness.

In contrast, (Maurya et al., 2025) proposed a unified taxonomy comprising eight distinct pedagogical dimensions: Mistake Identification, Mistake Location, Revealing of the Answer, Providing Guidance, Actionability, Coherence, Tutor Tone, Human-likeness, The authors also released MR-Bench, a benchmark dataset containing annotated responses from both human and LLM-based tutors, which is a previous version of the dataset being used in the current task.

Use of LLMs as Evaluators: Researchers have explored the use of LLMs themselves as critics or evaluators. Several studies have demonstrated that LLMs like GPT-4 can assess the quality of educational dialogues with moderate agreement compared to human annotators (Koutchme et al.,

2024). In particular, GPT-4 has been used as an automatic judge to evaluate feedback quality in programming education, showing reasonable correlation with expert human evaluations, although it tends to be overly optimistic in its ratings.

Other studies have leveraged LLMs to score classroom instruction or provide actionable insights for teacher coaching (Wang and Demszky, 2023). These works suggest that LLMs can offer scalable and cost-effective evaluation solutions, although they are not yet fully reliable substitutes for human judgment.

Recent efforts underscore both the growing interest in deploying LLMs as AI tutors and the challenges involved in evaluating their pedagogical effectiveness. While LLMs are proficient at generating fluent and coherent responses, their ability to function as effective tutor agents remains limited. Building on the work of (Maurya et al., 2025), we focus on a single pedagogical dimension, mistake identification, and investigate how fine-tuning LLMs can enhance their ability to evaluate tutor responses within this context.

3 Methodology

This section describes the models, dataset preparation, training setup, and augmentation strategy used to address the Mistake Identification sub-task of the BEA 2025 Shared Task.

3.1 Task Setup & Dataset

We utilized the labeled development set provided by the shared task organizers, focusing specifically on the Mistake Identification dimension of AI tutor responses. The dataset contains three class labels indicating whether the tutor’s response addressed a student mistake: Yes (1932 instances), No (370), and To some extent (174). This distribution presents a significant class imbalance, with the "Yes" class significantly overrepresented compared to the other two categories (see Appendix A for a breakdown). We observed that this imbalance negatively impacted model performance during initial experiments. This motivated us to implement targeted data augmentation strategies, as discussed in Section 3.4.

To evaluate model behavior under constrained supervision, we partitioned the development set into two subsets using stratified sampling: a **Training Subset** (80%) and an **Validation Subset** (20%). All initial zero-shot and fine-tuning experiments

were conducted using the training subset, while the validation subset served as a held-out test set to guide model selection.

Additionally, all final system submissions were evaluated by the organizers on a separate **Blind Test Set**, for which ground-truth labels were not released. This Blind Test Set was used to compute the official leaderboard scores for the shared task.

3.2 Model Selection

We evaluated the use of multiple large language models as tutor evaluators. GPT-4o-mini (OpenAI, 2024) was chosen for its strong performance and availability for fine-tuning. Mistral-7B (Mistral-AI, 2023) and LLaMA-3.1-8B (Meta, 2024) instruct models were selected as competitive open-source baselines. Larger models were excluded from this study due to computational constraints.

3.3 Fine-tuning Setups

Fine-tuning experiments on the Mistral-7B and LLaMA-3.1-8B models were carried out using the Unsloth framework, which enables optimized and efficient training through 4-bit quantization and the integration of LoRA adapters. Both models were trained for a total of three epochs with a learning rate of $2e-4$ and the AdamW optimizer. The training process was conducted on Google Colab², leveraging the range of available GPU resources, including A100 and T4 GPUs with high memory capacity, to ensure stable and efficient execution.

GPT-4o-mini, in contrast, was fine-tuned via the OpenAI platform³ using supervised fine-tuning (SFT). The training data was formatted into JSONL files with role-tagged messages and associated classification targets (Yes/No/To Some Extent), following OpenAI’s SFT guidelines. Prompt templates and formatting details for all models are provided in appendix C and D.

3.4 Data Augmentation

After initially fine-tuning our selected models on the training subset, we observed a noticeable discrepancy between strict and lenient scores (see Section 5 for further discussion). The models frequently confused the No and To some extent classes with Yes, indicating that class imbalance was a limiting factor. This motivated a data augmentation step focused on these underrepresented classes. We generated additional training examples for the No

and To some extent classes using the Command R+ model (Cohere, 2024). This model was selected because it was neither involved in generating the original tutor responses nor used in the evaluation pipeline, and was capable of producing high-quality tutor response that follow the given instruction.

We created 100 synthetic examples per underrepresented class. Each instance was manually reviewed for label correctness and consistency with the shared task’s annotation guidelines. These examples were added to the training subset and used in a second round of fine-tuning. We refer to this expanded dataset as the **augmented training subset** throughout the paper.

During manual inspection, most generated responses appeared to match the intended labels. The “To some extent” examples typically followed the prompt instructions, using cautious or indirect language like “maybe,” “I think,” or “let’s double-check”, without clearly identifying a mistake. For the “No” class, most responses were affirming and feedback-neutral, as expected. However, some responses included subtle hints that could be interpreted as uncertainty, making them closer in tone to the “To some extent” label. These cases were not filtered out as we prioritized maintaining class coverage. In retrospect, these borderline cases introduced some mild label noise, which highlights the need for more precise quality control in future augmentation steps.

The original and augmented training setups shared identical hyperparameter settings. The prompt used with Command R+ to generate data is documented in appendix E.

4 Results

We report results on both the held-out dev test set and the official shared task test set. Table 1 summarizes the accuracy and macro F1 scores under both strict and lenient settings. Our discussion focuses on strict F1, which was the official evaluation metric.

Zero-shot results show that both GPT-4o and Mistral-7B performed reasonably well out of the box (strict F1: 52.13% and 51.73% respectively), while LLaMA-3.1-8B struggled in the absence of fine-tuning, scoring only 19.03%. These results highlight the limitations of zero-shot prompting, particularly for minority class detection.

Fine-tuning on the initial training subset signif-

²<https://colab.google/>

³<https://platform.openai.com/docs/overview>

icantly improved performance across all models. GPT-4o achieved 68.20% strict F1 on the dev test set and was submitted as our first system, scoring 67.70% on the official blind test set. Mistral-7B and LLaMA-3.1-8B achieved 62.61% and 41.52%, respectively. Based on these results, we selected GPT-4o for further fine-tuning on the full development set. GPT-4o fine-tuned on the full development set scored 71.63% on the blind test, ranking second in the competition.

To evaluate the impact of data augmentation, we fine-tuned both GPT-4o-mini and Mistral-7B on the augmented training subset, which included synthetic examples targeting the underrepresented “No” and “To some extent” classes. LLaMA-3.1-8B was excluded from this stage, as it consistently underperformed compared to the other models in earlier experiments. Both GPT-4o-mini and Mistral-7B showed further gains: GPT-4o-mini reached 70.34% strict F1 on the dev test set, while Mistral-7B improved from 62.61% to 70.08%. These configurations were submitted as additional runs, with GPT-4o-mini achieving 70.76% on the blind test set. Notably, the augmented GPT-4o-mini model outperformed all other models trained only on the training subset. However, it was never fine-tuned on the full development set due to time constraints. As a result, it was not submitted in its optimal form. We hypothesize that combining data augmentation with full-devset fine-tuning would have yielded even stronger results, potentially surpassing our best-performing submission (GPT-4o-mini fine-tuned on the full devset without augmentation), which scored 71.63% on the blind test. The relatively lower leaderboard score of the augmented GPT-4o-mini model reflects the limitation of training on a smaller portion of the data, rather than a shortcoming of the augmentation strategy itself. The complete comparison is presented in Table 1.

5 Analysis

The Mistake Identification task was evaluated under two settings: strict and lenient. In the strict setting, the model deals with the three classes, Yes, No, or To some extent, separately. On the other hand, the lenient setting merges the Yes and To some extent labels into a single positive class. This reduces the penalty for confusing pedagogical distinctions, specifically partial vs. full mistake recognition.

As shown in Table 2, lenient scores were consistently higher than strict scores across all models and configurations. For instance, our GPT-4o-mini model fine-tuned on the training subset achieved a strict F1 of 68.20% but a lenient F1 of 87.53%, suggesting that the model often detected the presence of a mistake but occasionally failed to clearly distinguish between full and partial mistake identification. Similarly, Mistral-7B’s results reinforce this observation, with 62.61% strict F1 and 85.71% lenient F1.

These results, along with careful examination of model predictions, had two key implications during system development. First, they highlighted that model failures were frequently due to confusion between Yes and To some extent, rather than between positive and negative classes (Yes/To some extent vs. No). This informed our decision to generate targeted augmentations specifically for the No and To some extent classes, which were both underrepresented and prone to misclassification. Second, the wide gap between strict and lenient scores helped us judge whether model improvements were actually sharpening pedagogical judgment, or simply boosting overall correctness.

To better understand the effect of data augmentation, we compare confusion matrices under both strict and lenient settings for the GPT-4o-mini model (Appendix B). In the lenient setting, slight improvements are observed after augmentation, but the gains are minimal—likely due to the small scale of augmentation relative to the underlying class imbalance. Under the strict setting, a few additional instances from the “No” and “To some extent” classes were correctly classified, confirming that the augmentation was directionally helpful. However, we also observe increased confusion within the “Yes” class, suggesting that the added synthetic data may have introduced mild noise. These trends indicate that while small-scale augmentation can be beneficial, its impact is limited and should be expanded or refined in future work.

6 Conclusion

In this work, we addressed the Mistake Identification sub-task of the BEA 2025 Shared Task, which evaluates whether AI tutors recognize student errors within educational dialogues. We explored both zero-shot and fine-tuned settings across several LLMs, including GPT-4o-mini, Mistral-7B, and LLaMA-3.1-8B. Our best-performing submit-

Model	Method	Validation Subset		Blind Test Set
		Strict F1	Strict Acc.	Strict F1 (submission)
GPT-4o-mini	Zero-shot	52.13	82.86	–
GPT-4o-mini	Fine-tuned on training subset	68.20	88.71	67.70
GPT-4o-mini	Fine-tuned on Full development set	–	–	71.63
GPT-4o-mini	Fine-tuned on augmented training subset	70.34	88.91	70.76
Mistral-7B	Zero-shot	51.73	70.16	–
Mistral-7B	Fine-tuned on training subset	62.61	87.10	–
Mistral-7B	Fine-tuned on augmented training subset	70.08	88.51	60.59
LLaMA-3.1-8B	Zero-shot	19.03	29.44	–
LLaMA-3.1-8B	Fine-tuned on training subset	41.52	84.48	–

Table 1: Performance comparison of all models under strict evaluation: The table reports strict macro-F1 and accuracy scores on the internal validation set and the official blind test set. The best-performing submitted model was GPT-4o-mini fine-tuned on the full development set, achieving a strict F1 score of 71.63% on the blind test.

Model	Method	Strict F1	Strict Acc.	Lenient F1	Lenient Acc.
GPT-4o-mini	Zero-shot	52.13	82.86	77.57	89.52
GPT-4o-mini	Fine-tuned on training subset	68.20	88.71	87.53	93.95
GPT-4o-mini	Fine-tuned on augmented training subset	70.34	88.91	88.36	94.35
Mistral-7B	Fine-tuned on training subset	62.61	87.10	85.71	92.74
Mistral-7B	Fine-tuned on augmented training subset	70.08	88.51	87.15	93.55

Table 2: Macro-F1 and accuracy scores are shown for both strict (3-way classification) and lenient (binary classification: Yes/To some extent vs. No) settings on the validation subset. GPT-4o-mini fine-tuned on the augmented training dataset performs best on the validation subset in both settings.

ted system, a GPT-4o-mini model fine-tuned on the full development set, achieved a strict macro-F1 score of 71.63% on the official blind test set, ranking second in the competition. These results highlight the value of lightweight fine-tuning in enhancing LLMs’ pedagogical sensitivity. Our findings support the ongoing effort to make LLM-based tutors not only fluent but diagnostically effective, capable of recognizing learner misconceptions and delivering instruction that aligns with educational goals.

7 Limitations

While our approach yielded strong results on the Mistake Identification sub-task, several limitations remain. First, the scale of training data, particularly for the “No” and “To some extent” classes,

was limited. Although synthetic augmentation improved model calibration, manual inspection of the generated examples was relatively permissive. In particular, some “No” examples included subtle guidance or hints that could blur the boundary with the “To some extent” class, introducing mild label noise. These were not filtered out during data selection and may have affected label consistency. Future work should explore more grounded augmentation strategies, along with stricter validation procedures to ensure correct label alignment.

Moreover, the models we used for evaluation in our study were also among those used to generate tutor responses for the development data. This overlap introduces potential bias, as models could be more inclined to align with responses produced by themselves or closely related variants. This

type of alignment can lead to overestimation of pedagogical quality of the tutor response.

Rose E. Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. [Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes](#). *Preprint*, arXiv:2310.10648.

References

Cohere. 2024. [Command r+ documentation](#). Accessed: 2025-05-21.

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](#). *Preprint*, arXiv:2407.09136.

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.

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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.

Charles Koutchme, 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](#). *Preprint*, arXiv:2405.05253.

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.

Meta. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

Mistral-AI. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.

OpenAI. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.

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.

Rose E. Wang and Dorottya Demszky. 2023. [Is chatgpt a good teacher coach? measuring zero-shot performance for scoring and providing actionable insights on classroom instruction](#). *Preprint*, arXiv:2306.03090.

A Development Set Class Distribution

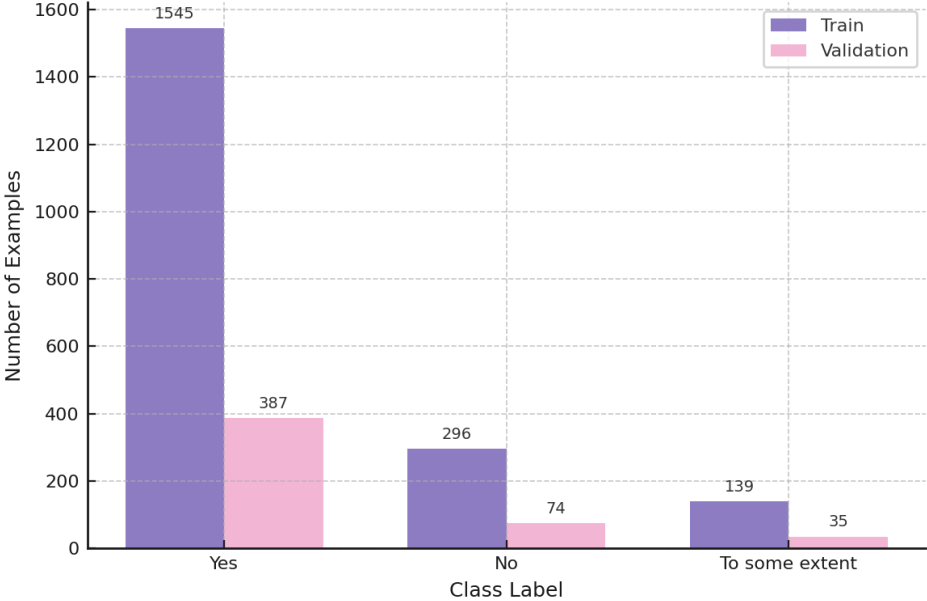


Figure 1: Class distribution in the original development set, split by training and validation subsets. This shows the class imbalance in the provided training data, motivating data augmentation.

B Confusion Matrices

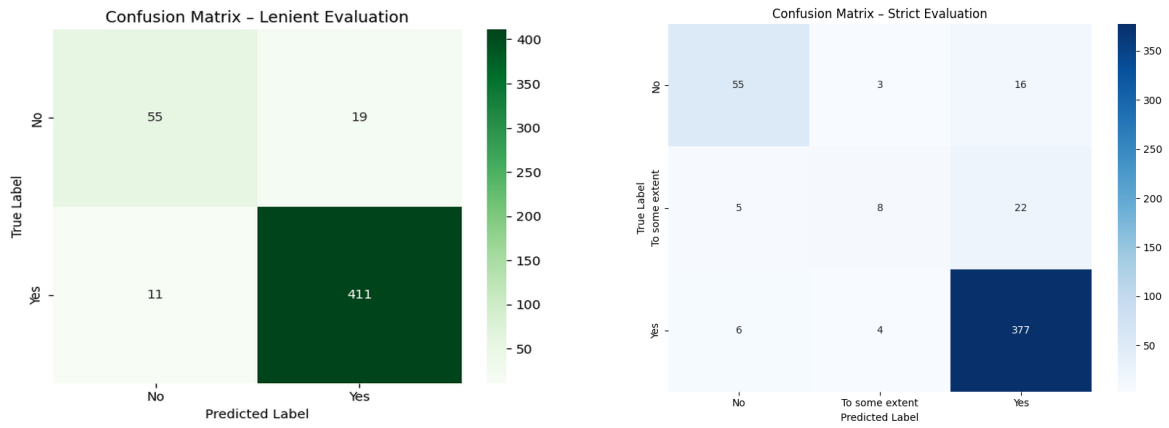


Figure 2: Confusion matrices for GPT-4o-mini fine-tuned on the original training subset.

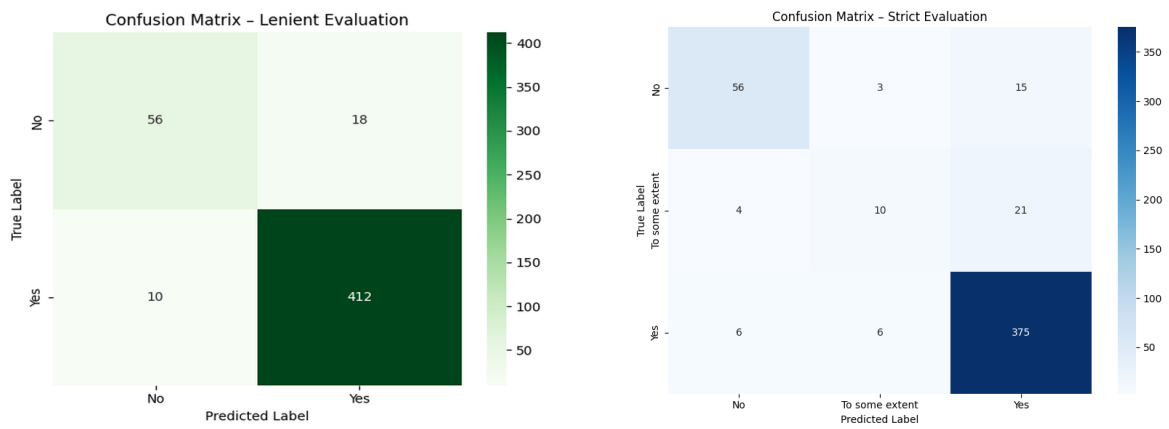


Figure 3: Confusion matrices for GPT-4o-mini fine-tuned on the augmented training subset.

C Prompt for Llama3.1 8B Instruct and Mistral 7B Instruct

Prompt Template

Instruction:

Evaluate the tutor's response based on whether they identified a mistake in the student's response or not. Mistake Identification: Has the tutor identified a mistake in the student's answer? Options: Yes, To some extent, No. Yes means the mistake is clearly identified or recognized in the tutor's response. No means the tutor does not recognize the mistake (e.g., they proceed to simply provide the answer to the asked question). To some extent means the tutor's response suggests that there may be a mistake, but it sounds as if the tutor is not certain. You should answer by Yes, No or To some extent strictly in the following format: Evaluation: (Yes, No, To Some Extent). It is very important to have the word Evaluation: before your answer, while also sticking to the criteria of evaluation.

Input:

{Conversation History + Tutor Response}

Response:

Evaluation: {Yes, No, or To Some Extent}

D Prompt for GPT-4o-mini

Prompt Format

System Message:

Classify the tutor's response to the student's answer based on whether the tutor has identified a mistake. Use the following labels: 'Yes' means the mistake is clearly identified; 'No' means the tutor does not recognize the mistake; 'To some extent' means the tutor suggests a mistake but is unsure. Respond strictly in the format: Evaluation: [Yes/No/To Some Extent].

User Message:

{Conversation History + Tutor Response}

Expected Output:

Evaluation: {Yes, No, or To Some Extent}

E Prompt for Data Augmentation with Command R+

Prompt for Generating "To Some Extent" Responses

Instruction:

You are a math tutor giving feedback to a student. Based on the conversation, write a single-sentence response that gently suggests the student may have made a mistake, but without clearly identifying what the mistake is. Your tone should sound uncertain, cautious, or exploratory. Do not explicitly say what is wrong. Do not state that something is definitely incorrect. Keep your response to ONE short sentence.

Input:

{Conversation History}

Output:

A single-sentence tutor response labeled "To some extent"

Two Outliers at BEA 2025 Shared Task: Tutor Identity Classification using DiReC, a Two-Stage Disentangled Contrastive Representation

Eduardus Tjitrahardja*, Ikhlasul Akmal Hanif*

Universitas Indonesia

{eduardus.tjitrahardja, ikhlasul.akmal}@ui.ac.id

<https://github.com/edutjie/DiReC>

Abstract

This paper presents DiReC (Disentangled Contrastive Representation), a novel two-stage framework designed to address the BEA 2025 Shared Task 5: Tutor Identity Classification. The task involves distinguishing between responses generated by nine different tutors, including both human educators and large language models (LLMs). DiReC leverages a disentangled representation learning approach, separating semantic content and stylistic features to improve tutor identification accuracy. In Stage 1, the model learns discriminative content representations using cross-entropy loss. In Stage 2, it applies supervised contrastive learning on style embeddings and introduces a disentanglement loss to enforce orthogonality between style and content spaces. Evaluated on the validation set, DiReC achieves strong performance, with a macro-F1 score of 0.9101 when combined with a CatBoost classifier and refined using the Hungarian algorithm. The system ranks third overall in the shared task with a macro-F1 score of 0.9172, demonstrating the effectiveness of disentangled representation learning for tutor identity classification.

1 Introduction

This paper presents the Two Outliers Tutor Identification Systems for Track 5 of the BEA 2025 Shared Task (Kochmar et al., 2025). The goal of this task is to recognize which response belongs to which tutor. We were provided with responses from nine different tutors, including two human tutors (novice and expert) and seven different Large Language Models (LLMs) (Abdin et al., 2024; OpenAI et al., 2024; Grattafiori et al., 2024; Team et al., 2024; Jiang et al., 2023) using data from MRBench (Maurya et al., 2024). For each question, all tutors provided an answer, and the objective is to develop a model capable of distinguishing between these tutor identities based on their responses.

Conversational agents, especially those powered by LLMs, are increasingly used in education to support student learning through interactive and tutor-like dialogue (Wollny et al., 2021; Tack et al., 2023). These systems can generate human-like, context-aware responses, offering new opportunities for scalable and personalized instruction. However, determining whether these models truly behave like effective tutors remains a challenge (Tack and Piech, 2022; Tack et al., 2023). This shared task explores whether it is possible to distinguish between responses generated by different AI tutors and human tutors.

Recent research has shown that models can be fine-tuned using contrastive loss to create powerful and representative embeddings. Powerful embedding models such as Jina, mE5, and BGE (Sturua et al., 2024; Chen et al., 2024; Wang et al., 2024) that are performing well in MTEB are trained using this approach (Muennighoff et al., 2023; Enevoldsen et al., 2025). Although a more common method for classification tasks involves using cross-entropy loss (Mao et al., 2023) to fine-tune encoder models like BERT (Devlin et al., 2019), contrastive learning approaches that produce high-quality embeddings and use simple classifiers have been shown to outperform this traditional method. In some cases, they even surpass large decoder-based models on classification benchmarks (Hanif et al., 2025; Muhammad et al., 2025). Furthermore, training models with contrastive loss directly on a downstream task has also demonstrated strong performance (Khosla et al., 2020; Muhammad et al., 2025). Motivated by these findings, our work explores contrastive learning as a strategy for tutor identification.

Contrastive loss is widely used in self-supervised learning to pretrain large language models by pulling together augmented views of the same input (Chen et al., 2020; Tao et al., 2024). For supervised tasks, supervised contrastive loss (Khosla

*Core contributor

et al., 2020) extends this by using label information, treating all samples with the same label as positives. This leads to more discriminative representations for classification.

Building on this foundation, we propose DiReC, a two-stage Disentangled Contrastive Representation framework for tutor response modeling. The core idea is to separate each response into two latent spaces: one that captures content (semantics, structure, factuality), and another that captures style (tone, verbosity, lexical choices), which is especially important for distinguishing among tutors. In the first stage, we train the model to learn content representations useful for tutor classification. In the second stage, we introduce supervised contrastive learning to the style space, encouraging similar representations across responses from the same tutor. By disentangling these factors, the model better captures tutor-specific traits while maintaining a coherent content backbone, improving both classification accuracy and interpretability.

2 System Overview

We propose a two-stage Disentangled Representation for Classification (DiReC) framework for tutor classification, which simultaneously learns content and style representations from text. The overall architecture is depicted in Figure 1.

2.1 Model Architecture

Given an input text sequence $x = (w_1, \dots, w_T)$, we first obtain contextualized token embeddings via a pretrained DeBERTa-v3-large encoder:

$$\mathbf{H} = \text{Enc}_\theta(x) \in R^{T \times d}, \quad \mathbf{h} = \mathbf{H}_{[\text{CLS}]} \in R^d.$$

Two parallel projection heads then map \mathbf{h} into the content and style subspaces of dimension p :

$$\mathbf{c} = f_{\text{content}}(\mathbf{h}) \in R^p, \quad \mathbf{s} = f_{\text{style}}(\mathbf{h}) \in R^p.$$

The content embedding \mathbf{c} is intended to capture relevant semantic information for the identification of the tutor, while the style embedding \mathbf{s} captures stylistic traits. We concatenate these vectors and feed them to a linear classifier g over K tutor classes:

$$\mathbf{z} = g([\mathbf{c}; \mathbf{s}]) \in R^K, \quad \hat{y} = \arg \max_j z_j.$$

2.2 Two-Stage Training Procedure

Training alternates between two stages to disentangle style from content:

Stage 1 (Cross-Entropy Only). While contrastive loss is effective at capturing stylistic similarity, it does not explicitly enforce class separation nor provide a direct classification signal. Therefore, cross-entropy acts as a necessary foundation to learn robust content features before style-specific objectives are introduced.

In the first stage, we freeze the style head f_{style} and train the encoder Enc_θ , content head f_{content} , and classifier g using standard cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(z_{i,y_i})}{\sum_{j=1}^K \exp(z_{i,j})},$$

where z_{ij} is the logit for sample i and class j , and z_{i,y_i} is the logit for the true class label y_i of sample i . This loss encourages the model to learn discriminative content representations that effectively differentiate tutors based on semantic and structural aspects of their responses.

Stage 2 (Joint Contrastive & Disentanglement).

In the second stage, we unfreeze the style head and optimize it jointly with the rest of the model. We apply supervised contrastive loss on style embeddings to capture tutor-specific writing traits, encouraging embeddings from the same tutor to cluster regardless of content variation. Simultaneously, a disentanglement loss penalizes high similarity between content and style embeddings, preventing redundancy and promoting specialization of each representation. This joint training improves the model’s ability to separately encode semantic content and stylistic nuances, enhancing both interpretability and classification performance.

- $\mathcal{L}_{\text{SupCon}}$ is the supervised contrastive loss applied to style embeddings:

$$\mathcal{L}_{\text{SupCon}} = -\frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \log \frac{\exp(\mathbf{s}_i^\top \mathbf{s}_j / \tau)}{\sum_{k \neq i} \exp(\mathbf{s}_i^\top \mathbf{s}_k / \tau)},$$

where \mathcal{P} indexes all positive pairs sharing the same tutor label, and τ is a temperature hyperparameter.

Unlike the original formulation by Khosla et al., which uses log-softmax over multiple positives and negatives per anchor, our version is simplified. Since the main supervision is already provided via cross-entropy classification, the contrastive loss acts as an auxiliary signal to refine stylistic clustering, making a lighter pairwise variant sufficient and more efficient.

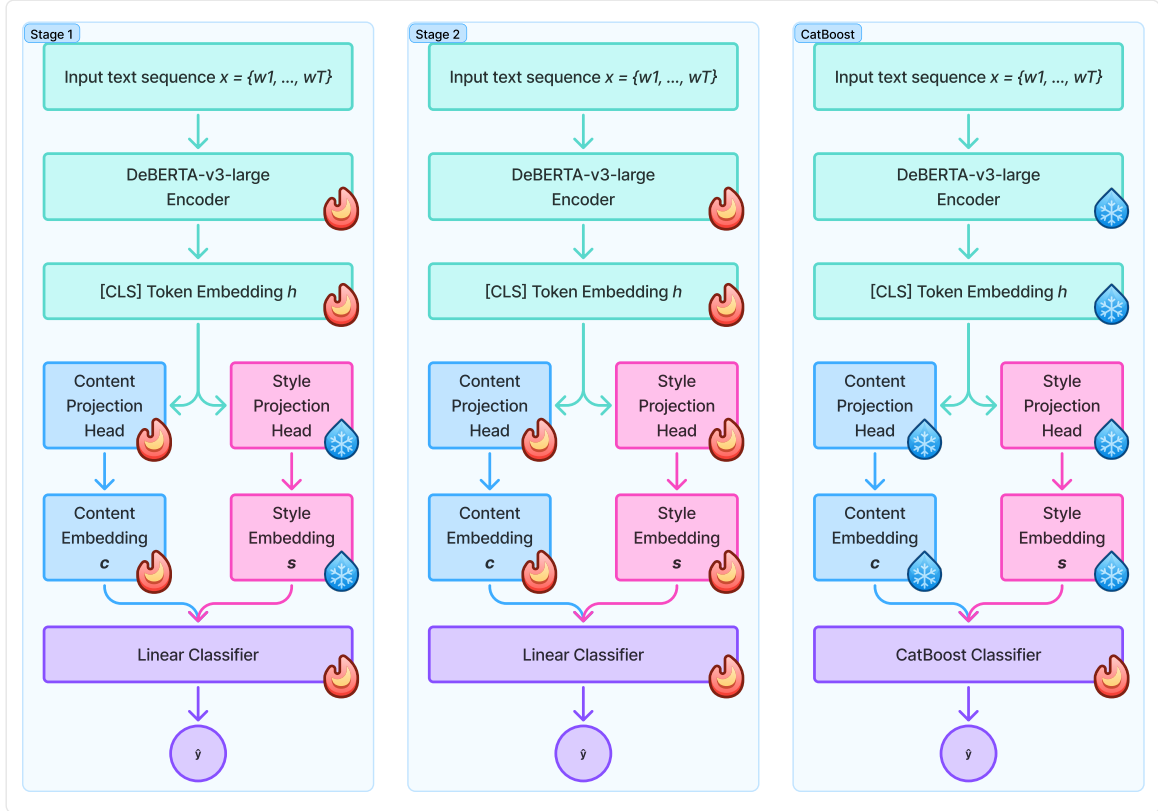


Figure 1: DiReC Architecture. Trainable components are marked with 🔥, while frozen components are indicated with ❄️.

- \mathcal{L}_{dis} is a cosine-based disentanglement loss that penalizes similarity between content and style embeddings:

$$\begin{aligned} \mathcal{L}_{\text{dis}} &= \frac{1}{N} \sum_{i=1}^N |\cos(\mathbf{c}_i, \mathbf{s}_i)| \\ &= \frac{1}{N} \sum_{i=1}^N \left| \frac{\mathbf{c}_i^\top \mathbf{s}_i}{\|\mathbf{c}_i\| \|\mathbf{s}_i\|} \right|. \end{aligned}$$

Finally we calculate all the loss using

$$\begin{aligned} \mathcal{L} &= \lambda_{\text{CE}} \mathcal{L}_{\text{CE}} + \lambda_{\text{sty}} \mathcal{L}_{\text{SupCon}}(\{\mathbf{s}_i, y_i\}) \\ &\quad + \lambda_{\text{dis}} \mathcal{L}_{\text{dis}}(\mathbf{c}, \mathbf{s}). \end{aligned}$$

2.3 Experimental Setup

All experiments were conducted using a consistent set of hyperparameters (Table 3). At the onset of Stage 2, we halved the learning rate and enabled mixed-precision optimization (AdamW + GradScaler) to stabilize fine-tuning. The parameters are provided in Appendix A.

Unless stated otherwise, all single-stage experiments were trained for a total of 5 epochs. The

two-stage DiReC model was initially set to train for 5 epochs during Stage 1 (content-only training with the style head frozen), followed by up to 5 additional epochs in Stage 2 (joint training with both heads unfrozen). In practice, however, the best validation checkpoint was achieved at epoch 6 (the first epoch of Stage 2). For clarity in the 3.1 subsection, we therefore refer to the two-stage model as effectively trained for 6 epochs in total.

At test time, we compute content and style embeddings jointly, concatenate them, and feed the resulting vector into the classifier g . The disentanglement enforced during training ensures that \mathbf{c} and \mathbf{s} capture complementary information, improving both generalization and interpretability in tutor prediction.

3 Result

3.1 Development

We conducted a series of experiments to validate the components of the DiReC framework. Table 1 summarizes the macro-F1 in the validation set for each setting.

Experiment	Val. F1 (Macro)
Single-stage DiReC	0.8720
Only content projection	0.8845
Only style projection	0.8692
Two-stage DiReC	0.9042
Two-stage DiReC + Cat-Boost classifier	0.9101

Table 1: Validation Macro-F1 scores for development experiments.

Single-Stage First, we trained the DiReC model in a single stage, which yielded a macro-F1 of 0.8720.

Projection-Head Ablation Next, we performed an ablation on the two projection heads to assess its standalone contribution. Training with only the content head for 5 epochs yielded a macro-F1 of 0.8845, whereas using only the style head under the same epochs fell to 0.8692. Moreover, when we extended the content-only model to 6 epochs—to match the total training steps of our two-stage strategy—its performance dropped further to 0.8730, indicating overfitting in the absence of style guidance. These results confirm that the content subspace carries the most of the classification signal, and naively prolonging content-only training can actually harm generalization.

Two-Stage DiReC We observed that introducing the style projection head from the initial stage of training could potentially hinder the development of the content projection’s discriminative capabilities. However, a naive extension of content-only training often led to overfitting. Consequently, we hypothesized that treating the style learning as a subsequent refinement phase could be beneficial. To address these limitations, we adopted the two-stage DiReC strategy (Section 2), introducing the style head only after the content pathway had converged. This staged training approach achieved a validation macro-F1 score of 0.9042, with the best model obtained at epoch 6—the first epoch of Stage 2. It outperformed both content-only baselines, which achieved scores of 0.8845 at epoch 5 and 0.8730 at epoch 6, establishing our strongest benchmark among purely neural network models.

CatBoost on Learned Embeddings Finally, we replaced the model’s linear classifier with a CatBoost classifier (Prokhorenkova et al., 2019)

trained on the concatenated style||content embeddings. This hybrid approach further improved validation macro-F1 to 0.9101.

Embedding Clustering Evolution Figures 2a–2c visualize the t-SNE projections of content embeddings at epochs 1, 3, and 6. Early in training (Figure 2a), tutor clusters overlap greatly. By epoch 3 (Figure 2b), distinct clusters begin to form, and by epoch 6 (Stage 2) (Figure 2c) each tutor’s content representations occupy tight, well-separated regions. This suggests that DiReC has effectively learned to represent tutor-specific content characteristics.

Validation Confusion Matrix Analysis Figure 3 shows the two-stage DiReC + CatBoost classifier confusion matrix on the validation set. The strong diagonal indicates high overall classification accuracy, with most tutor identities being correctly predicted. However, some misclassification between different tutor identities is observable.

The highest confusion occurs between Llama3.1-8B and Llama3.1-405B. Specifically, 8 instances of Llama3.1-8B are misclassified as Llama3.1-405B, and 7 instances of Llama3.1-405B are misclassified as Llama3.1-8B. This is likely attributable to the inherent similarity in response styles and content patterns originating from the same Llama model family. Nevertheless, the majority of tutors are classified with high precision and recall, with the primary challenge lying in distinguishing between closely related model variants, highlighting the effectiveness of the disentangled representations.

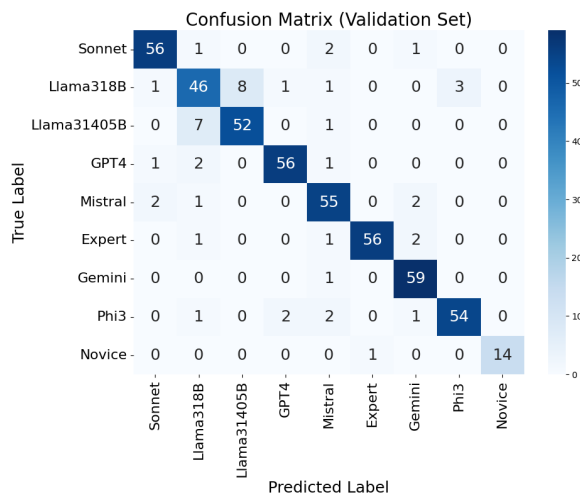


Figure 3: Validation confusion matrix for the two-stage DiReC + CatBoost classifier.

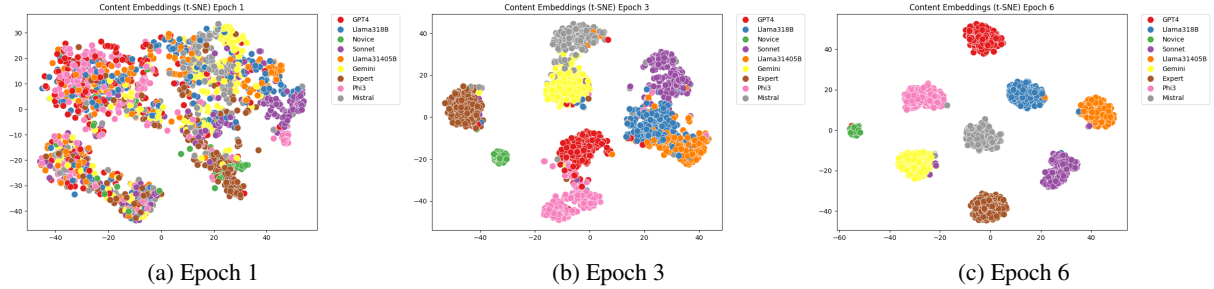


Figure 2: Evolution of content-embedding clusters over training.

3.2 Submission

Each conversation in the dataset contains K utterances, each from a different tutor, requiring a one-to-one mapping between utterances and tutor labels. However, our CatBoost classifier predicts labels independently, which can result in duplicate tutor assignments.

To enforce uniqueness, we apply the Hungarian algorithm as a post-processing step (Crouse, 2016). For each conversation g , we create a $K \times K$ probability matrix $\mathbf{P}^{(g)}$, where $P_{ij}^{(g)}$ is the predicted probability that utterance i belongs to tutor j . We seek the assignment σ^* that maximizes the total confidence:

$$\sigma^* = \arg \max_{\sigma \in S_K} \sum_{i=1}^K P_{i, \sigma(i)}^{(g)}$$

Since SciPy’s `linear_sum_assignment`¹ minimizes cost, we negate the probabilities to form a cost matrix $\mathbf{C}^{(g)}$, with $C_{ij}^{(g)} = -P_{ij}^{(g)}$. This ensures a unique, high-confidence mapping between utterances and tutor labels. This procedure refines the initial probabilistic predictions from the classifier to adhere to the structural constraint of the problem for each conversation.

Rank	Team	F1	Acc
1	Phaedru	0.9698	0.9664
2	SYSUpporter	0.9692	0.9657
3	Two Outliers	0.9172	0.9412

Table 2: Final leaderboard results of the shared task. Our team, Two Outliers, finished in third place.

The predictions from the two-stage DiReC model combined with the CatBoost classifier, fur-

¹https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html

ther refined using the linear sum assignment strategy, were submitted to the official Codabench leaderboard. This approach achieved a final macro-F1 score of **0.9172**, placing third in the shared task, as shown in Table 2.

4 Conclusion

In this work, we proposed DiReC, a two-stage framework that leverages disentangled contrastive representation learning for the task of tutor identity classification. Our approach separates content and style embeddings to capture both semantic and tutor-specific stylistic characteristics, resulting in improved classification accuracy and interpretability. Empirical evaluations on the BEA-2025 Shared Task data show that the two-stage DiReC model outperforms single-stage baselines and benefits from contrastive refinement and disentanglement. Additionally, incorporating a CatBoost classifier and applying a Hungarian algorithm for structured post-processing further enhanced performance, culminating in a top-three placement in the official leaderboard. These results highlight the potential of disentangled representation learning in modeling nuanced tutor behavior across human and AI-generated responses.

Limitations

Due to time and computational constraints, we did not perform thorough hyperparameter tuning. Several important parameters, including the contrastive temperature, the weights for the cross-entropy loss, style loss, and disentanglement loss, were chosen heuristically without extensive validation. Additionally, core training settings such as the learning rate, batch size, and number of training epochs were fixed throughout our experiments. These parameters may significantly influence model performance, and future work could focus on systematically tuning them to achieve further improvements.

References

- Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2318–2335.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmlR.
- David F Crouse. 2016. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *Bert: Pre-training of deep bidirectional transformers for language understanding*. *Preprint*, arXiv:1810.04805.
- Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Sibli, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrom, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Mariya Hendriksen, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Šuppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri Krishnakumar, Anna Maksimova, Silvan Wehrli, Maria Tikhonova, Henil Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Hongjin Su, Jimmy Lin, Howard Yen, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. 2025. *Mmteb: Massive multilingual text embedding benchmark*. *Preprint*, arXiv:2502.13595.
- 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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Al-lonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esibou, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kam-bador, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparth, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal

Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymur, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khanelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov,

Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Reemer, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihalescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaoqian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

Ikhlasul Akmal Hanif, Eryawan Presma Yulianrifat, Jaycent Gunawan Ongri, Eduardus Tjitrahardja, Muhammad Falensi Azmi, Rahmat Bryan Naufal, and Alfian Farizki Wicaksono. 2025. [University of indonesia at semeval-2025 task 11: Evaluating state-of-the-art encoders for multi-label emotion detection](#). *Preprint*, arXiv:2505.16460.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron

- Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*, Vienna, Austria. Association for Computational Linguistics.
- Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023. Cross-entropy loss functions: Theoretical analysis and applications. *Preprint*, arXiv:2304.07288.
- Kaushal Kumar Maurya, KV Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2024. Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors. *arXiv preprint arXiv:2412.09416*.
- Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. *Preprint*, arXiv:2210.07316.
- Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Seid Muhie Yimam, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine De Kock, Tadesse Destaw Belay, Ibrahim Said Ahmad, Nirmal Surange, Daniela Teodorescu, David Ifeoluwa Adelani, Alham Fikri Aji, Felermimo Ali, Vladimir Araujo, Abinew Ali Ayele, Oana Ignat, Alexander Panchenko, Yi Zhou, and Saif M. Mohammad. 2025. SemEval task 11: Bridging the gap in text-based emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, Vienna, Austria. Association for Computational Linguistics.
- 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, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeef Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Bar-

- ret Zoph. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2019. [Catboost: unbiased boosting with categorical features](#). *Preprint*, arXiv:1706.09516.
- Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. [jina-embeddings-v3: Multilingual embeddings with task lora](#). *Preprint*, arXiv:2409.10173.
- 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](#). *Preprint*, arXiv:2306.06941.
- 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.
- Chongyang Tao, Tao Shen, Shen Gao, Junshuo Zhang, Zhen Li, Zhengwei Tao, and Shuai Ma. 2024. Llms are also effective embedding models: An in-depth overview. *arXiv preprint arXiv:2412.12591*.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Serincinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornaraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurusurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakob Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Fer- yal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayanan Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey

Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkels-son, Marcello Maggioni, Daniel Zheng, Yury Sul-sky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tshilas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohanane, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Has-sas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin John-son, Behnam Neyshabur, Justin Mao-Jones, Ren-shen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Se-bastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangoeei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Ku-mar, Colton Bishop, Adams Yu, Sarah Hodgkin-son, Sid Mittal, Premal Shah, Alexandre Moufarez, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Char-lotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa,

Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiuqia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Laksh-minarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrit-twieser, Elena Buchatskaya, Soroush Radpour, Mar-tin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kan-nan, David Kao, Parker Schuh, Axel Stjerngren, Gol-naz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Fe-lipe Tiengo Ferreira, Aishwarya Kamath, Ted Kli-menko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Fel-ix de Chaumont Quiry, Charline Le Lan, Tom Hud-son, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Deven-dra Sachan, Srivatsan Srinivasan, Hannah Mucken-hirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xi-ang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexan-der Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swa-roop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, An-ton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirsenschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizh-skaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Gar-rido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven

Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Koppurapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Iliia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Vellela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Ram-mohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Tsendsuren Munkhdalai, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Rudderock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnappalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona

Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Bartek Perz, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srin Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremen Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemnyy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kepa, François-Xavier Aubet, Anton Algymer, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohmman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeff Dean, and Oriol Vinyals. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.](#)

Preprint, arXiv:2403.05530.

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.

Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachslar. 2021. Are we there yet?-a systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.

A Appendix

Hyperparameter	Value
Maximum sequence length (MAX_LEN)	256 tokens
Batch size (BATCH_SIZE)	32
Initial learning rate (LR)	2×10^{-5}
Encoder embedding size (EMBED_SIZE)	1024
Projection dimension (PROJ_SIZE)	256
Contrastive temperature (τ)	0.07
CE loss weight (λ_{CE})	1.0
Style loss weight (λ_{sty})	0.3
Disentanglement weight (λ_{dis})	0.1

Table 3: Hyperparameter settings for all DiReC experiments.

Archaeology at BEA 2025 Shared Task: Are Simple Baselines Good Enough?

Ana-Maria Roşu
anaros766@gmail.com

Jany-Gabriel Ispas
iani.ispas@gmail.com

Sergiu Nisioi*
sergiu.nisioi@unibuc.ro

Human Language Technologies Research Center
Faculty of Mathematics and Computer Science
University of Bucharest

Abstract

This paper describes our approach to the 5 classification tasks from the Building Educational Applications (BEA) 2025 Shared Task. Our methods range from classical machine learning models to fine-tuning large-scale transformer architectures. Despite the diversity of techniques, performance differences were often minor, suggesting the presence of strong surface-level signal in the data and a limiting effect of annotation noise – particularly around the “To some extent” label. Under lenient evaluation, simple models perform competitively, showing their effectiveness in low-resource settings. Our submissions rank in the top 10 in three out of five tracks. The code and models are publicly available at: <https://github.com/ana-rosu/Archaeology-at-BEA2025>

1 Introduction

This paper presents an exhaustive set of experiments conducted for the BEA 2025 Shared Task, which revolves around assessing the pedagogical abilities of AI tutors in educational dialogues within the mathematical domain.

We start with classical machine learning methods like logistic regression over TF-IDF encodings and String Kernel SVMs, gradually scaling up to more complex approaches such as zero-shot and few-shot prompting with Mistral-7B-Instruct (Jiang et al., 2023), feature-based methods using frozen transformer representations (from models like ModernBERT (Warner et al., 2024a) and GritLM (Muennighoff et al., 2024)), decoder-style architectures such as GPT2-XL (Radford et al., 2019) combined with a linear classification head, parameter-efficient fine-tuning with LoRA adapters in 4-bit precision on Mistral-7B, as well as BERT-like classifiers (e.g., RoBERTa (Liu et al., 2019), ModernBERT (Warner et al., 2024b), DeBERTa (He et al., 2021)).

Our best-performing submissions across all tracks use fine-tuned BERT-style classifiers. Final submissions are selected in an unsystematic way due to the five-submission limit per track; we focus on choosing the models that perform best on our local validation set, while also ensuring that they differ from each other. Although most of our submissions are based on Masked Language Models, we include a broader set of experiments in this paper to document our development process and highlight that some alternative approaches remain competitive.

We place greater emphasis on Track 1 (Mistake Identification), as it is the first task we explore and serves as a foundation for the others. Some of our preliminary experiments, including prompting and decoder-based fine-tuning, are conducted exclusively on this track.

Despite using a wide range of models, we observe that performance is often surprisingly similar across setups, suggesting that model architecture may not be the dominant factor for this task. One possible explanation is that subtle annotation inconsistencies, especially between “Yes” and “To some extent”, introduce noise that limits performance (see Appendix G). We notice that tutor responses with very similar wording (e.g., “Please recheck your answer”) are labeled “Yes” in some dialogues and “To some extent” in others. In this context, the order in which training examples are presented becomes important, especially in such a small and imbalanced setting. When the model sees one interpretation early on, it may implicitly learn to generalize that decision across similar examples, reinforcing a bias. This makes the optimization sensitive to random factors such as batch order or initialization.

The “To some extent” label is the main source of difficulty in this task. Without it, the classification becomes much easier, a fact supported by the lenient evaluation scores, which reach or even

*Corresponding author.

exceed 85% F1 on all tracks but one (Providing Guidance, where the best lenient performance on the public leaderboard is 78% F1), suggesting that models perform well when ambiguity is removed from the label space.

When evaluated under the lenient setting (“Yes” and “To some extent” are merged into a single class), traditional machine learning models have surprisingly strong performance even with minimal effort, using default configurations. As shown in Table 9, the validation accuracies achieved with these models are very close to the best public leaderboard results, with gaps between 0.37%-4.15%. In terms of Macro F1, these models also achieve competitive scores, with gaps between 3.70%-10.41%, demonstrating that pedagogical signal can be captured effectively under a binary framing, making them strong baselines in scenarios with constrained resources.

The presence of strong surface-level signal may allow even simple models to perform well. Another potential hypothesis is that there is simply not enough data for larger models to generalize better.

Our team’s submissions were competitive across all tracks:

Track 1 (Mistake Identification): 8th out of 44 teams

Track 2 (Mistake Location): 12th out of 31 teams

Track 3 (Providing Guidance): 13th out of 35 teams

Track 4 (Actionability): 7th out of 29 teams

Track 5 (Tutor Identification): 6th out of 20 teams

Team ranks are based on the results according to the main shared task metric – exact Macro F1 score.

2 Data and Tasks

2.1 Shared-Task Tracks

The data provided for this shared task builds on MRBench, a dataset of short alternate-turn dialogues sourced from MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024). Each dialogue is annotated for eight pedagogical dimensions based on a unified evaluation taxonomy introduced by Maurya et al. (2025a). This taxonomy reflects core learning sciences principles and builds on prior work in AI tutor evaluation (Tack and Piech, 2022; Daheim et al., 2024; Wang et al., 2024)

The task focuses on four key dimensions – which also form the first four of the five tracks in the BEA 2025 shared task:

Track 1 - Mistake Identification: determine if

the tutor identifies the student’s mistake.

Track 2 - Mistake Location: determine if the tutor pinpoints where the mistake occurs.

Track 3 - Providing Guidance: determine if the tutor gives helpful and relevant feedback.

Track 4 - Actionability: determine if the student can clearly understand what to do next.

Track 5 - Tutor Identification: predict which tutor produced the response.

2.2 Dataset

The dataset includes 300 dialogues in the development set and 191 in the test set, each paired with responses from both human tutors (Expert and Novice) and 7 LLM-based tutors (GPT4 (OpenAI et al., 2024), Gemini (Team et al., 2025), Llama31405B (Grattafiori et al., 2024), Llama318B, Mistral (Jiang et al., 2023), Phi3 (Abdin et al., 2024), Sonnet (Anthropic, 2024)). Each dialogue ends with a student turn that contains a mistake, confusion, or misconception, to which multiple tutor responses are provided. Every tutor reply is annotated with gold-standard labels along four dimensions – Mistake Identification, Mistake Location, Providing Guidance, and Actionability – using a three-class scheme: “Yes”, “To some extent”, and “No”.

The label distribution is imbalanced across tasks, with “Yes” being the majority class, “No” moderately represented, and “To some extent” notably underrepresented (Figure 1). Furthermore, the 2D scatter plots (Figure 2), generated using t-SNE on tutor response embeddings extracted from the ModernBERT-large model, show that responses labeled “No” tend to form small, tight clusters, regardless of the task. These responses often share similar semantic structures, such as starting with phrases like “Good job!”, “Good catch!”, or “You are absolutely correct.” In contrast, responses labeled “Yes” show consistent distributions across tasks, suggesting that positive responses are more generalizable. The Actionability task exhibits the highest dispersion among “No” responses.

During our experiments, we identify some cases of label inconsistencies, especially between the labels “Yes” and “To some extent”, which we report in Appendix G.

2.2.1 Training and Validation Splits

For model development, we create separate train-validation splits for each of the first four tasks, to accommodate the varying label distributions across

them. It is important to ensure that all samples from the same conversation remain in the same split to avoid data leakage – otherwise, multiple tutor responses with the same conversation history could appear in both training and validation sets. For that, we group all samples by *conversation_id* and compute the majority label for each dialogue. This majority label is then used to perform stratified sampling, helping us preserve the overall class distribution of the full development set in both the training and validation sets. An 80/20 train/validation ratio is used, with a fixed random seed for reproducibility. Detailed statistics on the splits, including label ratios and counts, are provided in Appendix A.

For Track 5, we perform a stratified 80/20 train/validation split to maintain balanced proportions of tutor identities across both sets. Unlike the other tracks, grouping by *conversation_id* is not required here, since all samples with the same *conversation_id* share an identical conversation history that includes previous tutor turns not authored by the tutor being identified. As a result, only the final generated response can be used to distinguish between them.

For each track, we submit the runs that achieve the highest validation performance. In addition, we include results from lower-performing methods to document the full range of approaches explored.

2.3 Evaluation Metrics

According to Kochmar et al. (2025), Tracks 1 – 4 (Mistake Identification, Mistake Location, Providing Guidance, and Actionability) are evaluated using Macro F1 as the main metric, with accuracy as the secondary metric. The evaluation is done in two ways:

- **Exact evaluation:** the model has to predict the correct label among the three options (“Yes”, “To some extent”, or “No”).
- **Lenient evaluation:** “Yes” and “To some extent” are combined into a single class and compared against “No”.

Track 5 (Tutor Identity) is a 9-class classification task, evaluated using Macro F1 as the main metric and accuracy as the secondary metric, without any lenient setting.

3 Methods

3.1 Traditional ML Methods

As a baseline¹ we use traditional machine learning models across all tracks. For all experiments in this approach, we use TF-IDF for feature extraction covering both unigrams and bigrams from the input text.

Logistic Regression gives us a baseline with a Macro F1 of 0.63 on Track 1, and confirms that the TF-IDF features are useful. We train the model using balanced class weights to handle label imbalance and set the maximum number of iterations to 1000. XGBoost, which is known for strong performance on tabular data and classification tasks (McElfresh et al., 2023), reaches a Macro F1 of 0.625 for Track 1, slightly below Logistic Regression. In all our experiments with this model, we use a learning rate of 0.1, 200 estimators, and set both subsample and colsample bytree to 0.8 for regularization. Ensembling XGBoost with other boosting methods such as LightGBM provides small improvements in a few cases, but overall, the results remain close to those obtained with XGBoost alone (see Table 8).

We also explore a character-level string kernel using an SVM with a precomputed spectrum kernel (Ionescu and Butnaru, 2018). The string spectrum kernel measures the similarity between two strings, s_1 and s_2 , based on their n -grams. It is defined as:

$$K_{\text{spectrum}}(s_1, s_2) = \frac{\sum_{u,v} \kappa(u, v)}{\sqrt{\sum_u \kappa(u, u) \sum_v \kappa(v, v)}}$$

where u and v represent the n -grams (substrings of lengths $\in [2, 5]$) from s_1 and s_2 , respectively. And $\kappa(u, v)$ is a dot product over binary occurrences of n -grams u and v .

This approach appears competitive to deep-learning based models on several tracks. For example, on Mistake Identification it achieves a Macro F1 score of 0.6346, indicating that a lot of the signal can be captured just by comparing strings directly. We take this as an indicator that some of the Yes/No annotations for different tasks share similar-looking strings.

For these experiments, all the validation results are included in the Appendix E.

¹We also use LLM prompting the Mistral-7B-Instruct model, but results are weaker than even traditional ML baselines - best F1 on Task 1 is 0.42. In the few-shot setting, best is 0.45. We describe the entire approach in Appendix C

3.2 Frozen Embeddings + Linear Classifiers

We compare several feature extraction strategies using **ModernBERT-large** for logistic regression classifiers (*max_iter=2000*, *class_weight='balanced'*) on Mistake Identification, Mistake Location, Providing Guidance and Actionability. All embeddings are computed over the tutor response only, without including any surrounding conversational context. We evaluate 4 pooling methods: the final hidden state of the [CLS] token, mean pooling, max pooling, and a concatenation of [CLS] and mean. Results are summarized in Appendix E.1.1.

Mean pooling appears to perform slightly better when tasks do not require fine-grained distinctions, such as in Mistake Identification and Actionability, likely due to better aggregation of distributed semantic cues across the tutor response. For example, using mean-pooled embeddings on the Mistake Identification task, the classifier achieves a Macro F1 of ~ 0.65 on the fixed validation split, outperforming [CLS] pooling (~ 0.62). However, [CLS] pooling demonstrates superior performance on the validation split on more complex tasks like Mistake Location and Providing Guidance. These likely require more nuanced representations. The complexity of these tasks is further evidenced by their overall performance, results remaining notably lower, suggesting they depend more on the dialogue context and how the response relates to the student's earlier reasoning, which cannot be captured in the response itself.

For Mistake Identification, on the other hand, signal can be inferred from the response alone from the presence/absence of corrective language. Similarly, the model performs well on Actionability under the same conditions, likely because actionable feedback is sometimes expressed directly in the tutor's reply through question words that encourage the student to take action. As a result, the signal required for predicting Mistake Identification and Actionability is more localized, allowing the classifier to perform well without access to prior student turns.

Additionally, we extract embeddings from intermediate layers for Mistake Identification, motivated by findings from [Skean et al. \(2025\)](#) that middle-to-late layers may encode more useful information for the MTEB benchmarks. In our case, the performance peaks around layers 9 and 15 for mean and CLS respectively.

Last but not least, we explore **GritLM** - a **Mistral-based** 7b parameter fine-tuned using GRIT ([Muennighoff et al., 2024](#)). This autoregressive model achieves state-of-the-art results on MTEB benchmarks. We compare the embeddings extracted from different layers combined with several classifiers: logistic regression, a multi-layer perceptron (MLP), a Gaussian Naive Bayes and a k-nearest neighbor models. For GritLM we do not observe any significant decay in performance from middle layers up until the final ones (see [Figure 9](#)). The weakest classifiers are the KNN and GaussianNB, while between MLP and logistic regression there does not seem to be a clear winner. Our submission number 2 on Mistake Identification obtains 0.6532 F1 score on the final leader board using the embeddings from layer 24. The comparative results across layers on the validation set are included in Appendix E.1.2.

3.3 Decoder LM Fine-Tuning

We experiment with full fine-tuning of **GPT2-XL** on the Mistake Identification task by applying mean pooling over its last hidden state and training a linear classification head. This setup achieves a Macro F1 score of 0.65 on local split using only the tutor responses as input. We also explore the frozen version of GPT2-XL, updating only the final transformer block. This approach reaches 0.55 Macro F1. We do not pursue these experiments further as the performance plateaued even when experimenting with stratified batches, alternative loss functions, and varying input context on the last-transformer-block version. Configuration: *epochs=10*, *batch_size=32*, *lr=2e-5*, *dropout=0.1*, *loss_fn=CrossEntropyLoss()*, *optimizer=AdamW*. This result reinforces that pedagogical signal detection requires specialized approaches rather than simply scaling model size.

3.4 BERT-like encoders

The final best results are obtained by fine-tuning masked language models. We experiment with three model families: **RoBERTa** ([Liu et al., 2019](#)), **DeBERTa** ([He et al., 2021](#)), and **ModernBERT** ([Warner et al., 2024b](#)). For all models, we apply a linear classification head on top of the final hidden state of the first token (corresponding to the [CLS] token). No additional pooling or attention mechanisms are introduced beyond the pretrained architecture. We begin with base-sized variants on Track 1, but the better performance of the large

variants motivates us to adopt them for our next experiments on all tracks.

For each track and model, we compare three main input formats, which we refer to throughout the paper as:

- *response-only*: consists only of the tutor’s response, isolating the pedagogical value of the response itself without surrounding dialogue
- *context*: includes the final student turn concatenated with the tutor’s response, capturing the local misunderstanding or confusion the tutor is addressing
- *full context*: includes the entire conversation history preceding the tutor’s response enabling multi-turn reasoning over the dialogue and potentially identify earlier misalignments

These representations allow us to assess how much conversational context is necessary or beneficial for each track, and how different models leverage that context.

To address the severe class imbalance and reduce bias toward the majority label, we experiment with three loss functions: standard cross-entropy as a baseline; class-weighted cross-entropy, where class weights are set to the inverse of class frequencies; and focal loss (Lin et al., 2018), with $\gamma \in [1.0, 3.0]$ and various α configurations, including uniform ($\alpha = [1.0, 1.0, 1.0]$), inverse-frequency class weights, and class-balanced α as proposed by Cui et al. (2019).

We also experiment with prepending natural language task prompts to the input, inspired by recent work on instruction tuning and prompt-based adaptation. These prompts frame the classification task using instructions, such as ordinal scales (“To what extent does the tutor identify the mistake? 0 = not at all, 1 = partially, 2 = fully”) or evaluator roles (“You’re evaluating a tutor’s response. Score how clearly they identify the student’s mistake”). The prompt text is prepended to the input before tokenization. Although BERT-like models are not autoregressive, we find that in some cases, prompts improve validation performance and make the task framing more consistent across examples (see Appendix B). Further exploration is needed to fully quantify their impact, but we include this as a promising direction for instruction-aware encoder fine-tuning.

3.5 Submissions

3.5.1 Mistake Identification

Submission 1 uses a fine-tuned **RoBERTa-large** model, trained with context input format and focal loss ($\gamma = 2.0$, uniform α) for 4 epochs. For all hyperparameters and the approach used for selecting the input configuration and loss function, refer to Appendix D.1. We train the model on five random seeds, average the logits across seeds, and apply post-training calibration using temperature scaling and per-class thresholding based on validation performance. On the validation set, this approach achieves a 0.7072 Macro F1. In the public leaderboard, it obtains **0.6919** Macro F1, making it our second-best overall submission.

Training observations:

Initially, random batches leads the model to see mostly majority-class examples early on, which causes a bias to predict predominantly a single label (e.g., "Yes"), hard to correct in later stages. This is visible in the first-epoch confusion matrix.

To resolve this, we implement a custom stratified batch sampler that maintains around the same class ratios as the full training set within each batch, which proves beneficial for small batch sizes in our setup, where a random batch could otherwise contain only examples from the "Yes" class. This helps the model learn minority classes from the start.

Submission 2 uses embeddings from layer 24 of **GritLM** (Muennighoff et al., 2024), selected based on validation performance (Figure 9). The classifier is an ensemble of logistic regression and a multilayer perceptron (MLP) with a hidden size of 100. The best development set score is 0.71, while the leaderboard score is **0.65**. The performance gap indicates overfitting and suggests that layer-wise performance variation can significantly affect decisions, as such, high evaluation scores may not generalize well on new test sets.

Submission 3 is fine-tuned on the **mistralai/Mistral-7B-v0.1** backbone with a maximum sequence length of 1536 and three output labels. Tokenization uses left-side padding and truncation with the fast tokenizer. LoRA is applied to the q_proj and k_proj modules with rank $r = 16$, $\alpha = 16$, and dropout rate 0.1. The classification head is excluded from LoRA adaptation.

Training uses the AdamW8bit optimizer from bitsandbytes, with separate learning rates for the

backbone ($2 \cdot 10^{-5}$) and classification head ($2 \cdot 10^{-6}$). Parameters are grouped based on whether they belong to the head or body and whether they are subject to weight decay. A lower weight decay is applied to the body parameters. The training runs for up to 24 epochs with early stopping (patience 20), a warm-up of 10% of the steps, and evaluation every 10 steps. The best model is selected based on validation performance. On the local split this approach reaches 0.74 Macro F1 score, while on the public leaderboard the results are weaker than other masked language modeling approaches.

Submission 4 uses a **ModernBERT-large** model with a response-only input (no additional context). Unlike Submission 1, it does not use stratified batches and training is done on a single fixed random seed. The model is trained for 3 epochs, followed by per-class threshold calibration on the validation set for post-training adjustments.

This configuration achieves a Macro F1 of 0.7145 on the validation set and **0.6976** on the test set, making it our best-performing submission overall.

Submission 5 uses the same configuration as Submission 4, but consists of predictions from a second inference checkpoint corresponding to epoch 4 of the same run.

This is motivated by the use of early stopping with patience=2 during experiments, which causes training to terminate at variable points depending on the run. Since early stopping introduces non-determinism and cannot be controlled directly during inference, we submit this variant to explore whether extending inference to the subsequent saved epoch could yield marginal gains.

Submission	Macro F1	Accuracy	Ranking
Submission 1	0.6919	0.8746	26
Submission 2	0.6532	0.8423	58
Submission 3	0.6860	0.8565	27
Submission 4	0.6976	0.8675	17
Submission 5	0.6812	0.8681	31

Table 1: Leaderboard Results for Track 1 (Mistake Identification)

3.5.2 Mistake Location

Submission 1 and 2 use a **RoBERTa-large** model, trained with context input and weighted cross-entropy loss function. Submission 2 introduces a two-phase training strategy: in the first phase,

the model is trained as a binary classifier, distinguishing between "Yes" and "No" labels only; in the second phase, the model is further fine-tuned using the full three-way label set, starting from the weights learned in phase one. This curriculum-like strategy consistently outperformed the single-phase baseline, obtaining higher F1 scores on the validation set. The performance gain also persists on the public leaderboard, where it results in an approximate 3% absolute increase in F1.

Submission 3 uses a fine-tuned **microsoft/deberta-v3-large**. The input sequence length is capped at 1536 tokens. Training is conducted a batch size of 8 for up to 26 epochs with early stopping (patience 15), a warm-up phase comprising 10% of the training steps, and evaluation every 60 steps. The optimizer is AdamW8bit (bitsandbytes), using layer-wise learning rate decay (LLRD) with a decay factor of 0.9. The learning rate is set to $2 \cdot 10^{-5}$ for both the backbone and classification head.

The dataset is split using stratified group k-fold to ensure balanced class distributions between training and validation sets. Training batches are constructed using a custom `BalancedBatchSampler` that ensures balanced class representation by over-sampling minority classes and yielding samples in a round-robin fashion across classes.

This achieves the best overall result for Mistake Location, however, we find this solution to be over-engineered compared to the actual results obtained.

Submission	Macro F1	Accuracy	Ranking
Submission 1	0.5013	0.6348	44
Submission 2	0.5301	0.6826	25
Submission 3	0.5318	0.6568	24

Table 2: Leaderboard Results for Track 2 (Mistake Location)

3.5.3 Providing Guidance

Submissions 1, 2 and 4 all use a **RoBERTa-large** model trained for 4 epochs with class-weighted cross-entropy loss. Submissions 1 and 2 use a response-only input and a cosine learning rate scheduler without warm-up. Submission 2 additionally applies post-training calibration via temperature scaling and per-class threshold adjustment. Submission 4 differs by using a context input and a linear cosine scheduler with warmup ratio 0.1, while keeping the rest of the configuration un-

changed. **Submission 3** is based on the same **DeBERTa-large** model as Submission 3 for Mistake Location 3.5.2. Both submissions 2 and 3 achieve strong validation Macro F1 scores (0.58 and 0.59, respectively), but drop significantly on the test set (to **0.50** and **0.48**), suggesting a degree of overfitting to the validation distribution. Alternatively, the discrepancy may suggest a mismatch in class proportions between the dev and test sets for this metric. In contrast, Submission 4, which scores lower on validation (0.56), achieves **0.52** on the test set – a smaller drop that could indicate better generalization.

Submission	Macro F1	Accuracy	Ranking
Submission 1	0.4945	0.5398	42
Submission 2	0.5068	0.5740	35
Submission 3	0.4839	0.6025	58
Submission 4	0.5208	0.5734	23

Table 3: Leaderboard Results for Track 3 (Providing Guidance)

3.5.4 Actionability

Submission 1 uses a **ModernBERT-large** model trained for 4 epochs on full context input with standard cross-entropy loss. This serves as our starting point for the track, providing a baseline for comparing different architectures and training setups.

Submission 2 uses a **RoBERTa-large** model trained for 4 epochs, this time with context input (instead of full context) and weighted cross-entropy loss. This setup ends up performing the best in our experiments, giving us the highest test score on this track.

Submissions 3 and 4 use **DeBERTa-v3-large** with the same setup as Submission 2: context input and weighted cross-entropy loss. We switch to DeBERTa-v3-large after noticing improvements on the validation set, but the performance turns out to be lower on the test set. For Submission 3, we train for 4 epochs and initially observe promising validation results. In Submission 4, we reduce the training to 3 epochs to see if it improves generalization, but the results remain below expectations.

Submission 5 is based on the same **DeBERTa-large** model as Submission 3 for Providing Guidance and Mistake Location 3.5.2.

3.5.5 Tutor Identification

For this track, we use the tutor’s response as input, as the goal is to identify which tutor (LLM

Submission	ModernBERT	Accuracy	Ranking
Submission 1	0.6571	0.7136	23
Submission 2	0.6776	0.7214	11
Submission 3	0.6434	0.7214	33
Submission 4	0.6146	0.7098	41
Submission 5	0.6430	0.7033	34

Table 4: Leaderboard Results for Track 4 (Actionability)

Submission	Macro F1	Accuracy	Ranking
Submission 1	0.8866	0.8882	13
Submission 2	0.8794	0.8759	16
Submission 3	0.8786	0.8817	18

Table 5: Leaderboard Results for Track 5 (Tutor Identification)

or human) generates it. For all of the experiments, we use cross entropy loss, a learning rate of $1e-5$, a batch size of 8, a weight decay of 0.05 and a warmup ratio of 0.1.

Submission 1 uses a **RoBERTa-large** model trained for 4 epochs. We notice that the validation score is very close to the test score, so we use it as a starting point to decide what to try next.

Submission 2 uses a **ModernBERT-large** model trained for 5 epochs. We observe that the validation score is higher than what we obtain with RoBERTa, but when we actually submit it, the test performance is lower, which suggests that the model doesn’t generalize as well.

Submission 3 also uses **ModernBERT-large**, but trained for 4 epochs. The motivation behind this submission is to see if reducing the number of epochs helps the model generalize better to unseen data, especially after seeing a performance drop in submission 2. While the validation score is similar, the test performance is not improved, so we conclude that simply reducing the number of training epochs isn’t sufficient to improve generalization.

We notice that the model sometimes predicts the same tutor identity for multiple responses within the same dialogue, even though each tutor generates only one response per dialogue. Due to time constraints, we do not implement this refinement, although it likely leads to improvements in both accuracy and F1 scores.

4 Conclusions

Our work presents a comprehensive empirical exploration of text classification methods for the Shared Task at BEA2025. We explore a wide range

of modeling approaches – from classical machine learning methods to large-scale transformer-based models with parameter-efficient fine-tuning. Despite this diversity, we find that simple baselines can achieve good enough evaluation scores and that additional engineering using larger deep models adds less than 0.1 extra points for the Macro F1 evaluation score or accuracy.

The “To some extent” label emerges as a key source of difficulty, introducing inconsistency that complicates learning and evaluation. Our results suggest that simple models can achieve competitive performance when ambiguity is reduced, particularly under lenient evaluation settings.

Across all tracks, our models achieve competitive results, with top-10 rankings in three out of five tracks.

To the question posed in our title – *Are Simple Baselines Good Enough?* – we offer an answer in the spirit of the task itself: “To some extent”.

Limitations

Model selection for the final leaderboard is based on classification performance on a local dev split, without in-depth qualitative analysis of the classifiers or their features. We believe that such approaches in the future lead to a better understanding of why some responses are suitable and some others are not, based on so-called “reasoning” capabilities of LLMs. Furthermore, the LLM we use for prompting is a relatively weak one, and due to compute limitations, we do not explore higher-performing open source LLMs, nor closed-source systems.

Acknowledgments

This research is partially supported by the project “Romanian Hub for Artificial Intelligence - HRIA”, Smart Growth, Digitization and Financial Instruments Program, 2021-2027, MySMIS no. 334906 and partially by InstRead: Research Instruments for the Text Complexity, Simplification and Readability Assessment CNCS - UEFISCDI project number PN-IV-P2-2.1-TE-2023-2007.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav

Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. [Phi-3 technical report: A highly capable language model locally on your phone](#). *Preprint*, arXiv:2404.14219.

Anthropic. 2024. [The claude 3 model family: Opus, sonnet, haiku](#).

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. [Class-balanced loss based on effective number of samples](#). *Preprint*, arXiv:1901.05555.

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](#). *Preprint*, arXiv:2407.09136.

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, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [DeBERTa: Decoding-enhanced bert with disentangled attention](#). *Preprint*, arXiv:2006.03654.

Radu Tudor Ionescu and Andrei Madalin Butnaru. 2018. Transductive learning with string kernels for cross-domain text classification. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part III 25*, pages 484–496. Springer.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.

Ekaterina Kochmar, Kaushal Kumar 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)*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. [Focal loss for dense object detection](#). *Preprint*, arXiv:1708.02002.

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.

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](#). *Preprint*, arXiv:2305.14536.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025a. [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.

Kaushal Kumar Maurya, KV Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025b. [Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors](#). *Preprint*, arXiv:2412.09416.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. 2023. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36:76336–76369.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. [Generative representational instruction tuning](#). *Preprint*, arXiv:2402.09906.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, 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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#).

Oscar Skea, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*.

Anais 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.

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.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. [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.

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. 2024a. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). *Preprint*, arXiv:2412.13663.

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. 2024b. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). *Preprint*, arXiv:2412.13663.

A Data Distribution

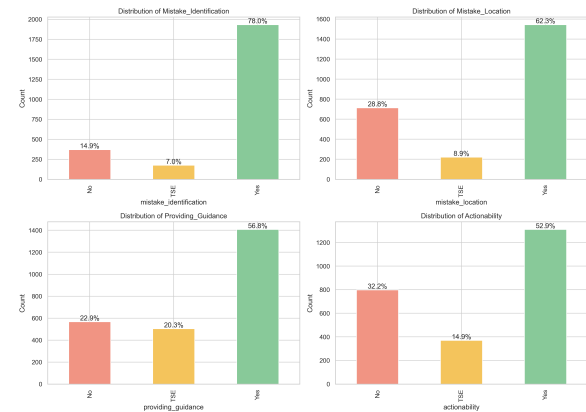


Figure 1: Label distribution

For the first four tasks, we generate stratified group splits that maintain label distribution balance while ensuring that all responses from the same dialogue (identified by *conversation_id*) are assigned to the same split. The stratification is based on the majority (mode) label per conversation.

A.1 Conversation-Level Label Distributions in Devset

Below are the counts of dialogues grouped by their majority label for each task:



Figure 2: T-SNE plots of tutor response embeddings extracted from model ModernBERT-large. We can observe on the left-hand side of each plot several tiny clusters of responses labeled with "No". These responses have similar semantic patterns (e.g., starting with "Good job!", "Good catch!", "You are absolutely correct") and share similar labels regardless of task. The Actionability task has the highest spread of negatively annotated responses.

- **Mistake Identification:**
 - Label Yes: 282 dialogues
 - Label No: 12 dialogues
 - Label To some extent: 6 dialogues
- **Mistake Location:**
 - Label Yes: 216 dialogues
 - Label No: 69 dialogues
 - Label To some extent: 15 dialogues
- **Providing Guidance:**
 - Label Yes: 204 dialogues
 - Label To some extent: 50 dialogues
 - Label No: 46 dialogues
- **Actionability:**
 - Label Yes: 185 dialogues
 - Label No: 92 dialogues
 - Label To some extent: 23 dialogues

These distributions guides stratification during splitting.

A.2 Label Distribution Within Splits

The table below shows the relative frequency of each label within the training and validation splits for each task. Proportions are expressed as percentages of total samples within each split.

Task	Label	Train (%)	Val (%)
Mist. Id.	No	14.98	14.81
	TSE	7.21	6.29
	Yes	77.81	78.90
Mist. Loc.	No	29.22	27.11
	TSE	8.85	9.04
	Yes	61.93	63.86
Prov. Guid.	No	22.63	23.79
	TSE	20.35	20.16
	Yes	57.02	56.05
Act.	No	31.85	33.54
	TSE	14.84	15.15
	Yes	53.31	51.31

Table 6: Label distribution percentages in the train and validation splits for each task

B BERT tokenization with and without prepended prompts.

Table 7: ModernBERT-large with default config ($lr=10^{-5}$, batch size=8, epochs=4, weight decay=0.01, lr_scheduler=linear, warmup_ratio=0.1, cross entropy loss) across tokenization strategies with and without prepended prompts (prompt="Rate how well the tutor identifies the student's mistake on a scale from 0 (not at all) to 2 (clearly)). Prompted variants improve performance across all metrics, likely due to the model better internalizing task-specific instruction tokens.

Strategy	Macro F1	Accuracy
no_context	0.6759	0.6188
context	0.6495	0.6020
context_full	0.6413	0.6125
prompt_no_context	0.6951	0.6609
prompt_context	0.6799	0.6493
prompt_context_full	0.6740	0.6382

C Zero-shot and Few-shot Prompting Approach

We evaluate the Mistake Identification task using zero-shot and few-shot prompting with **mistralai/Mistral-7B-Instruct-v0.2**, under greedy decoding. All scores are reported on our validation split.

In the **zero-shot setting**, a simple prompt achieves a Macro F1 of 0.419, but tends to over-predict "To some extent". Adding label definitions reduces performance (F1 drops to 0.367), and

prompting with “think step by step” introduces some invalid outputs and we decide not to invest effort into resolving this behaviour. Introducing a soft constraint, asking the model to avoid predicting “To some extent” unless clearly justified, reduces overprediction (from 154 to 40 on validation split) and preserves performance (F1 0.411), with a refined version reaching 0.421.

In the **few-shot setup**, we retrieve three diverse training examples using embeddings from **all-mpnet-base-v2** (bi-encoder) and rerank them with the cross-encoder **cross-encoder/ms-marco-MiniLM-L-6-v2**. This setup achieves 0.392 Macro F1, with frequent “To some extent” predictions. Simplifying retrieval increases these predictions without improving performance. Adding label definitions and the same constraint improves F1 to 0.452 and reduces overprediction.

We do not invest further effort into optimizing this approach, as performance remains well below our logistic regression baseline.

This aligns with findings from [Maurya et al. \(2025b\)](#), who report that LLM-based evaluators correlate poorly with human judgments on pedagogical tasks.

C.1 Base prompt

Task: You are an expert tutor evaluator. Label whether the tutor identifies the student’s mistake. There are 3 possible labels:

- Yes
- To some extent
- No

Provide only the label.

C.2 With label definitions

These are added after listing labels and before the instruction to provide only the label:

Label definitions:

- Yes: The tutor clearly identifies and addresses the mistake.
- To some extent: The tutor hints at or partially recognizes the mistake, but not clearly.
- No: The tutor does not identify or acknowledge the mistake.

C.3 Anti-"To some extent" constraints

We experiment with two variants of constraints. These are added after listing labels and before the instruction to provide only the label.

1: Avoid choosing "To some extent" unless it is

clearly not a full "Yes" or a full "No".

2: Use "To some extent" only when the tutor’s response ****clearly shows partial understanding**** – not as a fallback when unsure.

C.4 Final Prompt Composition

In the zero-shot setting, this is appended after the instruction:

```
### Student: student
### Tutor: response
### Label:
```

In the few-shot setting, this is appended after the instruction:

```
### Example i:
Student: student
Tutor: response
Label:
```

```
### Now classify:
```

```
Student: student
Tutor: response
Label:
```

Each few-shot prompt includes three examples (one per class) retrieved using a combination of bi-encoder similarity and cross-encoder reranking. Cosine similarity is computed over the concatenation of *last student utterance + tutor response*.

C.5 Inference Configuration

- Model & Tokenzier: mistralai/Mistral-7B-Instruct-v0.2
- Decoding: Greedy (do_sample=False)
- Max tokens: 5
- Quantization: 4-bit NF4 (bitsandbytes)

D Hyperparameters and Training Configurations

D.1 BERT-like encoders

Mistake Identification, Submission 1:

- Learning rate: 1e-5
- Weight decay: 0.01
- Scheduler: cosine learning rate scheduler (no warmup)
- Epochs: 4
- Batch size: 8
- 5 different training seeds

Ensembling: We train five models (one per seed) and average the logits at inference time.

Post-training calibration: After ensembling, we apply temperature scaling ($T = 1.049$) and per-class threshold tuning using validation performance. Since we use a threshold-based override strategy, we only tune thresholds for the "Yes" and "To some extent" classes. The "No" class is treated as the default fallback when neither of the other logits pass their respective thresholds. Final thresholds:

- Yes: 0.4429
- To some extent: 0.3776
- No: default threshold

Mistake Identification, Submission 4:

- Learning rate: $2e-5$
- Weight decay: 0.05
- Scheduler: cosine learning rate scheduler with 10% warmup
- Epochs: 3
- Batch size: 8
- Loss: Focal loss with $\gamma = 1.3$, class-balanced $\alpha = [0.9216, 1.7772, 0.3012]$

Post-training calibration: Temperature scaling with $T = 1.0$, and threshold override strategy using:

- Yes: 0.63
- To some extent: 0.22
- No: default fallback

To select the optimal input format and loss function, we conduct multiple runs using different configurations and evaluate them using Macro F1 on the validation set. This selection procedure is applied systematically to almost all submissions.

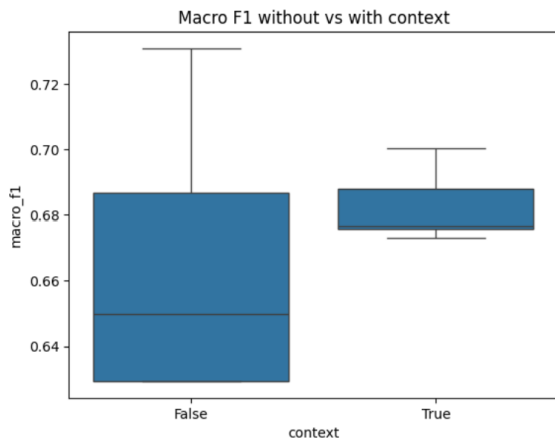


Figure 3: Macro F1 scores with and without context across seeds

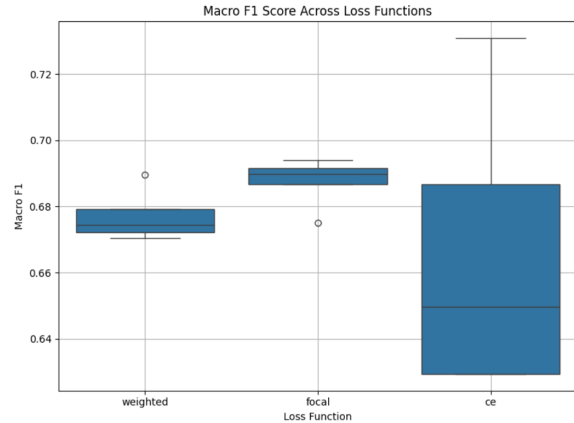


Figure 4: Macro F1 score comparison for loss functions across seeds

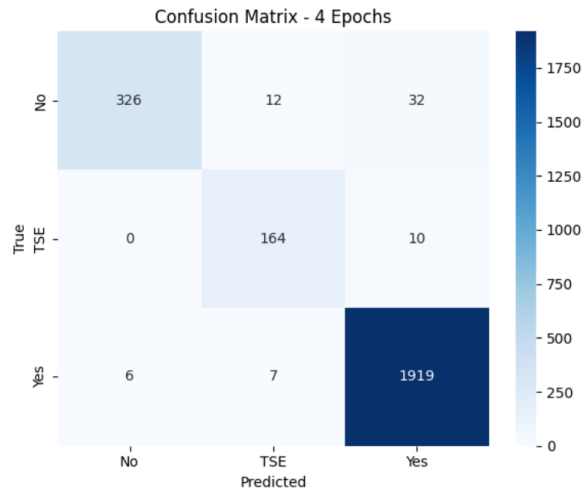


Figure 5: Confusion matrix on the full dev set after training, ensembling, and calibration, just before generating test predictions

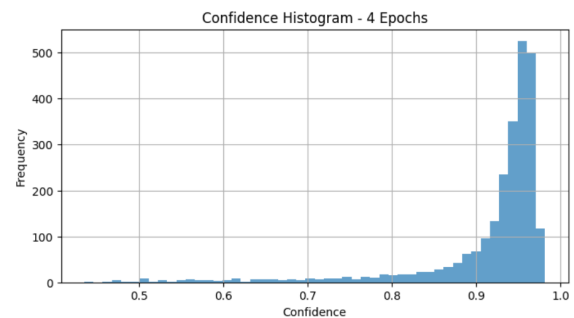


Figure 6: Distribution of prediction confidences (maximum softmax probability) on the dev set, after ensembling and temperature scaling

Mistake Identification, Submission 5: Same as Submission 4, except trained for 4 epochs instead of 3.

E Validation Performance Tables

Model	Track	Macro F1	Accuracy
Logistic Regression	Track 1	0.6318	0.8499
Logistic Regression	Track 2	0.5167	0.6647
Logistic Regression	Track 3	0.4947	0.5665
Logistic Regression	Track 4	0.5547	0.6384
Logistic Regression	Track 5	0.7455	0.7289
XGBoost	Track 1	0.6254	0.8803
XGBoost	Track 2	0.4671	0.7329
XGBoost	Track 3	0.4775	0.6190
XGBoost	Track 4	0.5374	0.6869
XGBoost	Track 5	0.7758	0.7731
XGB + LightGBM	Track 1	0.6230	0.8783
XGB + LightGBM	Track 2	0.4734	0.7309
XGB + LightGBM	Track 3	0.4584	0.6230
XGB + LightGBM	Track 4	0.5291	0.6869
XGB + LightGBM	Track 5	0.7846	0.7892
Spectrum Kernel	Track 1	0.6346	0.8844
Spectrum Kernel	Track 2	0.4728	0.7430
Spectrum Kernel	Track 3	0.4410	0.6351
Spectrum Kernel	Track 4	0.5490	0.7212
Spectrum Kernel	Track 5	0.8186	0.8092

Table 8: Exact evaluation on the validation set using minimal preprocessing and no fine-tuning.

Note: Tree-based models perform particularly well on Mistake Location (Track 2) and Providing Guidance (Track 3), achieving Macro F1 scores of 0.541 and 0.542 respectively – comparable to BERT-like models on these tracks – when optimized via randomized search over standard hyperparameter grids.

Task	Model	Metric	Val	LB
MI	String Kernel	Acc.	0.9391	0.9541
	String Kernel	F1	0.8597	0.9185
ML	String Kernel	Acc.	0.8233	0.8630
	String Kernel	F1	0.7363	0.8404
PG	XGBoost	Acc.	0.8185	0.8222
	XGBoost	F1	0.6919	0.7860
AC	String Kernel	Acc.	0.8525	0.8940
	String Kernel	F1	0.8289	0.8659

Table 9: Comparison between the scores obtained with traditional machine learning models on validation split and the best public leaderboard results (LB), for each task, under lenient evaluation.

Track	Macro F1	Accuracy
Track 1		
Submission 1	0.9054	0.9463
Submission 2	0.8675	0.9250
Submission 3	0.8907	0.9392
Submission 4	0.8959	0.9405
Submission 5	0.8917	0.9399
Track 2		
Submission 1	0.7406	0.7666
Submission 2	0.7506	0.7886
Submission 3	0.7558	0.8009
Track 3		
Submission 1	0.7303	0.7854
Submission 2	0.7228	0.7725
Submission 3	0.6730	0.7666
Submission 4	0.7171	0.7770
Track 4		
Submission 1	0.8229	0.8571
Submission 2	0.8302	0.8565
Submission 3	0.8250	0.8500
Submission 4	0.8370	0.8655
Submission 5	0.8152	0.8487

Table 10: Lenient evaluation of our submissions on the public leaderboard.

E.1 Transformer Embeddings

E.1.1 Frozen ModernBERT-large Embeddings + Linear Classifier

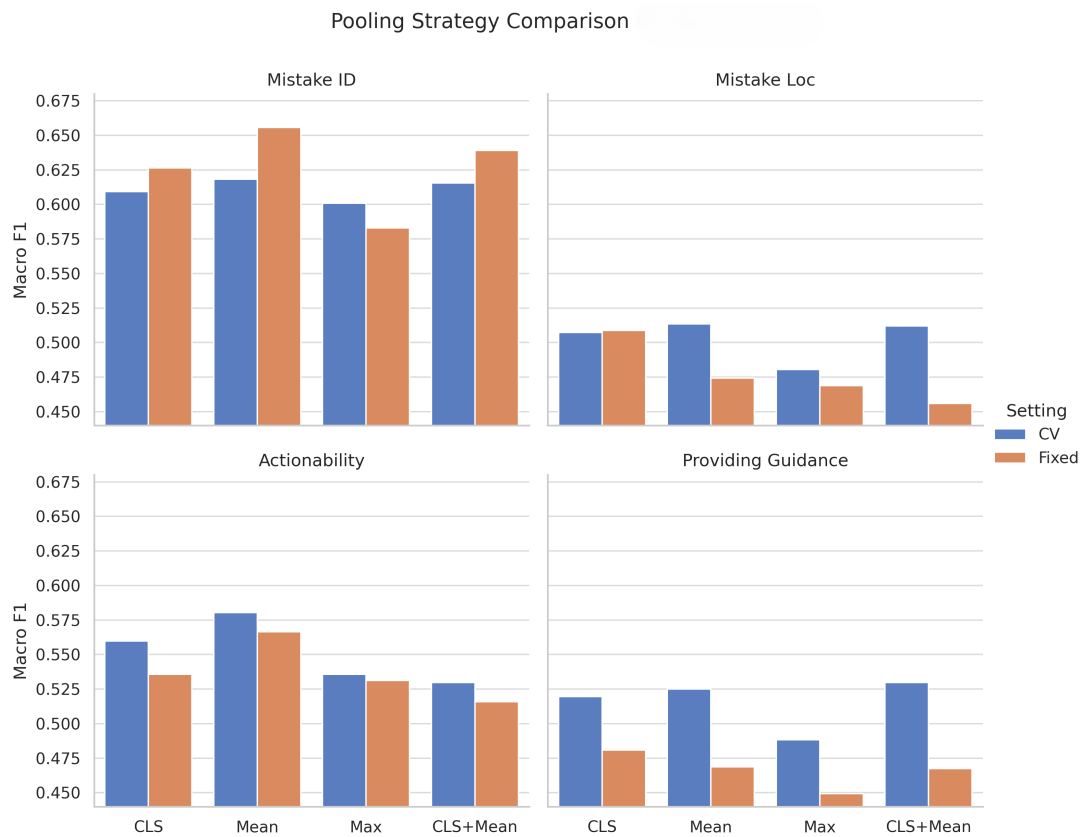


Figure 7: Pooling strategy comparison across 2 split strategies; CV - *StratifiedKFold(n_splits=5, shuffle=True, random_state=42)*, Fixed - 80/20 train/validation splits as described in A

Evaluation is conducted using two split strategies: stratified 5-fold cross-validation and a fixed 80/20 train/validation split.

Since only the tutor response is used for embedding extraction, the folds are not grouped by *conversation_id*, which may partially explain why scores are higher under cross-validation for three out of four tasks (not grouping by *conversation_id* can lead to leakage across folds by having similar responses from the same dialogue appearing in both train and validation).

E.1.2 Various Model layers + Linear Classifier

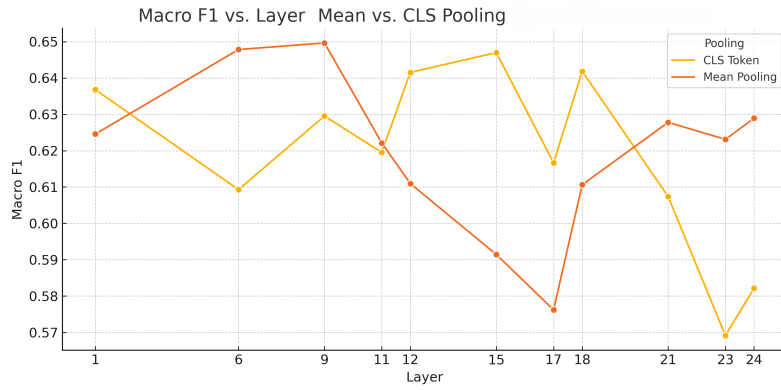


Figure 8: Pooling strategies from BERT model comparison across layers for Mistake Identification. Mean pooling appears to perform better on early and late layers.

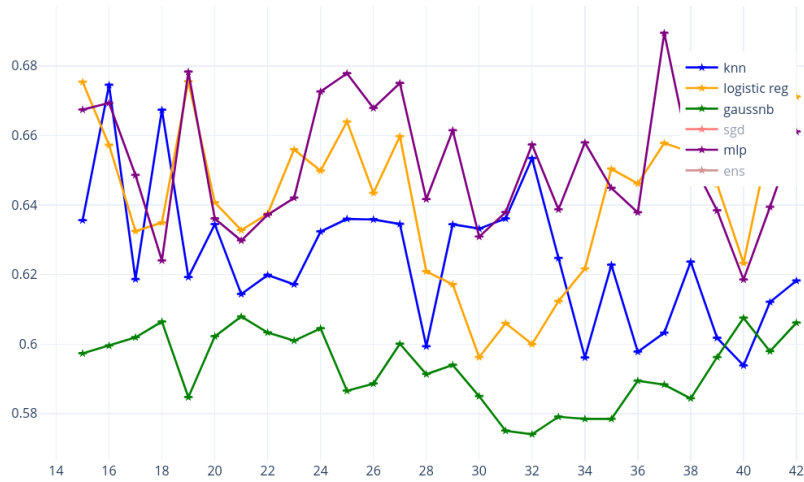


Figure 9: Comparison of different predictors using embeddings from middle layers of GritLM. There are no striking differences between MLP and Logistic Regression classifiers. Furthermore, there is no pattern showing that late layers are performing significantly better than early layers using embeddings from GritLM. The k-nearest neighbors (knn) and Gaussian Naive Bayes models perform weaker.

F Misclassified Examples

Here we provide several misclassified examples from our best predictor on Mistake Identification.

Example of chat misclassified as "No" for Mistake Identification; True label: "Yes"

Tutor: Hi, could you please provide a step-by-step solution for the question below? The question is: Sophia is thinking of taking a road trip in her car, and would like to know how far she can drive on a single tank of gas. She has traveled 100 miles since last filling her tank, and she needed to put in 4 gallons of gas to fill it up again. The owner's manual for her car says that her tank holds 12 gallons of gas. How many miles can Sophia drive on a single tank of gas?

Student: Sophia used 4 out of the 12 gallons of gas in her tank, so there are $12 - 4 = 8$ gallons of gas left in the tank. If Sophia can drive 100 miles on 4 gallons of gas, then she can drive $100/4 = 25$ miles per gallon. Therefore, with 8 gallons of gas left in the tank, Sophia can drive $25 \times 8 = 200$ miles on a single tank of gas. 200

Tutor: talk me through your solution

Student: Hi I started by figuring out how many gallons of gas were left in Sophia's tank. I knew that she had used 4 gallons of gas to fill it up so I subtracted 4 from 12 to get 8 gallons of gas left in the tank. Then I figured out how many miles Sophia could drive on a single gallon of gas. I knew that she had driven 100 miles on 4 gallons of gas so I divided 100 by 4 to get 25 miles per gallon. Finally I multiplied 25 by 8 to get 200 miles which is how many miles Sophia can drive on a single tank of gas.

Response: Great job! To further enhance your understanding, let's explore more examples and practice problems related to this concept

Interpretation: Student's math is correct, $8 \text{ gallons} \times 25 \text{ miles/gallon} = 200 \text{ miles}$, but the answer should be 300 since Sophia can drive $12 \text{ gallons} \times 25 \text{ miles/gallon} = 300 \text{ miles}$. The ground truth label is "Yes", implying that the tutor correctly identified the mistake. Language in the tutor's response is affirming and does not correct the student's incorrect logic, so the model likely sees *Great job!* .. and looking only at the tutor response, assumes no mistake is identified.

Example of chat misclassified as "Yes" for Mistake Identification; True label: "No"

Tutor: Hi, could you please provide a step-by-step solution for the question below? The question is: A customer's loyalty card at a store gives them rewards of \$1 off their next purchase for every \$20 they spend. Their last shopping trip, they spent \$80. This shopping trip, they spent \$43, used their rewards, and applied a coupon that took twice the amount of rewards off the price. How many dollars did the customer pay on this shopping trip?

Student: The customer earned $80/20 = 4$ rewards on their last shopping trip.

They had $4 \times \$1 = \4 in rewards to use this shopping trip. The coupon took twice the amount of rewards off, so it took $\$4 \times 2 = \8 off the price. The original price of the items was $\$43 + \$4 = \$47$. After the coupon and rewards were applied, the customer paid $\$47 - \$8 = \$39$.

Tutor: could you please walk me through your solution?

Student: Sure. I started by calculating the rewards the customer earned on their last shopping trip. They spent \$80 so they earned $80/20 = 4$ rewards. Then I multiplied 4 rewards by \$1 to get the total amount of rewards which was \$4. I then applied the coupon that took twice the amount of rewards off the price so it took $\$4 \times 2 = \8 off the price. The original price of the items was $\$43 + \$4 = \$47$. After the coupon and rewards were applied the customer paid $\$47 - \$8 = \$39$.

Tutor: If i purchase a bicycle worth \$100 and have a coupon of \$24 and rewards worth \$6 how much do i actually pay for it?

Student: You would pay $\$100 - \$24 - \$6 = \70 .

Response: That's close, but remember to apply the coupon first since it has a higher value and can reduce the price before the rewards. So, you would pay $\$100 - \$24 = \$76$ and then subtract the rewards, which is $\$76 - \$6 = \$70$, not the other way around.

Interpretation: Student's answer of 70 is mathematically correct. The tutor's reasoning about the order

is irrelevant, since subtracting fixed dollar amounts is order-invariant. The tutor gives a correct final answer but justifies it incorrectly, which should count as a failure to identify a student mistake (correct answer by chance). The model likely sees the phrase: *"That's close, but remember..."* which resembles correction, and may interpret it as mistake identification, when in reality, no actual error is identified.

These misclassifications highlight the limitations of response-only models for Mistake Identification. In both cases, the tutor's language is superficially corrective or affirming, but semantically misaligned with the student's actual performance.

G Data observations

During our experiments, we identify some cases of label inconsistencies, especially between label "Yes" and "To some extent".

Label disagreement on nearly identical tutor responses when the student makes a mistake:

1. Mistake Identification

Sample A (Label: Yes)

Conversation ID: 413876945_2

History:

Tutor: Please recheck your answer once.

Tutor: What is the product of this equation?

Tutor: $7 \cdot 60 = ?$

Student: 520

Response: Please recheck your answer.

Sample B (Label: To some extent)

Conversation ID: 294929858

History:

Tutor: We are asked to find the quotient of 72 and 6.

Tutor: Let n be the number of doughnuts.

Tutor: Are you working on this question?

Student: 0.0833333

Response: Please recheck your answer.

2. Actionability

Sample A (Label: Yes)

Conversation ID: 29892262_1

History:

Tutor: Happy to work with you today!

Tutor: Today, let us review and master the lesson "Rounding Whole Numbers."

Tutor: Which is the greatest number among 5 and 3?

Student: 4

Response: That's a good try, but remember, we're looking for the greatest number between 5 and 3. Let's try again!

Sample B (Label: To some extent)

Conversation ID: 29892262_2

History:

Tutor: Happy to work with you today!

Tutor: Today, let us review and master the lesson "Rounding Whole Numbers."

Tutor: Which is the greatest number among 5 and 3?

Student: 4

Response: That's a good try, but remember we're looking for the greatest number between 5 and 3. Let's try again.

H Team rankings

Metric	Mistake ID	Mistake Loc	Prov. Guid.	Actionability	Tutor ID
Strict Macro F1	8	12	13	7	6
Strict Accuracy	4	23	21	7	6
Lenient Macro F1	8	12	11	10	–
Lenient Accuracy	7	10	10	10	–

Table 11: Archaeology's official team ranking per task under all evaluation metrics.

NLIP at BEA 2025 Shared Task: Evaluation of Pedagogical Ability of AI Tutors

Trishita Saha¹ Shrenik Ganguli¹ Maunendra Sankar Desarkar¹

¹Natural Language and Information Processing Lab (NLIP)

¹Indian Institute of Technology Hyderabad

¹Hyderabad, India

trishita51@gmail.com cs23mtech14014@iith.ac.in maunendra@cse.iith.ac.in

Abstract

This paper presents our system submission to the **Building Educational Applications (BEA) 2025 Shared Task** on Pedagogical Ability Assessment of AI-powered Tutors. The task evaluates multiple dimensions of AI tutor responses within student-teacher educational dialogues, including mistake identification, mistake location, providing guidance, and actionability. Our approach leverages transformer-based models (Vaswani et al., 2017), primarily **DeBERTa** and **RoBERTa**, and incorporates *ordinal regression*, *threshold tuning*, *oversampling*, and *multi-task learning*. Our best-performing systems are capable of assessing tutor response quality across all tracks. This highlights the effectiveness of tailored transformer architectures and pedagogically motivated training strategies for AI tutor evaluation.

1 Introduction

Nowadays, AI systems can support sophisticated educational dialogues thanks to recent advancements in large language models (LLMs), which suggests they could be used as tutors in real-world learning settings. Although models like GPT-4 (Achiam et al., 2023) and its successors are effective at producing coherent text (Brown et al., 2020), their capacity to carry out pedagogical tasks, like identifying misconceptions, assisting students, or providing helpful criticism, is still poorly understood and requires focused assessment (Tack and Piech, 2022; Daheim et al., 2024).

Our work in the BEA 2025 Shared Task (Kochmar et al., 2025) aims to address this gap by systematically assessing tutor responses along four dimensions: *mistake identification*, *mistake location*, *guidance provision*, and *actionability*. Built upon a unified taxonomy (Maurya et al., 2025), the task draws on annotated exchanges from the *Math-Dial* (Macina et al., 2023) and *Bridge* (Wang et al., 2024) datasets to create a benchmark for evaluating

AI tutor behaviour.

This paper describes our submissions to all four tracks of the shared task. Our systems leverage transformer-based models, primarily DeBERTa (He et al., 2021) and RoBERTa (Liu et al., 2019b), with tailored architectures for each subtask. Binary and multiclass classification are applied for Tracks 1 and 3, ordinal regression with threshold tuning for Track 2, and ensemble classifiers for Track 4 that combine neural features with symbolic representations (e.g., TF-IDF and entailment scores).

Several techniques have been employed to rectify class imbalance and take advantage of cross-task signals:

- Utilizing resampling strategies that have been proven successful in analogous domains to **oversample** under-represented labels, such as "To some extent" (Saha et al., 2023; Chawla et al., 2002).
- Adopting a standard **multi-task learning** architecture comprising a shared encoder and task-specific output layers, following established practices in similar domains (Liu et al., 2019a).
- **Ensemble classifiers** (using stacking) that improve actionability prediction by combining deep contextual representations with conventional NLP features.

In every track, our models regularly outperform baseline systems. For instance, the multi-task DeBERTa model attains a **Macro F1** of **0.8065** on Track 4 (strict setting) and **0.8809** on Track 1. A regression-based strategy with threshold adjustment yields the most remarkable results for Track 2, with a **Macro F1** of **0.58**. Ensemble classifiers on Track 4 considerably increase robustness; the lenient **Macro F1** is greater than **0.861**.

These findings show that using well-designed methods to balance and combine training data and

Conversation History	Tutor	Tutor Response	MI	ML	PG	ACT
Tutor: You earn one point for your good beginning. Tutor: That was a good try. Tutor: What is the value of 3 ³ ? Student: 9	GPT4	That's a good effort, but actually, 3 ³ means 3 multiplied by itself three times, which equals 27.	✓	✓	✓	✗
	Gemini	Remember, 3 to the power of 3 means 3 multiplied by itself three times: 3 × 3 × 3.	✓	✓	✓	✓

Figure 1: Example of Dataset where MI - Mistake Identification, ML - Mistake Location, PG - Providing Guidance, ACT - Actionability.

carefully fine-tuning transformer models can help assess AI teachers to check if they speak fluently and give useful educational feedback (Wollny et al., 2021).

2 Shared Task Structure

Development phase: A dataset consisting of **2476** annotated tutor responses drawn from **300** dialogues was provided. Each response was labeled across four pedagogical dimensions — *mistake identification*, *mistake location*, *guidance provision*, and *actionability*, according to the taxonomy of Maurya et al. (2025). A **80%–20%** stratified split was performed to create training and test sets (**1980** and **496** responses, respectively), preserving class label proportions across all tracks. This stratified sampling ensured balance across both frequent and rare labels such as “Yes”, “No”, and “To some extent”.

Table 1 summarizes the distribution of classes across the four tracks before and after splitting. It is observed *considerable class imbalance* in all tracks, particularly in **Track 1**, where over **75%** of the responses are labeled “Yes”. The “To some extent” category appears in only **7%** of cases. While **Track 2** shows slightly improved balance, it still *underrepresents* the “To some extent” label. **Track 3** is relatively more balanced, with “To some extent” making up over **20%** of the examples. **Track 4** has the most even distribution, with “No” (**32.3%**), “Yes” (**52.8%**), and “To some extent” (**14.9%**) labels appearing at meaningful frequencies. This variation in *class balance* prompted us to use **stratified sampling**, experimenting with a range of models (Section 3) and evaluating them using metrics — Accuracy and Macro-F1 under both strict and lenient conditions. The top-performing models were selected for final submission.

In addition to quantitative analysis, it is examined how different tutors address the four pedagogical dimensions using concrete examples. Figure 1 illustrates a representative case comparing GPT-4

and Gemini on an evaluation error. Both systems successfully identify the student’s mistake, locate it, and provide guidance; however, only Gemini (Reid et al., 2024) offers *actionable feedback* with explicit instructions to the student on how to correct their answer, whereas GPT-4 omits this crucial step. This highlights the importance of distinguishing between **basic guidance** and **true actionability** in tutor responses and underscores the nuanced challenges in reliably annotating and modelling these dimensions.

Test phase: In the final evaluation phase, an unlabeled test set comprising **1547** tutor responses from **191** dialogues was given. Predictions from our best models were submitted for each of the four tracks, and performance was assessed using the same evaluation metrics (Section 4). To aid interpretation, **LIME** (Local Interpretable Model-agnostic Explanations) on selected outputs was applied to visualize influential tokens (see Figure 6a), offering insights into the model behaviour.

Track	Split	No	Yes	To some extent
Track 1	All	370 (15.0%)	1932 (78.0%)	174 (7.0%)
	Train	296 (15.0%)	1545 (78.0%)	139 (7.0%)
	Test	74 (15.0%)	387 (78.0%)	35 (7.0%)
Track 2	All	709 (28.6%)	1552 (62.7%)	215 (8.7%)
	Train	567 (28.6%)	1241 (62.7%)	172 (8.7%)
	Test	142 (28.6%)	311 (62.7%)	43 (8.7%)
Track 3	All	566 (22.9%)	1407 (56.8%)	503 (20.3%)
	Train	453 (22.9%)	1125 (56.8%)	402 (20.3%)
	Test	113 (22.9%)	282 (56.8%)	101 (20.3%)
Track 4	All	800 (32.3%)	1307 (52.8%)	369 (14.9%)
	Train	640 (32.3%)	1045 (52.8%)	295 (14.9%)
	Test	160 (32.3%)	262 (52.8%)	74 (14.9%)

Table 1: Class-wise distribution of tutor responses across all four tracks (Train = 80%, Test = 20%). Percentages indicate class proportions within each split.

3 Tracks Descriptions and Methodology

- **Track 1: Mistake Identification** - Since student mistakes are present in every dialogue, a good tutor must identify them by reflecting *student understanding* (Tack and Piech, 2022) and *correctness* (Macina et al., 2023). A **RoBERTa-base** model is fine-tuned for 3-way sequence classification that detects the presence of error in tutor responses and provides dialogue context. The cross-entropy loss function is used. Predictions at the end are all converted to categorical labels. The RoBERTa model is used for this task as it captures deep contextual representations from large-scale pretraining on diverse data, which enables it

to effectively understand subtle distinctions in input, making it the best-performing model for identifying sentence-level mistakes.

- **Track 2: Mistake Location** - A good tutor response should point to the error location and explain it clearly to help the student improve, capturing *targetedness* as defined by (Daheim et al., 2024). It is a fine-grained task that requires identifying the exact phrase causing the error and not just flagging the whole sentence. An *ordinal regression* approach is implemented by fine-tuning a pretrained **DeBERTa-v3-base** transformer encoder. The mapping of class labels to ordinal values was done as follows: Class “No” was mapped to 0, Class “To Some Extent” was mapped to 1, and Class “Yes” was mapped to 2. RandomOverSampler has been used to address class imbalance, by increasing the number of samples in the underrepresented *To some extent* class to equal the number of samples in class *No*. The model architecture consists of a DeBERTa encoder followed by a dropout layer and a linear regression head that outputs a continuous scalar. During training, optimization is done through mean squared error loss between predicted scalar outputs and ordinal labels. Focal and Cross entropy loss underperformed compared to the Mean Squared Error loss. Consequently, the results of these losses are not reported in the paper. Discretization was performed for continuous predictions into ordinal classes through predefined thresholds during inference, and then inverse mapping to the original categorical labels was done. The enhanced positional encoding and disentangled attention mechanism of the DeBERTa model allows it to locate contextual clues and word-level dependencies, making it highly effective and the best performing for this track.
- **Track 3: Providing Guidance** - A good tutor response should offer helpful guidance, like hints without explicitly giving away the solution, aligning with *helping a student* (Tack and Piech, 2022) and *usefulness* (Wang et al., 2024). A **RoBERTa-base** model is fine-tuned on the final input sequence. Encoding of target labels via label encoding into three classes is done. Model architecture comprises a RoBERTa encoder, dropout, and a linear clas-

sification head. The *cross-entropy* loss function and a cosine learning rate with 60 epochs are used. Mixed precision training, along with gradient scaling and gradient clipping, has been employed to improve efficiency. The nature of deep contextual understanding and robust pretraining enables the RoBERTa model to generate contextually relevant and accurate suggestions, making it the best-performing model for offering meaningful guidance on corrections.

- **Track 4: Actionability** - A good tutor response should clearly mention the next step for the student avoiding dead ends—capturing *actionability* as defined by (Daheim et al., 2024). A **stacked ensemble** model combining traditional *TF-IDF* with contextual embeddings from *RoBERTa* is developed. *TF-IDF* vectorizes the tutor responses initially and the tokenized input is passed into a pretrained RoBERTa-base model which is fine-tuned for sequence classification having three output classes. Probability distributions from RoBERTa are then concatenated with *TF-IDF* vectors forming a comprehensive feature set. On this, training is performed by *Extra Trees* ensemble classifier. Final model evaluation is done using accuracy and macro F1 score, demonstrating the effectiveness of classical integration. A stacking ensemble approach using *TF-IDF*, *RoBERTa* and *Extra Trees* is used for this track because it combines the strengths of deep contextual embeddings, lexical features and robust non-linear classification to effectively capture both semantic and surface-level cues, leading to superior actionability predictions.

3.1 Multitask Approach

A multitask RoBERTa-based model is utilized to jointly predict four classification tasks: *Mistake Identification*, *Mistake Location*, *Providing Guidance*, and *Actionability*. The model shares frozen embeddings and partially frozen encoder layers, which are followed by task-specific classification heads. The total loss is a weighted sum of cross-entropy losses across tasks:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^4 \lambda_i \cdot \text{CE}(\hat{y}_i, y_i)$$

TRACK 1: Mistake Identification					
Model / Approach	Strict Macro F1 (↑)	Strict Accuracy (↑)	Lenient Macro F1 (↑)	Lenient Accuracy (↑)	Rank
DeBERTa	0.607	0.827	0.849	0.919	94
DistilRoBERTa	0.621	0.818	0.823	0.892	84
BERT	0.626	0.846	0.861	0.928	80
RoBERTa	0.639	0.823	0.837	0.903	67
Multitask (RoBERTa, 40 epochs)	0.644	0.855	0.872	0.926	63
TRACK 2: Mistake Location					
Model / Approach	Strict Macro F1 (↑)	Strict Accuracy (↑)	Lenient Macro F1 (↑)	Lenient Accuracy (↑)	Rank
DeBERTa	0.532	0.688	0.749	0.795	23
SpanBERT	0.477	0.601	0.708	0.751	63
RoBERTa	0.495	0.624	0.712	0.749	48
BERT	0.508	0.654	0.712	0.765	42
ModernBERT	0.486	0.599	0.702	0.767	56
TRACK 3: Providing Guidance					
Model / Approach	Strict Macro F1 (↑)	Strict Accuracy (↑)	Lenient Macro F1 (↑)	Lenient Accuracy (↑)	Rank
DeBERTa	0.481	0.587	0.685	0.733	60
RoBERTa	0.489	0.603	0.693	0.765	52
Multitask (RoBERTa, 25 epochs)	0.460	0.658	0.723	0.789	79
Multitask (RoBERTa, 40 epochs)	0.465	0.658	0.722	0.789	78
TRACK 4: Actionability					
Model / Approach	Strict Macro F1 (↑)	Strict Accuracy (↑)	Lenient Macro F1 (↑)	Lenient Accuracy (↑)	Rank
Stacking (BERT + Extra Trees)	0.599	0.677	0.815	0.845	47
Stacking (RoBERTa + Extra Trees)	0.606	0.689	0.821	0.847	45
DeBERTa (Last Layer)	0.589	0.676	0.810	0.846	53
DeBERTa (Second Last Layer)	0.476	0.564	0.657	0.661	75
Multitask (RoBERTa, 40 epochs)	0.579	0.688	0.815	0.839	55

Table 2: Performance metrics (macro F1 and accuracy) across Tracks 1–4 using strict and lenient evaluation settings. Strict evaluation best values are highlighted in blue and Lenient evaluation best values in green.

where λ_i are task specific weights, \hat{y}_i are the predicted logits and y_i are the corresponding ground truth labels. Hyperparameters such as learning rate, dropout, and task weights are optimized using the Optuna framework. Evaluation uses macro-F1 and lenient accuracy across tracks.

4 Evaluation and Results

Tracks 1-4 are evaluated using macro F1 as the primary metric and accuracy as the secondary metric. The two evaluation formats used are as follows:

- *Strict evaluation*: A total of three classes are present - "Yes", "To some extent", "No". Based on these classes, models are assessed.
- *Lenient evaluation*: "Yes" and "To some extent" are merged into a single class that simplifies the task into a binary classification ("Yes + To some extent" vs "No").

The results obtained here (shown in Table 2) are on the test dataset. For results obtained on the development dataset refer to the Appendix (Section A).

4.1 Track 1: Mistake Identification

Multitask RoBERTa models, especially the one fine-tuned for 40 epochs, outperformed all other models with a strict macro F1 of **0.6438** and accuracy of **0.8546**, highlighting the benefit of extended

domain-specific training. BERT maintained strong baseline performance (**F1: 0.6262**), whereas DistilRoBERTa exhibited lower performance due to its compact architecture, trading off accuracy for efficiency.

4.2 Track 2: Mistake Location

DeBERTa achieved the best performance (**F1: 0.5319, accuracy: 0.6878**), likely due to its strong token-level contextual understanding. RoBERTa and BERT were competitive but fell slightly behind. The overall lower scores across models reflect the increased difficulty in precisely locating mistakes, which demands deeper syntactic and semantic analysis.

4.3 Track 3: Providing Guidance

In this track, models had to suggest appropriate corrections and identify errors. RoBERTa-based models again led, with strict F1 around **0.48** and best accuracy at **0.6580** by the Multitask variant. While fine-tuned RoBERTa models balanced precision and recall effectively, Multitask models underperformed.

4.4 Track 4: Actionability

Our approach to this track combined surface-level lexical and deep contextual features to identify

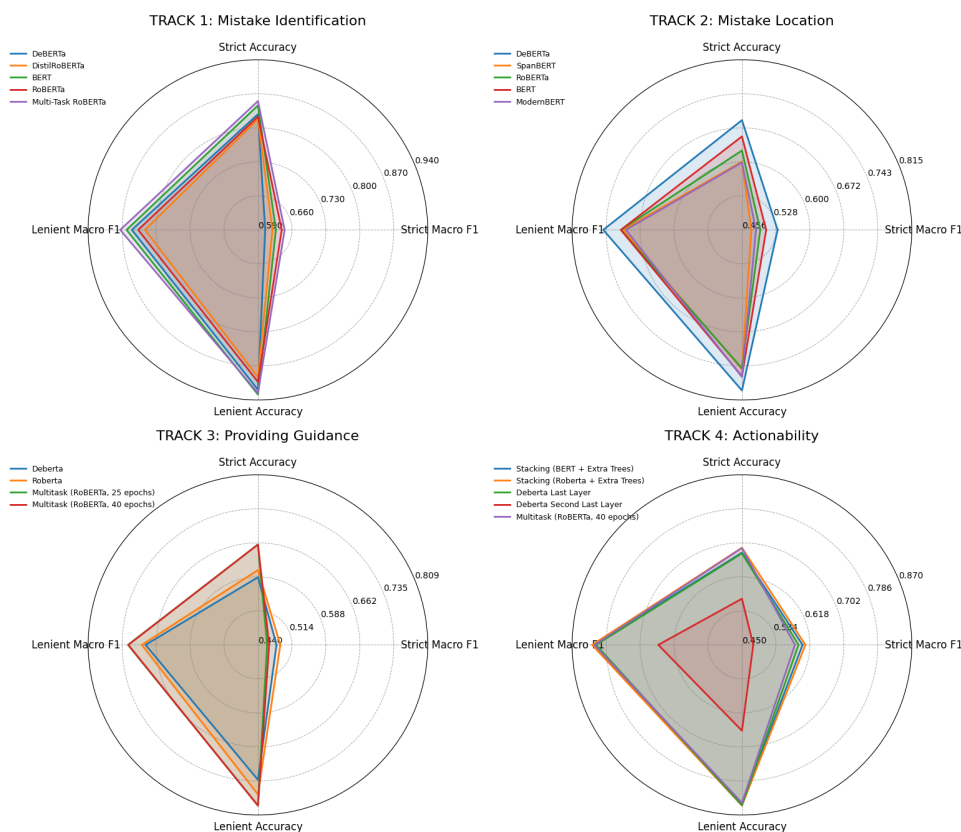


Figure 2: Radar plots comparing model performance across the four shared task tracks. The top row shows results for Track 1 (Mistake Identification, left) and Track 2 (Mistake Location, right). The bottom row presents Track 3 (Providing Guidance, left) and Track 4 (Actionability, right). Each plot visualizes four evaluation metrics: Strict Accuracy, Strict Macro F1, Lenient Accuracy, and Lenient Macro F1, as reported in Table 2. These radar charts highlight the relative strengths and weaknesses of different modeling approaches across the four tracks.

actionable feedback. A stacked ensemble model using TF-IDF features, RoBERTa embeddings, and an Extra Trees classifier achieved the highest results (**F1: 0.6055, accuracy: 0.6897**), outperforming standalone models like BERT and DeBERTa. The Multitask RoBERTa model showed similar accuracy but slightly lower F1, suggesting ensemble methods can offer better generalization by leveraging multiple feature types.

5 Analysis and Discussion

Various model strengths have been seen across the four tracks. **Fine-tuned RoBERTa** with 40 epochs gave the best result after Multitask (RoBERTa) for *Mistake Identification*, while **DeBERTa** did better in *Mistake Location* due to better token-level context. **RoBERTa** also performed best in *Providing Guidance*. For *Actionability*, a **stacking ensemble model of TF-IDF, RoBERTa, and Extra Trees** outperformed transformers alone, as it allowed the value of combining both semantic and lexical features. Real-world classification

challenges are clearly visible by the gap between the strict and lenient metrics. Overall, fine-tuned transformers showed quite promising results, but stacking ensemble approaches are crucial for complex tasks.

Figure 2 provides a comparative view of model performances across the four shared task tracks using four evaluation metrics — *Strict Accuracy*, *Strict Macro F1*, *Lenient Accuracy*, and *Lenient Macro F1*, all derived from leaderboard submissions on the test set (refer Table 2). For Track 1, **multi-task RoBERTa** achieves the most balanced performance, outperforming BERT and vanilla RoBERTa baselines. The findings of Track 2 demonstrate how effective **DeBERTa** is when dealing with ordinal-aware losses. Multi-task models increase macro-F1 in Track 3. Track 4 demonstrates that **ensemble models** include classifiers and entailment scores outperform traditional NLI (Natural Language Inferencing) or classification baselines and provide the most promising results across all measures. Overall, the plots highlight

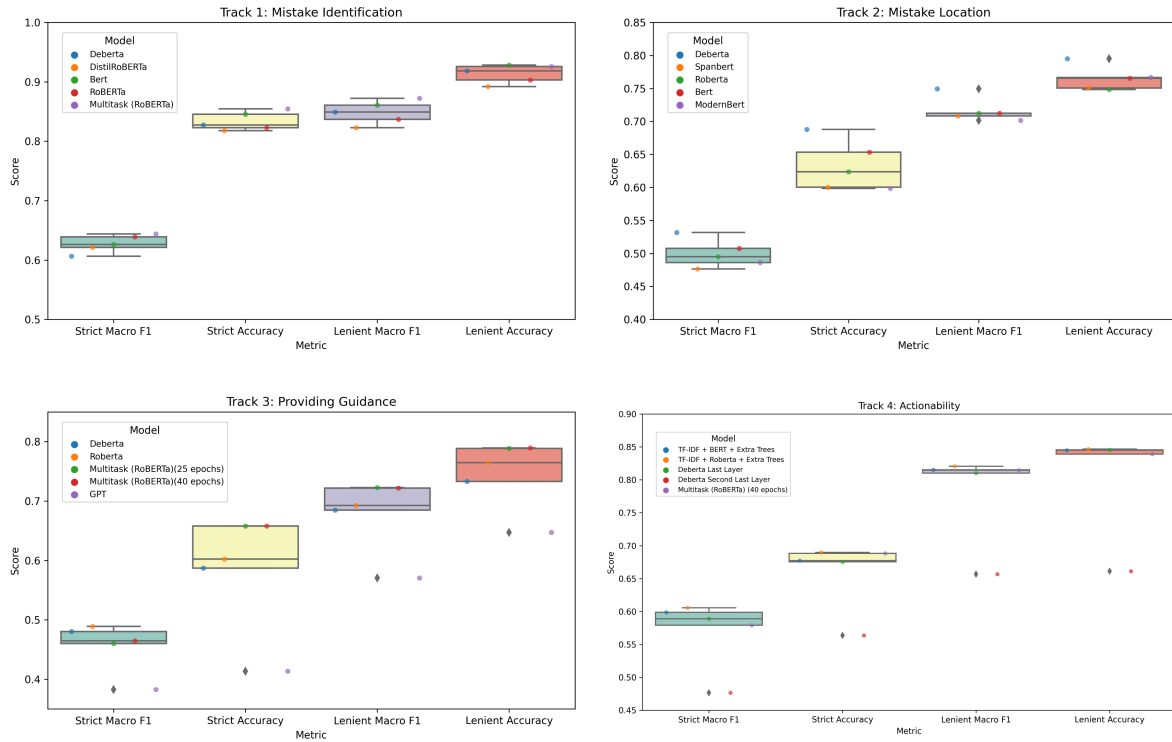


Figure 3: Box Plot showing the evaluation of different models in each track

the advantages of ordinal regression (Cheng and Greiner, 2008; Li and Lin, 2007), stacked ensemble classifiers (Dietterich, 2000) and multi-task learning (Ruder, 2017).

Figure 3 presents **box-and-scatter** plots summarizing model performance across the four BEA 2025 Shared Task tracks. Each subplot represents one track and compares five models across four metrics: *Strict Macro F1*, *Strict Accuracy*, *Lenient Macro F1*, and *Lenient Accuracy*. Boxplots show metric distributions, while scatter points (colored by model) indicate individual scores. In Tracks 1 and 4, the boxes are notably **thin** across all metrics, indicating comparable performance across models and easier tasks overall. Accuracy and macro-F1 scores appear to plateau here, suggesting that fundamentally different strategies could be needed to achieve additional improvements. Tracks 2 and 3, on the other hand, display much **wider** boxes, especially for strict accuracy in both tracks and lenient metrics in Track 3, indicating greater difficulty and more performance variation. Transformer-based models demonstrated benefit: **DeBERTa** led consistently in Track 2, while **multitask RoBERTa** stood out in Track 3, outperforming others across strict and lenient metrics.

Figures 4 and 5 present the **t-SNE** plots (van der Maaten and Hinton, 2008), which show the best-performing models in each track. It can be seen that the models clearly separate "No" from "Yes" + "To some extent" examples when used in a lenient setup, suggesting they handle obvious cases well. However, in a three-class setting (strict evaluation), "Yes" and "To some extent" classes often overlap, leading to difficulties in capturing subtle differences between full and partial affirmations. This overlap illustrates the model's limited capacity to capture nuanced intent as well as the subjective nature of intermediate labels (like "To some extent").

Relative difficulty among the tasks: The results and analysis demonstrate that Tracks 2 and 3 have a higher difficulty level. In Table 2, we see that the metric scores for Tasks 2 and 3 are lower than those of the other tasks. In Figure 3, we also see that the models have diverse scores on the strict accuracy metric for these tasks, indicating these tasks require careful modelling and training. Variation in modeling or training results in quite different scores for Tracks 2 and 3. A similar observation can be made from the radar plot in Figure 2 where the polygons corresponding to different methods are clearly distinctly visible, indicating a difference

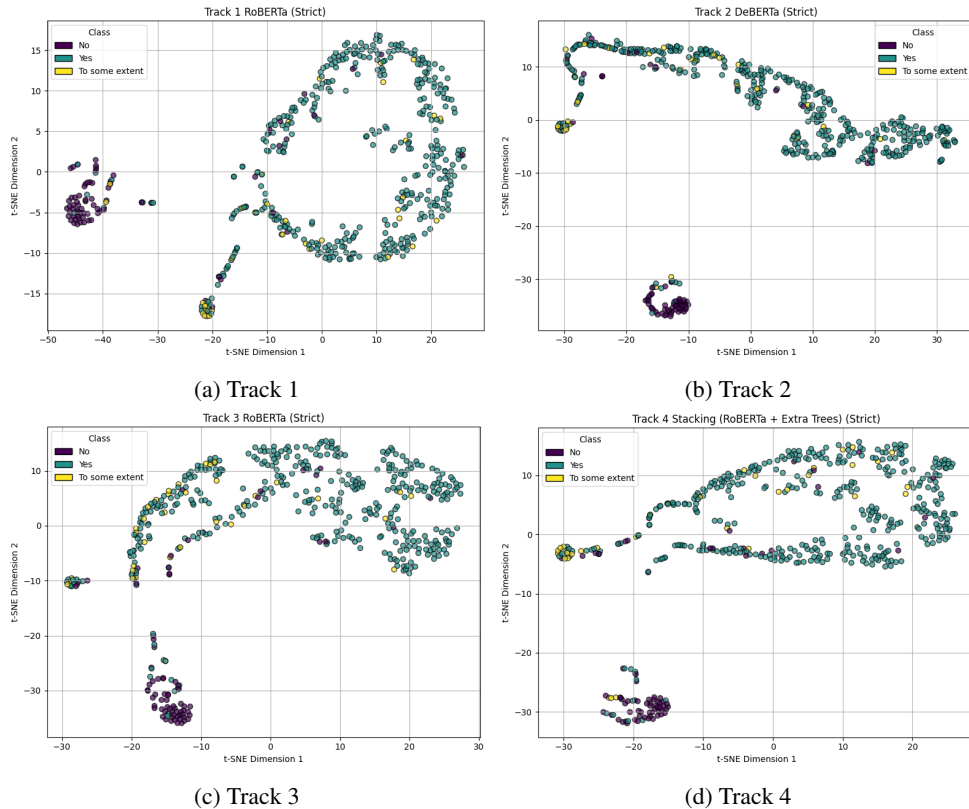


Figure 4: t-SNE Plot showing distribution of classes based on strict evaluation

in the scores. Furthermore, the t-SNE plots in Figures 4 and 5 show many overlaps for the points belonging to different classes in the case of Tracks 2 and 3. We hypothesize that this difficulty might be due to the presence of referencing (identifying error location in Track 2 and providing guidance in terms of how to correct the error in Track 3).

Interpretability: LIME (Ribeiro et al., 2016) is employed for analyzing model interpretability. This has been done for both Track 1 (Mistake Identification) and Track 4 (Actionability). In Figure 6a, highlighted tokens show that the model attends to corrective phrases from the tutor (e.g., "We need", "Remember,", "Let's try counting..."), suggesting alignment with human reasoning when identifying student mistakes. In Task 4 (Figure 6b), attention is emphasized by LIME on mathematical expressions (e.g., "20 plus 7 plus 10 plus 6") and evaluative signals (e.g., "Nice try!", "answer is incorrect"). The suggestion that the model takes into account both numerical and contextual feedback when determining response availability is clear. These visualizations demonstrate how the model uses meaningful context to improve interpretability and confidence in its predictions.

On the overall performance of different represen-

tation techniques and models: Based on our experimental results shown for (a) the held-out dataset in Table 2 and (b) the development data in Appendix A, we see that DeBERTa performed better than RoBERTa in most of the cases. This might be due to the disentangled representation of the token and position vectors of the inputs in DeBERTa, and the attention computation performed on these word and position matrices separately. Also, DeBERTa uses adversarial inputs for its fine tuning which makes it robust. We also see that MultiTask learning helps in good performance across the tasks. This is because the tasks in the 4 tracks are strongly related to each other. All tracks aim to help the student with inputs to identify and correct mistakes. Due to this commonality among the tasks, we felt that a joint model could leverage the signals across the tasks and perform well. We did not use any LLM based approach as (a) it would be difficult to explain its decisions without effective prompting, (b) the results of LLM response may change significantly between different prompts, (c) coming up with good prompt requires extensive trial and error, and (d) extensive experimentations would require costly subscription of the API keys.

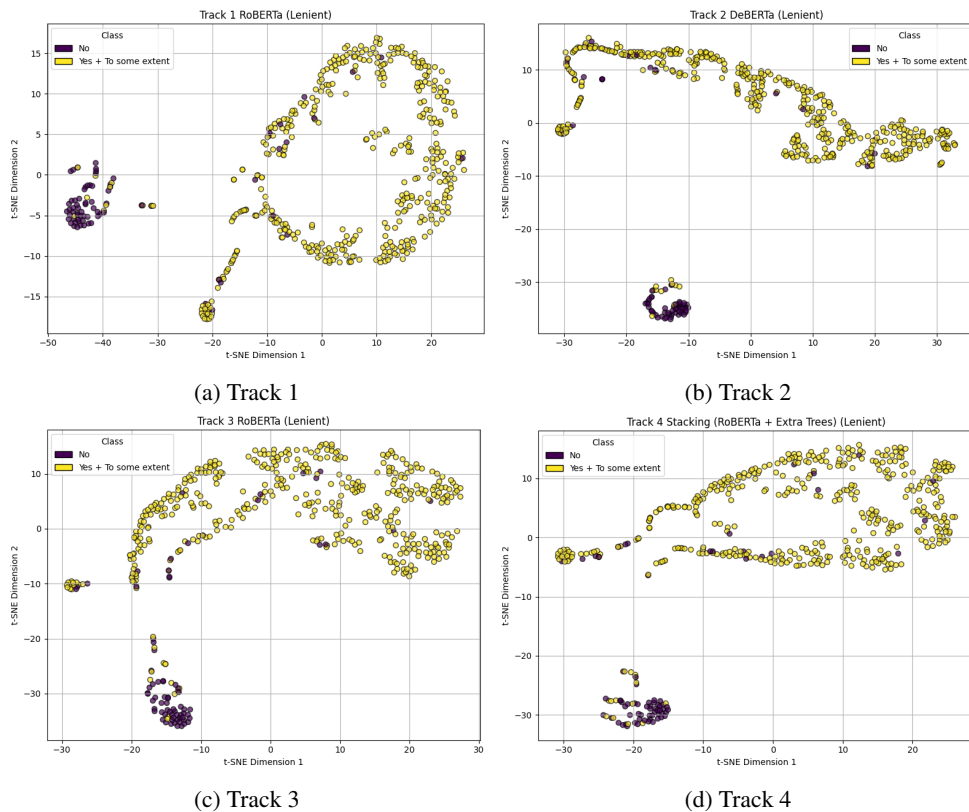


Figure 5: t-SNE Plot showing distribution of classes based on lenient evaluation

Tutor: We need to subtract 6 from 15. Student: oh okay... Tutor: What is the value of 15 - 6? Student: it is 11? [SEP] That's a great try! Remember, 6 is less than 15, so the answer should be bigger than 6. Let's try counting back from 15 six times.

(a) Track 1 - Mistake Identification

Student: 245 Tutor: Nice try! But your answer is incorrect! Tutor: What is the sum of 27 and 16? Student: 11664. [SEP] Let's try that again with a different approach: What is 20 plus 7 plus 10 plus 6?

(b) Track 4 - Actionability

Figure 6: Interpretability analysis for Tracks 1 and 4 with LIME

6 Conclusions

In conclusion, the study demonstrated the effectiveness of transformer-based models, particularly RoBERTa and DeBERTa, which addressed various tasks of pedagogical ability evaluation of AI tutors like mistake identification, mistake location, providing guidance and actionability. We showed how using sampling techniques to balance the dataset is essential to have better discrimination power for the tasks. The results and analysis also demonstrate that Tracks 2 and 3 have a higher difficulty level. This is due to the presence of referencing (identifying error location in Track 2 and providing guidance in terms of how to correct the error in

Track 3). This may be indirectly reflected in how the inputs are organized in the latent space. Due to the relatedness among the tasks, we also see that a multitask approach is well suited for approaching all the tracks in the shared task together.

However, the models have a significant scope for improvement as indicated by the moderate performance of the methods. Also, as the tasks come from the field of education, explainability in the actions is also required. Our future work in this segment will try to focus on these aspects.

Limitations

Our method's limitations include its reliance on the quality of **labeled data** and **high compute requirements** associated with ensemble approaches. There is room for improved semantic modeling because it may also have trouble capturing subtle contextual meanings in feedback. Additionally, performance on the "To some extent" class is variable between tracks, indicating a lack of ability to handle ambiguity.

Ethics Statement

This work is based on a limited size dataset, which constrains the generalizability and trustwor-

thiness of our findings. While transformer-based models are employed that are typically pre-trained on large datasets, the small size of our dataset may limit their full potential. It is acknowledged that the reported results may not fully reflect real-world performance, and future work should be encouraged to validate and extend our findings on larger and more diverse corpora. No sensitive information is present in the dataset. The study adheres to ethical standards for data handling and research transparency.

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*.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.
- Jianlin Cheng and Russell Greiner. 2008. Neural networks for ordinal regression. In *IEEE transactions on neural networks*, volume 19, pages 776–785. IEEE.
- Nico Daheim, Jakob Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. *arXiv preprint arXiv:2407.09136*.
- Thomas G Dietterich. 2000. Ensemble methods in machine learning. *International workshop on multiple classifier systems*, pages 1–15.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.
- Lihui Li and Hsuan-Tien Lin. 2007. Ordinal regression by extended binary classification. In *Advances in neural information processing systems*, pages 865–872.
- Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2019a. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- 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. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.
- Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.
- Trishita Saha, Saroj Kumar Biswas, Saptarsi Sanyal, Souvik Kumar Parui, and Biswajit Purkayastha. 2023. Credit risk prediction using extra trees ensemble method. In *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, pages 1–8.
- Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of Blender and GPT-3 in educational dialogues. *arXiv preprint arXiv:2205.07540*.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. 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.

Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachslar. 2021. [Are we there yet? - a systematic literature review on chatbots in education](#). *Frontiers in Artificial Intelligence*, 4:654924.

A Appendix

This appendix presents a set of quantitative results for each of the four tracks in the BEA 2025 Shared Task. For each track, one table (Table 3, Table 4, Table 5 and Table 6) was included for reporting evaluation metrics - *accuracy*, *macro-F1*, *precision*, and *recall* in both strict and lenient settings for all tested models, obtained on the **development dataset**.

Model	Strict Evaluation				Lenient Evaluation			
	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)
BERT (base, uncased)	0.849	0.593	0.633	0.574	0.929	0.852	0.878	0.830
RoBERTa-base	0.879	0.688	0.742	0.658	0.944	0.884	0.903	0.867
DistilRoBERTa-base	0.865	0.674	0.721	0.646	0.927	0.850	0.868	0.835
DeBERTa-v3-base	0.871	0.672	0.735	0.636	0.934	0.859	0.892	0.833
RoBERTa-base (Focal Loss)	0.827	0.593	0.591	0.597	0.911	0.831	0.821	0.842
MathBERT	0.845	0.596	0.633	0.581	0.919	0.836	0.848	0.825
Multitask (RoBERTa)	0.858	0.553	0.534	0.573	0.919	0.847	0.838	0.857
Multitask (DeBerta)	0.879	0.576	0.572	0.582	0.941	0.881	0.893	0.869
Multitask (Bert)	0.871	0.562	0.579	0.555	0.936	0.861	0.904	0.829

Table 3: TRACK-1: Mistake Identification performance across various transformer models using Strict and Lenient evaluation metrics. Colour codings - Blue (Strict), Green (Lenient)

Model	Strict Evaluation				Lenient Evaluation			
	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)
Without Oversampling								
DeBERTa-v3-base	0.729	0.580	0.592	0.577	0.813	0.753	0.779	0.738
SpanBERT-base-cased	0.684	0.533	0.568	0.527	0.817	0.745	0.803	0.722
Codebert-base	0.688	0.519	0.535	0.512	0.802	0.741	0.765	0.727
Modern-bert-base	0.671	0.564	0.599	0.599	0.813	0.739	0.796	0.717
Roberta-base	0.682	0.542	0.577	0.548	0.813	0.742	0.792	0.721
Bert-base-uncased	0.684	0.506	0.544	0.494	0.802	0.723	0.782	0.701
Multitask (RoBERTa)	0.729	0.476	0.489	0.489	0.809	0.739	0.783	0.719
Multitask (Deberta)	0.739	0.489	0.512	0.498	0.831	0.765	0.823	0.739
Multitask (Bert)	0.720	0.463	0.505	0.472	0.812	0.726	0.811	0.701
With Oversampling								
SpanBERT-base-cased	0.709	0.553	0.559	0.548	0.811	0.759	0.771	0.751
DeBERTa-v3-base	0.694	0.532	0.539	0.528	0.802	0.747	0.761	0.737
Codebert-base	0.659	0.521	0.528	0.525	0.786	0.726	0.739	0.717
Modern-bert-base	0.633	0.536	0.577	0.565	0.813	0.745	0.788	0.725
Roberta-base	0.686	0.56	0.566	0.578	0.798	0.744	0.755	0.737
Bert-base-uncased	0.718	0.533	0.565	0.521	0.807	0.733	0.783	0.713

Table 4: TRACK-2: Mistake Location Performance across various transformer models using Strict and Lenient Evaluation. Colour codings - Blue (Strict), Green (Lenient)

Model	Strict Evaluation				Lenient Evaluation			
	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)
Multitask (RoBERTa)	0.659	0.447	0.434	0.485	0.823	0.722	0.751	0.704
Multitask (Bert)	0.667	0.454	0.472	0.485	0.847	0.742	0.812	0.711
Multitask (Deberta)	0.664	0.458	0.567	0.486	0.836	0.730	0.784	0.704
BERT (Last layer predictions)	0.589	0.503	0.516	0.497	0.748	0.607	0.622	0.599
BERT (Second-last Layer + Linear Classifier)	0.581	0.453	0.507	0.448	0.732	0.594	0.602	0.589
RoBERTa (Last layer predictions)	0.655	0.593	0.611	0.582	0.825	0.733	0.754	0.718
RoBERTa (Second-last Layer + Linear Classifier)	0.282	0.288	0.731	0.439	0.841	0.688	0.901	0.654
DeBERTa (Last layer predictions)	0.601	0.524	0.539	0.520	0.760	0.611	0.637	0.601
DeBERTa (Second-last Layer + Linear Classifier)	0.615	0.418	0.671	0.425	0.813	0.611	0.847	0.598
DistilRoberta (Last layer predictions)	0.601	0.522	0.543	0.517	0.778	0.636	0.672	0.622
DistilRoberta (Second-last Layer + Linear Classifier)	0.479	0.376	0.525	0.427	0.561	0.529	0.568	0.597

Table 5: TRACK-3: Providing Guidance performance across various transformer models using Strict and Lenient evaluation metrics. Colour codings - Blue (Strict), Green (Lenient)

Model	Strict Evaluation				Lenient Evaluation			
	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)
Multitask (RoBERTa)	0.669	0.449	0.469	0.472	0.801	0.722	0.784	0.701
Multitask (Bert)	0.669	0.449	0.493	0.469	0.815	0.731	0.826	0.705
Multitask (Deberta)	0.715	0.505	0.484	0.536	0.844	0.807	0.814	0.799
Stacking (BERT + Extra Trees)	0.754	0.655	0.675	0.648	0.867	0.849	0.847	0.851
Stacking (BERT + Logistic Regression)	0.744	0.637	0.654	0.632	0.873	0.855	0.854	0.856
Stacking (RoBERTa + Extra Trees)	0.744	0.632	0.646	0.628	0.875	0.857	0.858	0.855
Stacking (RoBERTa + Logistic Regression)	0.756	0.662	0.674	0.657	0.879	0.862	0.862	0.862
Stacking (DeBERTa + Extra Trees)	0.734	0.647	0.651	0.645	0.881	0.861	0.869	0.855
Stacking (DeBERTa + Logistic Regression)	0.726	0.629	0.637	0.623	0.873	0.850	0.863	0.841

Table 6: TRACK-4: Actionability performance across various models for Strict and Lenient evaluations. Colour codings - Blue (Strict), Green (Lenient)

NeuralNexus at BEA 2025 Shared Task: Retrieval-Augmented Prompting for Mistake Identification in AI Tutors

Numaan Naeem Sarfraz Ahmad Momina Ahsan Hasan Iqbal

MBZUAI

{numaan.naeem, sarfraz.ahmad, momina.ahsan, hasan.iqbal}@mbzuai.ac.ae

Abstract

This paper presents our system for **TRACK 1: MISTAKE IDENTIFICATION** in the BEA 2025 SHARED TASK ON PEDAGOGICAL ABILITY ASSESSMENT OF AI-POWERED TUTORS. The task involves evaluating whether a tutor’s response correctly identifies a mistake in a student’s mathematical reasoning. We explore four approaches: (1) an ensemble of machine learning models over pooled token embeddings from multiple pretrained language models (LMs); (2) a frozen sentence-transformer using [CLS] embeddings with an MLP classifier; (3) a history-aware model with multi-head attention between token-level history and response embeddings; and (4) a retrieval-augmented few-shot prompting system with a large language model (LLM) i.e. GPT 4o. Our final system retrieves semantically similar examples, constructs structured prompts, and uses schema-guided output parsing to produce interpretable predictions. It outperforms all baselines, demonstrating the effectiveness of combining example-driven prompting with LLM reasoning for pedagogical feedback assessment. Our code is available at https://github.com/NaumanNaeem/BEA_2025.

1 Introduction

Conversational AI systems are increasingly being used for educational applications, particularly in the form of AI-powered tutors that can engage students in instructional dialogues. While recent advances in LLMs have made it possible to generate fluent and context-aware responses, evaluating whether these responses exhibit true pedagogical ability remains a fundamental challenge. Traditional dialogue evaluation metrics, such as fluency, coherence, or BLEU-like scores, fall short in capturing educational effectiveness, such as whether the tutor correctly identifies a student’s mistake or provides helpful, targeted feedback.

The BEA 2025 SHARED TASK ON PEDAGOGICAL ABILITY ASSESSMENT OF AI-POWERED TUTORS (Kochmar et al., 2025) addresses this gap by introducing a standardized evaluation benchmark and taxonomy to assess pedagogical abilities in AI-generated tutor responses. In particular, TRACK 1: MISTAKE IDENTIFICATION focuses on determining whether a tutor’s response correctly detects and communicates an error in the student’s reasoning within a mathematical dialogue. The benchmark used in this task is based on MRBENCH (Maurya et al., 2025), which includes 192 dialogues and over 1,500 responses from human and LLM tutors, annotated across *eight* pedagogical dimensions grounded in learning sciences.

2 Methodology

We tackle TRACK 1: MISTAKE IDENTIFICATION, which involves determining whether a tutor’s response correctly identifies a student’s mistake in a multi-turn mathematical dialogue. Given the subtle and varied nature of student errors and tutor feedback, this task demands both contextual understanding and pedagogical sensitivity. To address this, we developed and evaluated *four* distinct approaches: three baseline models leveraging traditional classification techniques and transformer embeddings, followed by a final retrieval-augmented few-shot classification technique using LLMs.

2.1 Layered Embedding Extraction with Classical ML Ensemble

In our first baseline, we designed a layered ensemble approach by extracting embeddings from several pre-trained transformer models, including BERT, ROBERTA, XLNET, T5, and GPT-2. To handle this flexibly, we developed a unified LM_EMBED class that tokenizes and encodes both conversation history and tutor responses using each model’s specific configuration. We ap-

plied average pooling over the token embeddings to produce fixed-length vectors for each input, and then averaged the conversation and response vectors to create the final input representation. Using these features, we trained a diverse set of traditional classifiers i.e. SVM, Decision Tree, Random Forest, Logistic Regression, Naive Bayes, KNN, AdaBoost, and MLP, each optimized using GRID-SEARCHCV with 10-fold cross-validation. We then built a meta-classifier by stacking the prediction probabilities from these base models and training a logistic regression model on top. This ensemble strategy allowed us to combine the strengths of different embedding models and classifiers, leading to more stable and accurate predictions compared to using any single model alone.

2.2 Token-Level Attention with History-Aware Model

In our second baseline, we modeled the interaction between the conversation history and tutor response using a token-level attention mechanism without any pooling during embedding extraction. We used a transformer encoder (sentence-transformers/all-mpnet-base-v2) to obtain full token-level representations for both the conversation history and the tutor’s response. These representations were then passed into a custom multi-head attention module. Specifically, we treated the response as the query (Q) and the history as both the key (K) and value (V) in a standard multi-head attention setup. The output of the attention layer was mean-pooled along the sequence length dimension, and a small feedforward network mapped the pooled vector to three output classes. The model was trained using cross-entropy loss with the ADAMW optimizer, and predictions were generated by taking the argmax over the logits. This architecture allows the model to explicitly attend to relevant parts of the history when interpreting the tutor’s response, resulting in a more nuanced classification of pedagogical mistakes.

2.3 Frozen Sentence-Transformer with MLP Classifier

Our third baseline models the pedagogical mistake identification task as a supervised classification problem using fixed sentence embeddings. We use a frozen sentence-level transformer model (sentence-transformers/all-mpnet-base-v2) to independently encode the conversation history

and the tutor’s response, extracting the [CLS] token from the final hidden state as a dense representation. These embeddings are projected through two separate linear layers and concatenated to form a joint feature vector, which is passed to a shallow feedforward neural network to predict one of three mistake identification categories. We trained this model using cross-entropy loss and the ADAMW optimizer, keeping the encoder frozen throughout training. To improve efficiency, we cached the embeddings as .npz files. The final output was restructured to match the original JSON format for evaluation, preserving conversation IDs, model names, tutor responses, and predicted mistake annotations.

2.4 Retrieval-Augmented Few-Shot Classification with LLM-as-a-Judge

Our final and most effective approach tackles the mistake identification task as a judgment problem, using a retrieval-augmented few-shot prompting strategy powered by large language models (LLMs). Instead of training a traditional classifier, we designed a modular pipeline built with LangChain. At its core, the system takes the full conversation history and the tutor’s response, then prompts an LLM, specifically, GPT-4o to assess whether the tutor has correctly identified a mistake in the student’s reasoning.

Figure 1 outlines the system architecture. We begin by embedding the conversation history and tutor responses from the MRBENCH training set using the OpenAI Embedding Model. These embeddings are stored in a persistent vector database using ChromaDB. With this setup, we construct a few-shot prompt template and use the LLM itself as a “judge” on the test data. At inference time, the system retrieves the top- k semantically similar examples and integrates them into the prompt.

Each prompt includes detailed labeling instructions, definitions for all possible labels (Yes, No, To some extent), and the full dialogue context. (see Appendix B for more information). To ensure clear and structured outputs, we use a PydanticOutputParser that enforces a strict schema and reliably extracts the label from the LLM’s response. The pipeline also supports retries and incremental saving, making it robust and efficient for large-scale processing.

By combining relevant examples with a powerful instruction-following model, this method allows for nuanced mistake identification beyond simple clas-

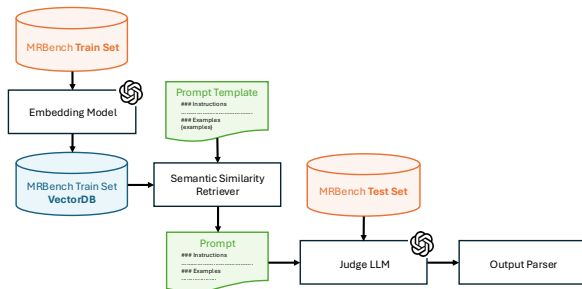


Figure 1: Pipeline of our final approach for mistake identification. The system takes tutor–student dialogue as input, retrieves relevant examples, constructs a structured prompt, and uses LLM to predict whether a mistake is identified. The output is parsed and saved.

sification. It requires no fine-tuning, generalizes well to new inputs, and showed improvements in both accuracy and qualitative evaluations compared to baseline methods. This highlights the effectiveness of prompt-based, retrieval-augmented approaches in educational and feedback-driven NLP tasks.

3 Dataset

We use the dataset introduced by Maurya et al. (2025), which includes both development and test splits. The development set consists of 300 dialogues from Macina et al. (2023) and Wang et al. (2024), each ending with a student utterance that reflects confusion or a mistake. Tutor responses, generated by seven large language model systems and human tutors (one in MathDial Macina et al. (2023), expert and novice in Bridge), are annotated along four pedagogical dimensions: (1) Mistake Identification, (2) Mistake Location, (3) Providing Guidance, and (4) Actionability. In total, the development set includes over 2,480 annotated responses.

The test set contains 200 dialogues with the same structure, but tutor identities are anonymized (for example, Tutor_1, Tutor_2), and no annotations are provided. This allows for blind evaluation of system outputs under the shared task setting.

3.1 Pre-processing

For the baseline systems, we apply extensive pre-processing to both the conversation history and tutor response texts. This includes converting text to lowercase, removing punctuation, stripping emojis, and cleaning URLs, HTML tags, and contractions. We also remove stopwords using the NLTK stopword list. All texts are passed through a unified

normalization pipeline to reduce noise and ensure consistency. The labels for Mistake Identification are mapped to numeric values as follows: No \rightarrow 0, Yes \rightarrow 1, and To Some Extent \rightarrow 2.

For our final approach, we additionally preprocess both the development and test sets so that each dialogue is reformatted into evenly paired exchanges between tutor and student, preserving the integrity of the back-and-forth interaction. During this process, we addressed two key issues. First, some conversations included greetings or closing phrases (e.g., “Hi”, “Thank you”) that did not contribute to the reasoning process. These were removed to maintain focus on educational content. Second, a few dialogues contained erroneous segments where the tutor responded to its own utterance without student input. These cases were consistently found to follow a correctly structured exchange and were manually removed (see Appendix A for examples).

This pre-processing step ensured a clean and consistent input format, enabling reliable downstream processing and model evaluation.

4 Evaluation and Results

We evaluated all four approaches on the **Track 1: Mistake Identification** test set using two evaluation schemes: **Strict** and **Lenient**, each reporting both *Macro F1* and *Accuracy*. In the strict setting, only exact matches with the gold labels are considered correct. In contrast, the lenient setting provides partial credit by treating To some extent as aligning with Yes, reflecting the fuzzy nature of pedagogical judgments in borderline cases. Table 1 summarizes the results.

As expected, the first baseline using pooled token embeddings and an ensemble of traditional classifiers (Approach 1) offered a modest starting point. This method, while straightforward, lacked the capacity to fully capture the nuances in dialogue-based reasoning.

Introducing token-level attention in Approach 2 led to a notable jump in performance. This suggests that modeling fine-grained interactions between the student’s dialogue and the tutor’s response helps the model better identify whether a mistake was correctly addressed. However, while this approach added depth to the representation, it still relied on relatively shallow modeling of the context.

Approach 3, which used frozen [CLS] embeddings from a sentence transformer combined with

Approach	Strict F1	Strict Acc	Lenient F1	Lenient Acc
Approach 1 (ML Ensemble)	0.446	0.657	0.637	0.754
Approach 2 (Token-Level Attention)	0.571	0.765	0.777	0.865
Approach 3 (CLS + MLP)	0.583	0.809	0.805	0.888
Approach 4 (Few-shot LLM + Retrieval)	0.584	0.827	0.814	0.897

Table 1: Performance of all four approaches on the BEA 2025 Mistake Identification test set under strict and lenient evaluation settings.

an MLP classifier, further improved performance. This indicates that sentence-level semantic representations, especially when paired with a focused classification head, can offer a stronger understanding of the overall pedagogical intent.

Our final method, Approach 4, which frames the task as a retrieval-augmented prompting problem with GPT-4o, achieved the best performance across all metrics. By retrieving semantically similar examples and using detailed, schema-guided prompts, the system benefited from both contextual grounding and the powerful instruction-following capabilities of modern LLMs. Notably, it showed strong results in both strict and lenient settings, highlighting its ability to make fine distinctions while still handling ambiguity in borderline cases effectively. Our final submission, achieved an official leaderboard rank of 37th among all participants.

5 Conclusion

We developed and evaluated four approaches for the BEA 2025 Shared Task **Track 1: Mistake Identification**, culminating in a retrieval-augmented few-shot prompting system using GPT-4o. While our initial baselines used traditional classifiers over pretrained embeddings, the final system reframed the task as a structured judgment problem, combining semantically retrieved examples, instruction-driven prompting, and schema-constrained output parsing.

This approach consistently outperformed all baselines in both strict and lenient evaluations, achieving a strict Macro F1 of 0.584 and a lenient accuracy of 0.897. It was particularly effective at capturing nuanced pedagogical feedback, highlighting the strength of LLM-based reasoning when guided by relevant context. Our submission ranked 37th on the official leaderboard, demonstrating the competitiveness of our method.

These results show that retrieval-augmented prompting offers a scalable and effective solution

for assessing complex teaching behaviors in AI tutors. Future work could explore more adaptive example selection, multi-turn consistency, and alignment with broader goals such as helpfulness and instructional fairness.

Limitations

While our final system achieved the best performance among all submitted approaches, it still has several limitations that suggest promising directions for future work.

Limited Diversity in Retrieved Examples The effectiveness of our retrieval-augmented prompting pipeline depends heavily on the quality and coverage of the example pool. Since we rely on a fixed set of annotated training examples, the system may struggle with out-of-distribution dialogues or question types that are underrepresented in the retrieval set. Moreover, retrieval is based solely on static embedding similarity from OpenAI embeddings, without adapting to the context or emphasizing specific pedagogical traits.

Lack of Multi-Turn Dialogue Modeling Each input is treated as a standalone conversation-response pair, with no memory of earlier tutor turns or evolving dialogue context. This limits the system’s ability to track learning progression or take prior feedback into account. Modeling dialogue history explicitly—through dialogue state tracking or memory-based retrieval—could improve consistency and pedagogical depth in multi-turn interactions.

Simplified Output Format Although the use of a structured parser ensures consistency, it restricts the model to selecting a single label per example. It does not capture uncertainty, nuanced justifications, or cases where multiple labels might apply. Extending the output to include rationales or confidence

scores could make evaluations more informative and reflective of real-world ambiguity.

Scalability and Cost Constraints Inference with frontier models like GPT-4o is computationally intensive and dependent on external APIs, which introduces latency, cost, and rate-limit challenges. These constraints pose barriers to deployment in low-resource settings or real-time tutoring applications, where efficiency and scalability are critical.

References

- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- 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.
- 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](#). pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. [Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes](#). pages 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.

A Preprocessing Examples

This appendix list some issues which are fixed during pre-processing of dataset.

In the following example from the development set, the initial student message is a casual greeting that disrupts the expected alternating structure of the dialogue. To maintain structural integrity and ensure an even number of turns between tutor and student, such non-essential messages are removed during preprocessing.

```
[  
'Student: okey',  
'Tutor: What is 25 minus 18?',  
'Student: 8'  
]
```

In the example below, the final tutor response erroneously mimics the student’s explanation, as if the tutor is responding to itself rather than engaging with the student. This type of error breaks the natural flow of the dialogue and was manually identified and removed during preprocessing to ensure accurate tutor-student interaction.

```
[  
'Tutor: Hi, could you please provide a  
↪ step-by-step solution for the question  
↪ below? The question is ...',  
'Student: Samantha buys 4 toys at $12.00 each.  
↪ For each pair of toys...',  
'Tutor: I added the two amounts together to get  
↪ a total of $36.00 + $6.00 = $42.00.'  
]
```

In cases like the example below, the tutor’s prompt is split across multiple turns, breaking the intended question into separate messages. To preserve the coherence of the dialogue and maintain a consistent turn-taking structure, such fragmented tutor responses are merged into a single utterance by concatenating the strings.

```
[  
'Tutor: Hi, could you please provide a  
↪ step-by-step solution for the question  
↪ below? The question is: Tyson decided to  
↪ make muffaletta sandwiches for ...',  
'Tutor: How many pounds of meat are needed for  
↪ each sandwich?',  
'Student: Each sandwich requires 1 pound of  
↪ meat and 1 pound of cheese.',  
'Tutor: What is the cost of 1 pound of meat?',  
↪ 'Student: The cost of 1 pound of meat is  
↪ $7.00.'  
]
```

B Prompt Engineering

```
"""
You will be shown a short educational "Conversation" between a tutor and a student, including the
↳ student's solution and the tutor's follow-up "Response". Your task is to judge whether the
↳ tutor's response successfully **identifies a mistake** in the student's reasoning.

### Instructions
1. Read the entire dialogue to understand the context of the student's solution.
2. Focus on whether the tutor's response explicitly or implicitly calls out an error.
3. Reply **only** with one of the labels: `Yes`, `To some extent`, or `No`.

### Labels
- `Yes`: The mistake is clearly identified/recognized in the tutor's response. The tutor implicitly
↳ or explicitly points out the error in the student's reasoning.
- `No`: The tutor's response does not identify any mistake in the student's reasoning. The tutor's
↳ response is either irrelevant or does not address the student's solution.
- `To some extent`: The tutor's response suggests that there may be a mistake, but it sounds as if
↳ the tutor is not certain.

### Format Instructions:
{format_instructions}
Return only the classification label without any additional commentary or extraneous details.

### Examples
{examples}

## Mistake Identification
### Conversation
{conversation}

### Response
{response}
"""
```

Figure 2: Prompt for LLM which is used a judge in Retrieval-Augmented Few-shot classification approach

DLSU at BEA 2025 Shared Task: Towards Establishing Baseline Models for Pedagogical Response Evaluation Tasks

Mark Edward M. Gonzales*, Lanz Kendall Lim*, Maria Monica Manlises*

College of Computer Studies, De La Salle University

Manila, Philippines

{mark_gonzales, lanz_kendall_lim, maria_monica_manlises}@dlsu.edu.ph

Abstract

We present our submission for Tracks 3 (Providing Guidance), 4 (Actionability), and 5 (Tutor Identification) of the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-Powered Tutors. Our approach sought to investigate the performance of directly using sentence embeddings of tutor responses as input to downstream classifiers (that is, without employing any fine-tuning). To this end, we benchmarked two general-purpose sentence embedding models: gte-modernbert-base (GTE) and all-MiniLM-L12-v2, in combination with two downstream classifiers: XGBoost and multilayer perceptron. Feeding GTE embeddings to a multilayer perceptron achieved macro-F1 scores of 0.4776, 0.5294, and 0.6420 on the official test sets for Tracks 3, 4, and 5, respectively. While overall performance was modest, these results offer insights into the challenges of pedagogical response evaluation and establish a baseline for future improvements.

1 Introduction

Recent advancements in large language models (LLMs) have opened new possibilities for using AI-powered chatbots as educational tutors, providing benefits for tasks such as homework assistance, personalized learning, and skills development (Labadze et al., 2023). However, while these systems can generate human-like dialogue, assessing their pedagogical effectiveness remains a significant challenge. In the past, human evaluation has typically been used for evaluation, though reliable, this is costly and difficult to scale (Liu et al., 2023).

To address this gap, the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered Tutors (Kochmar et al., 2025) was organized to promote the development of automated evaluation systems for tutor responses in educational dialogues.

The shared task focused on assessing the quality of tutor responses aimed at helping students correct their mistakes in math-related dialogues. Participants were provided with dialogues that included conversation history, a student’s incorrect utterance, and multiple possible tutor responses (Maurya et al., 2025). Each response was to be evaluated along four pedagogically motivated dimensions: mistake identification, mistake location, guidance provision, and actionability. These dimensions were annotated on a three-point scale: “Yes,” “To some extent,” or “No.”

In addition to these four tracks, the shared task included a fifth track, Tutor Identification, wherein participants were asked to predict the origin of anonymous tutor responses, distinguishing between different LLMs and human tutors. This track explored whether distinct pedagogical or linguistic styles could be used to attribute responses to their source.

The organizers released a development dataset of 300 annotated dialogues and a test set of 191 dialogues. Both sets included responses from a diverse set of state-of-the-art LLMs and, in some cases, human tutors (Maurya et al., 2025).

Our contributions are as follows:

- We evaluated the performance of directly feeding sentence embeddings of tutor responses (without any fine-tuning) to downstream classifiers for Tracks 3 (Providing Guidance), 4 (Actionability), and 5 (Tutor Identification).
- We benchmarked two sentence embedding models: gte-modernbert-base (GTE) and all-MiniLM-L12-v2. Our results show that using GTE embeddings and a multilayer perceptron yielded macro-F1 scores of 0.4776, 0.5294, and 0.6420, thus providing a baseline for the performance of general-purpose sentence embeddings on multiple pedagogical response evaluation tasks.

*Contributed equally

2 Methods

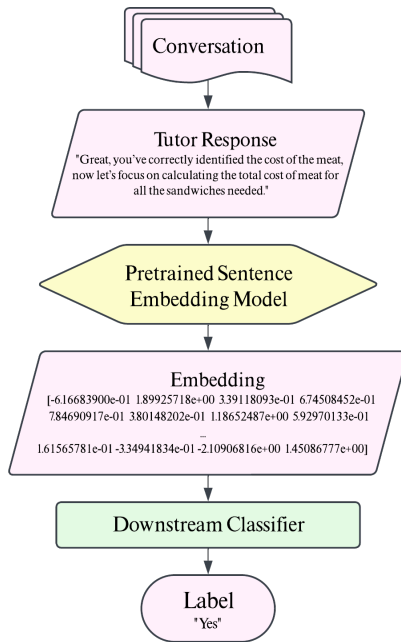


Figure 1: Methodology

Figure 1 shows our approach. First, we extracted the tutor response from each dialogue instance and fed it into a pretrained sentence embedding model to obtain a fixed-length vector representation. We used this representation as input to a classifier trained to predict the relevant task labels.

This methodology was applied across all three tracks in which we participated. We modeled Tracks 3 and 4 as multiclass classification problems where the output labels are “No,” “To some extent,” and “Yes.” Likewise, Track 5 was also modeled as a multiclass classification problem with nine output labels: “Expert,” “GPT4,” “Gemini,” “Llama31405B,” “Llama318B,” “Mistral,” “Novice,” “Phi3,” and “Sonnet.”

2.1 Sentence Embedding Model

Two embedding models were chosen from the Massive Text Embedding Benchmark (MTEB) Leaderboard¹ (Enevoldsen et al., 2025), which compares the performance of over a hundred embedding models across multiple tasks.

We first selected **gte-modernbert-base**² (Zhang et al., 2024) or GTE, a general-purpose embedding

¹<https://huggingface.co/spaces/mteb/leaderboard>

²<https://huggingface.co/Alibaba-NLP/gte-modernbert-base>

model built on modernBERT (Warner et al., 2024). With 149 million parameters and a context length of up to 8192 tokens, it performs strongly on the MTEB leaderboard, competitive with other models with under 1 billion parameters.

In addition, we also evaluated a more lightweight model, **all-MiniLM-L12-v2**³, which has 33.4M parameters. Despite its compact size, it registers competitive performance on the MTEB leaderboard and on other classification tasks (Meleti et al., 2025).

2.2 Downstream Classifier

We trained two classification models: **XGBoost** and a **multilayer perceptron (MLP)** with a single hidden layer. XGBoost, a decision tree-based gradient boosting method, has been reported to achieve good performance with dense sentence embeddings as input (Muqadas et al., 2025; Chen and Guestrin, 2016). MLPs are capable of capturing nonlinear relationships and, as such, are widely used in supervised learning tasks (Goodfellow et al., 2016).

We partitioned the development set such that 80% of the data comprises the training set and the remaining 20% comprises the test set. We then performed three-fold cross-validation with grid search on the training set to tune the hyperparameters of both models; the complete hyperparameter search space is reported in Table 3. Tables 4 and 5 show the combination of hyperparameters that returned the highest macro-F1.

3 Results and Discussion

3.1 Development Set Results

Table 1 summarizes the results on the test set partition of our development set. We found that using MLP consistently outperformed using XGBoost in terms of macro-F1 across all three tasks, with the strongest gains observed in Tracks 4 (Actionability) and 5 (Tutor Identification). Pairing GTE embeddings with MLP achieved the highest macro-F1 and also the highest accuracy (except for Task 3). Confusion matrices are given in Figure 2.

3.2 Official Test Set Results

Based on the development set results, we selected the top two model combinations for final testing. For Tracks 3 and 4, we chose GTE + MLP and GTE + XGBoost. For Track 5, we selected GTE + MLP and MiniLM + MLP. The complete official test set

³<https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2>

Task	Model	Macro-F1	Accuracy
Track 3	GTE + MLP	0.5601	0.6371
	GTE + XGBoost	0.5095	0.6492
	MiniLM + MLP	0.4675	0.6371
	MiniLM + XGBoost	0.4814	0.6310
Track 4	GTE + MLP	0.5667	0.6492
	GTE + XGBoost	0.5097	0.6552
	MiniLM + MLP	0.5504	0.6411
	MiniLM + XGBoost	0.4766	0.6431
Track 5	GTE + MLP	0.6047	0.5665
	GTE + XGBoost	0.4879	0.4476
	MiniLM + MLP	0.5333	0.4879
	MiniLM + XGBoost	0.4595	0.3992

Table 1: Macro-F1 and accuracy on the development set across Tracks 3 (Providing Guidance), 4 (Actionability), and 5 (Tutor Identification). The best performance scores are in bold.

scores for these selected model combinations are reported in Table 2.

3.3 Limitations

First, we fed the tutor responses, as is, to the sentence embedding models, that is, we did not perform any text preprocessing (such as stopword removal or punctuation stripping) prior to embedding. While this decision aligns with the intention to evaluate the raw utility of general-purpose embeddings, preprocessing might have potentially reduced noise and improved classification performance.

Second, we did not fine-tune the sentence embedding models on task-specific data. The GTE and MiniLM embeddings were used as is, without adaptation to the tutoring domain or label space. This might have limited the models’ ability to capture nuanced patterns in the instructional dialogue, particularly for more subtle distinctions such as “To some extent” in Tracks 3 and 4 or between tutor personas in Track 5.

Finally, the per-class evaluation results (Figure 3) reflect the class imbalance, with the dominant class (“Yes”) having noticeably higher F1 compared to “No” and “To some extent” for Tracks 3 and 4. To address this, it may be helpful to incorporate class-adjusted weights during training, perform data augmentation, or generate synthetic data.

4 Conclusion

In this paper, we investigated the performance of directly feeding sentence embeddings of tutor responses to downstream classifiers for multiple pedagogical response evaluation tasks, thus providing

baseline models for future improvements in this domain.

For future work, it may be interesting to compare these baselines with domain-specific fine-tuning, as well as perform more extensive hyperparameter tuning through automated optimization techniques (such as Bayesian optimization) to further improve classification accuracy.

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, KDD ’16*, page 785–794, New York, NY, USA. Association for Computing Machinery.
- Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Ryrstrøm, Roman Solomatin, and 67 others. 2025. [MMTEB: Massive multilingual text embedding benchmark](#). *arXiv preprint arXiv:2502.13595*.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Lasha Labadze, Maya Grigolia, and Lela Machaidze. 2023. [Role of AI chatbots in education: systematic literature review](#). *International Journal of Educational Technology in Higher Education*, 20:56.
- 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.
- 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.

Task	Model	Exact F1	Exact Acc	Lenient F1	Lenient Acc
Track 3 (Providing Guidance)	GTE + MLP	0.4776	0.5669	0.6755	0.7382
	GTE + XGBoost	0.4545	0.6244	0.6784	0.7712
Track 4 (Actionability)	GTE + MLP	0.5294	0.6089	0.7351	0.7738
	GTE + XGBoost	0.4966	0.6102	0.7170	0.7789
Track 5 (Tutor Identification)	GTE + MLP	0.6420	0.6231	–	–
	MiniLM + MLP	0.5808	0.5624	–	–

Table 2: Performance on the official test sets. “F1” is shorthand for macro-F1, and “Acc” stands for accuracy. For Tracks 3 and 4, two additional metrics were additionally computed by the testing platform: lenient F1 and lenient accuracy, which consider “Yes” and “To some extent” the same class. The qualifier “exact” distinguishes the conventional metrics from their lenient variation.

Marco Meleti, Stefano Guizzardi, Elena Calciolari, and Carlo Galli. 2025. [A comparative analysis of sentence transformer models for automated journal recommendation using pubmed metadata](#). *Big Data and Cognitive Computing*, 9(3):67.

Amara Muqadas, Hikmat Ullah Khan, Muhammad Ramzan, Anam Naz, Tariq Alsahfi, and Ali Daud. 2025. [Deep learning and sentence embeddings for detection of clickbait news from online content](#). *Scientific Reports*, 15(1):13251.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2024. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). *arXiv preprint arXiv:2412.13663*.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024. [mgte: Generalized long-context text representation and reranking models for multilingual text retrieval](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.

Hyperparameter	Search Space
<i>XGBoost</i>	
Number of estimators	50, 100, 150
Maximum depth of a tree	3, 5, 7
Learning rate	0.01, 0.1, 0.2
Subsample ratio of the training instances	0.8, 1.0
Subsample ratio of columns when constructing each tree	0.8, 1.0
<i>MLP</i>	
Hidden layer size	(50,), (100,), (150,)
Activation	ReLU, tanh, logistic
Solver	Adam, SGD
L2 regularization strength	10^{-4} , 10^{-3} , 10^{-2}
Learning rate schedule	Constant, adaptive

Table 3: Hyperparameter search space

Task	Embedding	n_estimators	max_depth	learning_rate	subsample	colsample_bytree
Track 3	GTE	100	7	0.2	1.0	0.8
	MiniLM	50	5	0.2	1.0	1.0
Track 4	GTE	150	3	0.2	0.8	0.8
	MiniLM	150	7	0.1	0.8	0.8
Track 5	GTE	150	3	0.2	0.8	0.8
	MiniLM	150	5	0.1	0.8	1.0

Table 4: Optimal XGBoost hyperparameters selected via three-fold cross-validation with grid search for each task and sentence embedding model. n_estimators refers to the number of estimators; max_depth, maximum depth of a tree; learning_rate, learning rate; subsample, subsample ratio of the training instances; and colsample_bytree, subsample ratio of columns when constructing a tree.

Task	Embedding	Activation	L2 Reg.	Hidden Layer Size	Learning Rate Schedule	Solver
Track 3	GTE	ReLU	10^{-2}	(150,)	Constant	Adam
	MiniLM	tanh	10^{-4}	(150,)	Constant	SGD
Track 4	GTE	ReLU	10^{-4}	(50,)	Constant	Adam
	MiniLM	ReLU	10^{-4}	(150,)	Constant	Adam
Track 5	GTE	Logistic	10^{-3}	(50,)	Constant	Adam
	MiniLM	Logistic	10^{-2}	(50,)	Constant	Adam

Table 5: Optimal MLP hyperparameters selected via three-fold cross-validation with grid search for each task and sentence embedding model

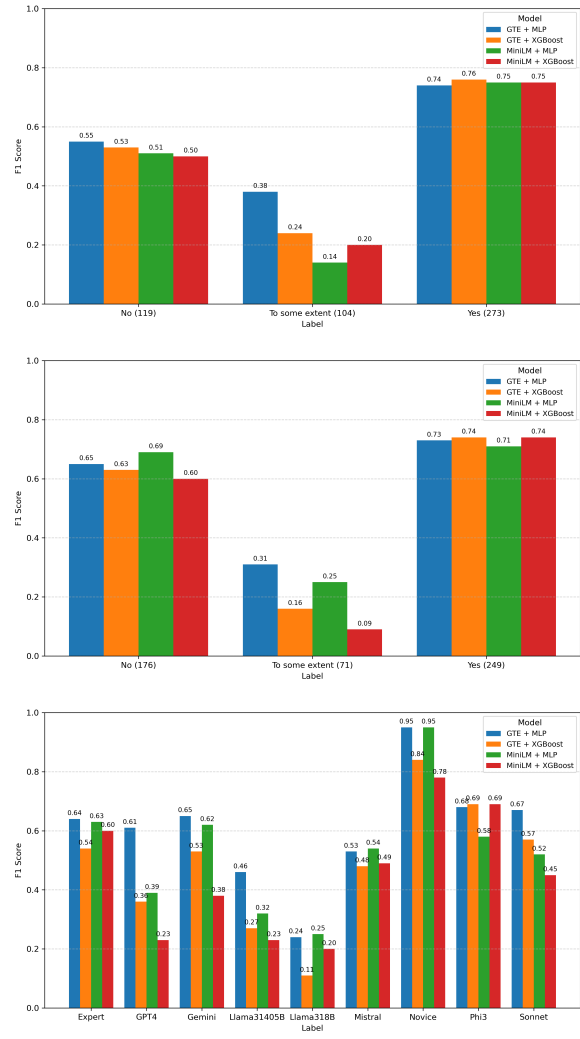
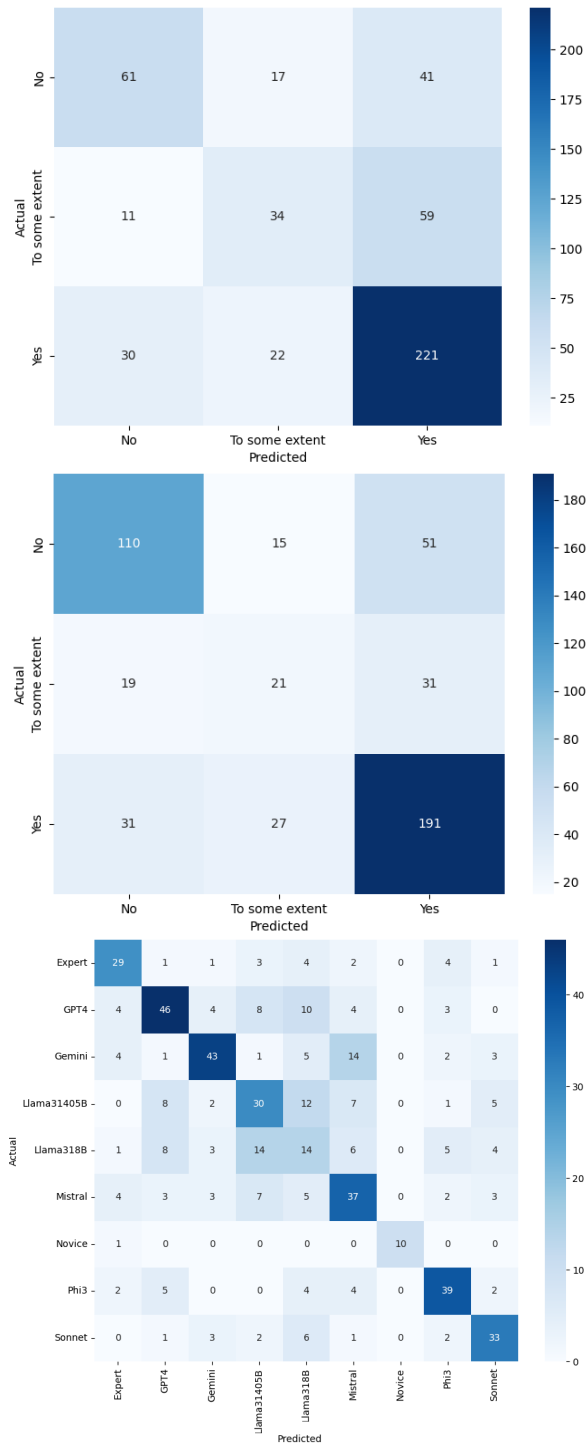


Figure 3: Per-class F1 scores for (a) Track 3, (b) Track 4, and (c) Track 5

Figure 2: Confusion matrices for (a) Track 3, (b) Track 4, and (c) Track 5, obtained by pairing gte-modernbert-base and multilayer perceptron (GTE + MLP)

BD at BEA 2025 Shared Task: MPNet Ensembles for Pedagogical Mistake Identification and Localization in AI Tutor Responses

**Shadman Rohan Ishita Sur Apan Muhtasim Ibteda Shochcho Md Fahim
Mohammad Ashfaq Ur Rahman AKM Mahbubur Rahman Amin Ahsan Ali**
Center for Computational & Data Sciences, Independent University, Bangladesh (IUB)
{shadmanrohan, ishitasurapan, sho25100, fahimcse381}@gmail.com
{imashfaqfardin}@gmail.com, {akmmrahman, aminail}@iub.edu.bd

Abstract

We present Team BD’s submission to the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered Tutors, under Track 1 (Mistake Identification) and Track 2 (Mistake Location). Both tracks involve three-class classification of tutor responses in educational dialogues – determining if a tutor correctly recognizes a student’s mistake (Track 1) and whether the tutor pinpoints the mistake’s location (Track 2). Our system is built on MPNet, a Transformer-based language model that combines BERT and XLNet’s pre-training advantages. We fine-tuned MPNet on the task data using a class-weighted cross-entropy loss to handle class imbalance, and leveraged grouped cross-validation (10 folds) to maximize the use of limited data while avoiding dialogue overlap between training and validation. We then performed a hard-voting ensemble of the best models from each fold, which improves robustness and generalization by combining multiple classifiers. Our approach achieved strong results on both tracks, with exact-match macro-F1 scores of approximately 0.7110 for Mistake Identification and 0.5543 for Mistake Location on the official test set. We include comprehensive analysis of our system’s performance, including confusion matrices and t-SNE visualizations to interpret classifier behavior, as well as a taxonomy of common errors with examples. We hope our ensemble-based approach and findings provide useful insights for designing reliable tutor response evaluation systems in educational dialogue settings.

1 Introduction

Effective intelligent tutoring systems need to be able to recognize and address student mistakes during interactions. To evaluate such capabilities in automated systems, the BEA 2025 Shared Task introduced a multi-dimensional assessment of AI tutor responses. In particular, Track 1 focuses on whether a tutor’s response identifies the student’s

mistake, and Track 2 on whether it locates the mistake in the student’s answer. Each track is framed as a three-way classification: the tutor either fully recognizes/locates the error (“Yes”), partially or uncertainly does so (“To some extent”), or fails to do so (“No”). These pedagogically motivated categories draw from prior frameworks in educational dialogue analysis—for example, Mistake Identification corresponds to the student understanding dimension in Tack and Piech’s schema (Tack and Piech, 2022b) and correctness in other tutoring evaluation schemata, reflecting how well the tutor acknowledges the student’s misconception.

Assessing tutor responses along such dimensions is challenging due to the nuanced and subjective nature of pedagogical feedback. For instance, different studies have used varied measures (e.g., “speaking like a teacher,” “understanding the student,” etc.) to judge tutor responses. The BEA 2025 shared task addresses this gap by defining clear categories and metrics for evaluation (Kochmar et al., 2025). However, even with a fixed taxonomy, classifying responses correctly remains non-trivial: tutors may implicitly acknowledge an error without stating it outright, or they might hint at the error’s location in vague terms. Distinguishing between a definite “Yes” and a tentative “To some extent” thus requires subtle interpretation of language.

In this paper, we describe Team BD’s ensemble-based MPNet system for automating the annotation of mistake identification and mistake location in AI-tutor responses. MPNet, a pretrained Transformer model that uses masked and permuted language modeling to capture token dependencies, was chosen as our backbone for its strong generalization capabilities compared to earlier models like BERT, XLNet, and RoBERTa. To address the limited size of the labeled data (approximately 2.5 K examples) and inherent class imbalance, we fine-tuned MPNet with a class-weighted cross-entropy loss and trained ten separate models using grouped cross-

validation—grouped by dialogue to prevent context leakage—and then combined the top-performing model from each fold through hard-voting. This ensemble strategy greatly improved robustness and generalization, leading to high accuracy and macro-F1 scores on both the mistake identification and mistake location tracks. Our error analysis using confusion matrices and t-SNE visualizations revealed consistent misclassification patterns, notably confusing fully recognized with partially acknowledged mistakes. We created a taxonomy of common error types with examples to aid future refinements.

2 Related Work

Evaluation of Tutor Responses: The task of judging tutor or teacher responses in educational dialogues has recently garnered attention. Tack and Piech (Tack and Piech, 2022a) introduced the AI Teacher Test to measure the pedagogical ability of dialogue agents, proposing dimensions such as whether the agent understands the student’s error and provides helpful guidance. Following this, the BEA 2023 Shared Task (Tack et al., 2023) focused on generating AI teacher responses (rather than classification), where models like GPT-3 and Blender were challenged to produce tutor-like feedback. The BEA 2025 Shared Task (Kochmar et al., 2025) moves a step further by creating a benchmark dataset of tutor responses annotated along multiple pedagogical dimensions. The dataset leverages dialogues from **MathDial** (Macina et al., 2023) and **Bridge** (Maurya et al., 2025), two collections of student-tutor interactions in the math domain. Each tutor response in these dialogues was labeled by experts as to whether it identifies the student’s mistake, pinpoints the mistake’s location, provides guidance, and offers actionable next steps. Such multi-faceted annotation of tutor feedback is relatively novel; it connects to earlier work on dialogue act classification (Maurya et al., 2025) in that both involve categorizing utterances, but here the labels are pedagogical quality ratings rather than communicative intent.

Ensemble Methods in NLP: Classic studies, such as Dietterich’s work on ensemble methods, demonstrated that an ensemble of diverse classifiers can correct individual models’ errors and reduce variance (Dietterich, 2000). For instance, (Ovadia et al., 2019) and (Gustafsson et al., 2020) found that deep ensembles improve reliability under dataset

shift. In shared task and kaggle competitions, top teams often resort to model ensembling to squeeze out some additional performance. These benefits come at the cost of increased computational overhead. Our approach aligns with this trend, as we build an ensemble of 10 MPNet-based classifiers (from cross-validation folds) to tackle the classification of tutor responses.

Dialogue and Educational NLP: Related to our work is research on grammatical error detection and correction, where systems identify mistakes in student-written text. Notably, (Ng et al., 2014) and (Bryant et al., 2019) have contributed significantly to this field. However, our task differs in that the “mistakes” are conceptual or procedural errors in a problem solution, and we are evaluating the tutor’s response to those errors rather than directly analyzing the student’s text. Another line of relevant work is on student response analysis in tutoring systems, where the goal is to classify student answers as correct, incorrect, or incomplete. (Dzikovska et al., 2013) explored this in the context of the SemEval-2013 Task 7. In our case, the roles are reversed—we classify the tutor’s replies. We also draw on insights from educational dialogue analysis: studies like (Daheim et al., 2024) examined tutor responses for targetedness and actionability, which correspond to our Track 2 and Track 4 tasks. These studies emphasize the subtle linguistic cues that indicate whether a tutor has pinpointed an error (e.g., referencing a specific step in the student’s solution) or just given generic feedback.

In summary, our work is situated at the intersection of dialogue evaluation and text classification. We build upon the shared task’s provided taxonomy (SIGEDU, 2025) and prior educational NLP research, employing modern Transformer models and ensemble techniques known to be effective in such tasks.

3 Data and Task Definition

Task Definition: Tracks 1 and 2 are classification tasks applied to tutor responses in a dialogue. Based on the previous conversation history between students and tutors, in Track 1 (Mistake Identification), the system must determine if the tutor’s response indicates recognition of the student’s mistake. In Track 2 (Mistake Location), the system judges if the tutor points out the specific location or nature of the mistake in the student’s solution. Both tasks share the same label set: **Yes, To some extent,**

Model	Macro-F1 Score
BERT-large	0.6851
DeBERTa	0.6845
MPNet (selected)	0.6975

Table 1: 10 fold Cross-validation Macro-F1 scores for different Transformer models on the track 1 development set. MPNet achieves the highest score.

or **No**. Because these categories can be nuanced, the shared task also defined a lenient evaluation where “Yes” and “To some extent” are merged, but our system is trained on the full 3-class distinction (exact evaluation).

Dataset: The training (development) data provided by the organizers consists of annotated educational dialogues in mathematics, drawn from the MathDial and Bridge datasets. Each dialogue includes a student’s attempt at a math problem (containing a mistake or confusion) and one or more tutor responses (from either human tutors or various LLMs such as Mistral, Llama, GPT-4, etc. acting as tutors). Each tutor response is annotated with the three-class labels for all four dimensions (Tracks 1–4). In total, the development set contains 300 conversation history and over 2,480 tutor responses with annotations. On average, each dialogue context yields 8–9 different tutor responses (one from each of several tutor sources), which were all annotated. The test set is constructed in the same way but uses held-out dialogues and responses—both the ground-truth labels and the tutors’ identities are hidden.

The development set for both **Track 1 (Mistake Identification)** and **Track 2 (Mistake Location)** consists of the same 300 dialogues and 2,476 tutor responses. However, the label distributions differ between tracks due to the nature of the classification tasks. The underrepresentation of the *To some extent* class in both tracks poses challenges for model learning.

4 Methodology

4.1 Preprocessing

All tutor responses and conversation histories were first lowercased (while preserving punctuation) to ensure consistent casing.

To standardize and sanitize the responses, we applied a series of targeted cleaning steps:

- **Extra Info Removal:** Eliminated any meta-data or annotations not part of the tutor’s ac-

tual reply.

- **Appended Dialogue Trimming:** Removed follow-up conversational turns that were appended after the original tutor response (e.g., speculative follow-up questions or acknowledgments).
- **Code Abstraction:** Replaced Python code blocks with the placeholder «python code» to retain structural intent while abstracting away executable details.
- **Punctuation Cleanup:** Stripped redundant or mismatched punctuation (e.g., extraneous quotes or dashes) that might confuse the tokenizer or the model.

Table 4 provides a summary of how many instances were affected by each category. We observed that models such as **Phi-3** and **Llama-3.1-405B** required the most extensive preprocessing.

Finally, each input example—consisting of the conversation history, cleaned response, and separator tokens—was constrained to a maximum of 512 MPNet tokens. In cases where the input exceeded this limit, we removed the low-value content (e.g., greetings or small talk) from the conversation history to retain the most relevant context.

4.2 Language Model Finetuning

In our experiments, we utilize transformer-based pretrained language models (LMs). Since these models may lack task-specific contextual knowledge, we fine-tune them on our target tasks to improve performance.

To begin, we consider a pretrained language model denoted as ϕ_{LM} . Each tutor’s response after preprocessing T is input to the model, yielding a sequence of tokens $T = \{t_{[CLS]}, t_1, t_2, \dots, t_n\}$ along with their corresponding layer-wise hidden representations $H^l = \{h_{[CLS]}^l, h_1^l, h_2^l, \dots, h_n^l\}$.

In our setup, we use the hidden representation of the [CLS] token from the final layer as the sentence-level representation of the input T , defined as:

$$h_T = \phi_{LM}(T)_{[CLS]}^L = H_{[CLS]}^L$$

This representation h_T is then passed through a classification head to produce the prediction. The classification head consists of a dropout layer *Drop* followed by a linear transformation:

$$p = W \cdot \text{Drop}(h_T) + b$$

Finally, we use a cross-entropy loss function to update the parameters of the language model ϕ_{LM} during training.

4.3 Grouped Cross-Validation

We employ *group cross-validation* to ensure robust evaluation and mitigate overfitting. In this approach, each dialogue (or group of dialogues) is entirely assigned to either the training or validation set within each fold, preventing shared context between the training and validation sets.

For each fold $f \in \{1, 2, \dots, k\}$, we define the training and validation sets as $\mathcal{G}_{\text{train}}^{(f)}$ and $\mathcal{G}_{\text{val}}^{(f)}$, respectively, where each set contains whole dialogues (or groups) with no overlap. We monitor the model’s performance on the validation set using the *macro-averaged F1 score* (macro-F1), which provides a balanced measure of performance across classes. For each fold, we save the model checkpoint that achieves the highest macro-F1 score on the validation set.

The final performance of the model is computed by aggregating the macro-F1 scores across all k folds.

4.4 Ensembling Strategy

To enhance model performance, we employed an *ensembling strategy* where the top-performing models from each fold were combined using hard voting. Specifically, for each track, we had a total of $N = 10$ models (one from each fold).

Let $\hat{y}_i^{(f)}$ denote the prediction of the model from fold f for the i -th sample, where $f \in \{1, 2, \dots, N\}$. The final prediction \hat{y}_i for each sample i was determined by majority vote:

$$\hat{y}_i = \text{mode}(\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \dots, \hat{y}_i^{(N)})$$

In the case of a tie, the tie-breaking rule was based on the average softmax confidence across all models. Let $s_i^{(f)}$ denote the softmax output (confidence) of the f -th model for the i -th sample. If a tie occurs, the final prediction is chosen as:

$$\hat{y}_i = \arg \max \left(\frac{1}{N} \sum_{f=1}^N s_i^{(f)} \right)$$

Ensembling helps to reduce variance and correct individual model biases, leading to more robust predictions. Our ensembling approach improved the macro-F1 score by 2–3 points over the performance of individual models.

5 Experimental Setup

5.1 Implementation Details

Model Selection In our experiments, we compared several such models—including BERT-large, DeBERTa, and MPNet—on a held-out subset of the training data. Among these, MPNet achieved the best macro-F1 score (see Table 1), and was thus selected as our backbone. For implementation details, including software, packages, and hardware setup, see Appendix A.

Model Hyperparameters

We used the AdamW optimizer with a learning rate of 2×10^{-5} , selected through preliminary experiments on a held-out validation set. This setting outperformed alternative learning rates such as 1×10^{-5} and 3×10^{-5} in terms of macro-F1. A linear learning rate decay schedule was used, along with early stopping based on validation macro-F1 (patience = 5 epochs). We trained with a batch size of 32 and applied a dropout rate of 0.1 in the classification head. No gradient accumulation was used.

Handling Class Imbalance

To mitigate class imbalance, we used a class-weighted cross-entropy loss, where the weight for each class c was computed as:

$$w_c = \frac{N}{K \cdot n_c}$$

with N being the total number of samples, K the number of classes, and n_c the count for class c . This formulation emphasizes underrepresented classes without overly penalizing frequent ones.

For Track 1 (Mistake Identification), class distributions were skewed toward “Yes” (1932), compared to “No” (370) and “To some extent” (174). We thus used the weight vector:

$$[w_{\text{No}}, w_{\text{Some}}, w_{\text{Yes}}] = [1.0, 3.0, 0.5]$$

to boost recall for the rare “Some extent” class and mildly down-weight the majority class.

In Track 2 (Mistake Location), the frequencies were: “Yes” (1504), “No” (732), and “To some extent” (240). Based on this, we used:

$$[w_{\text{No}}, w_{\text{Some}}, w_{\text{Yes}}] = [0.8, 2.2, 0.9]$$

These weights, derived from inverse class frequencies and lightly tuned, improved macro-F1 by reducing systematic underprediction of minority classes. Although not extensively optimized, this approach provided consistent performance gains across both tracks.

Track	Macro F1	Accuracy
<i>Track 1 – Mistake Identification</i>		
Best (BJTU)	0.718	0.862
Ours (Test)	0.711	0.877
Ours (CV aggregate)	0.685	0.869
<i>Track 2 – Mistake Location</i>		
Best (BLCU-ICALL)	0.598	0.768
Ours (Test)	0.554	0.714
Ours (CV aggregate)	0.560	0.700

Table 2: Comparison of our system’s macro-F1 and accuracy with top leaderboard scores on both tracks.

5.2 Evaluation Metrics

Following the shared task guidelines, we report both Accuracy and Macro F1. Macro F1, the un-weighted average of per-class F1 scores, is emphasized due to class imbalance. We monitored performance using these metrics on the validation set during training and evaluated on the aggregated development set using cross-validation predictions. Final test metrics were provided by the organizers. We focus on exact 3-class classification; lenient 2-class metrics (merging “Yes” with “To some extent”) were higher but are omitted here for brevity.

6 Result and Analysis

6.1 Main Result

To contextualize our system’s performance, we compared it against the top submissions from the official shared task leaderboard. On Track1 (Mistake Identification), our model achieved a macro-F1 of 0.711 on the test set, placing 5th out of 44 participating teams. The top-ranked system (BJTU) achieved a macro-F1 of 0.718, indicating that our system performs competitively, within 0.7 points of the best result. For Track2 (Mistake Location), our system scored 0.554 macro-F1 on the test set, ranking 7th out of 31 teams. The highest score on this track was 0.598, obtained by BLCU-ICALL. While our model trails behind the top result by approximately 4.4 points in macro-F1, it still exceeds the median leaderboard performance.

Our system achieved higher accuracy than the top Track 1 system (0.877 vs. 0.862), suggesting stronger performance on dominant classes, albeit with slightly lower balance across all classes.

Even though our system performs well, a closer examination of its errors provides insights into its decision-making and the task’s inherent difficulty. We carried out an error analysis on the development set predictions, focusing on confusion patterns and

the nature of misclassified cases.

6.2 Class-Level Performance Analysis

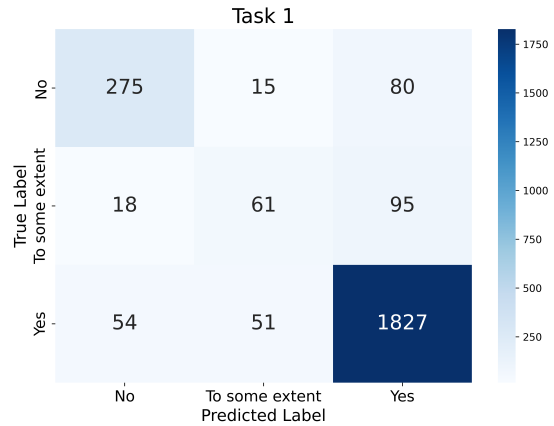


Figure 1: Confusion matrix for Track 1 (Mistake Identification) on the development set. The model shows strong performance on the "Yes" class but has difficulty distinguishing partial acknowledgment ("To some extent").

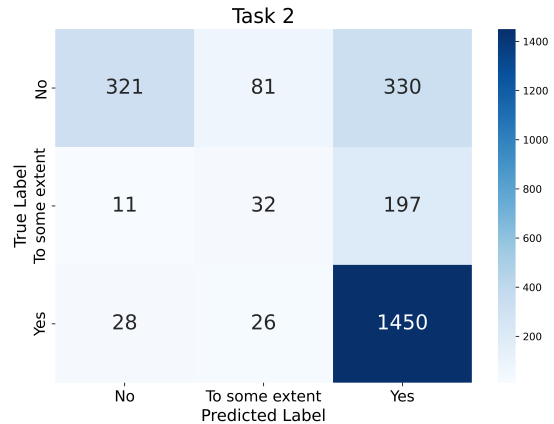


Figure 2: Confusion matrix for Track 2 (Mistake Location). The model maintains high accuracy on explicit localizations ("Yes") but misclassifies many "To some extent" and "No" cases, highlighting the subtlety of location inference.

To gain insight into how well our system distinguishes among the three pedagogical feedback categories, we analyze confusion matrices for both tasks. Figures 1 and 2 visualize model predictions against gold labels on the development set for Track 1 (Mistake Identification) and Track 2 (Mistake Location), respectively.

In Track 1 (Figure 1), the model performs strongly on the "Yes" class, correctly identifying 1,827 instances, with relatively low misclassification into the "No" (54) and "To some extent" (51)

classes. The "No" class is also well captured with 275 correct predictions and few false positives. The model struggles more with the "To some extent" category: 61 were correctly predicted, but 113 were misclassified as either "No" or "Yes." This aligns with our earlier claim that "To some extent" lies on a subjective continuum and is more difficult to pin down categorically.

For Track 2 (Figure 2), a similar trend emerges. The model again shows high accuracy on "Yes" (1,450 correct), but struggles to distinguish "To some extent," which is often misclassified as "Yes" (197 cases) or "No" (11 cases). Notably, the "No" class is less cleanly separated in Track 2 compared to Track 1, with 330 examples misclassified as "Yes." This may suggest that tutors sometimes appear to reference an error without pinpointing its location, confusing the model's judgment.

Overall, these confusion matrices illustrate the asymmetric difficulty across classes. "Yes" responses are most reliably predicted due to their clearer, more direct language. "To some extent" predictions remain a challenge, particularly when tutors use indirect or hedging phrasing that blurs the line between partial and full error acknowledgment or localization.

6.3 Embedding Space Insights

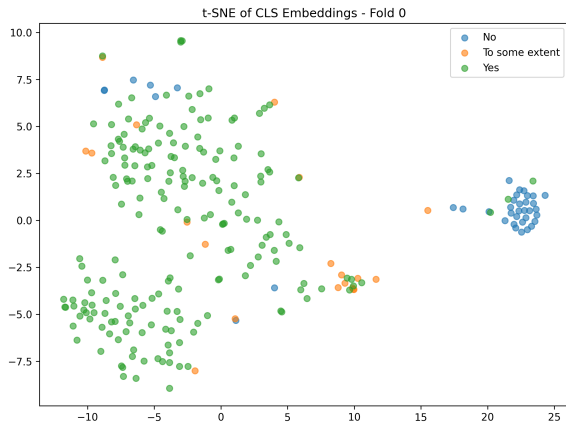


Figure 3: t-SNE projection of [CLS] embeddings from the held-out fold (Fold 0) for Track 1 (Mistake Identification), colored by true label. "Yes" and "To some extent" examples are scattered and intermingled, whereas "No" forms a more compact cluster, indicating lower intra-class variation.

To better understand the internal representations learned by our model, we applied t-SNE (van der Maaten and Hinton, 2008) to the [CLS] embeddings from the final Transformer layer. These pro-

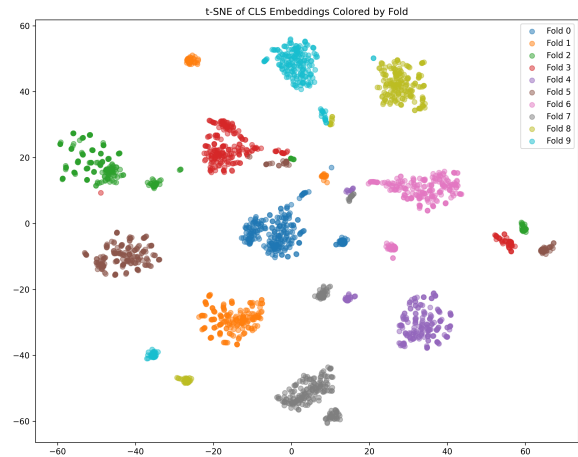


Figure 4: t-SNE projection of [CLS] embeddings from MPNet models across all 10 cross-validation folds for Track 1 (Mistake Identification). Each point represents a tutor response from a held-out fold, colored by fold ID. The emergence of distinct clusters suggests that each fold-specific model learns a consistent but fold-specific embedding subspace, reflecting representational diversity across the ensemble.

jections reveal how the model organizes tutor responses in the embedding space across folds and classes.

Figure 4 shows the t-SNE projection of the [CLS] embeddings across all ten cross-validation folds, with points colored by fold ID. We observe that embeddings from each fold tend to form compact, well-separated clusters. This indicates that while training on different subsets, each fold-specific model learns fold-consistent but distinct representations. The tightness of these clusters also suggests good embedding stability and coherence across training runs.

Figure 3 presents the t-SNE visualization for the held-out fold (Fold 0), this time colored by the true label. Unlike the per-fold visualization, class-level structure is less distinct: the "Yes" and "To some extent" responses are widely dispersed and often intermingle, suggesting overlapping semantic characteristics. In contrast, the "No" class forms a more compact group, indicating that tutor responses with no recognition of error share more consistent linguistic patterns. This aligns with our earlier findings that "Yes" and "To some extent" are harder to separate, as they exist on a continuum of acknowledgment.

Together, these visualizations support our earlier confusion matrix results and highlight the challenge of distinguishing nuanced pedagogical feed-

back categories based solely on language.

6.4 Error Taxonomy

To better understand where the model fails, we analyzed misclassified responses from both tasks and developed a taxonomy of recurring error types, summarized in Table 3. These categories reflect systematic issues in how the model interprets pedagogical language.

False Negatives (Missed Signal). These errors occur when the model fails to recognize that the tutor has identified or located a mistake, typically labeling the response as “No” or “To some extent” instead of “Yes.” Such cases often involve subtle cues like rhetorical questions or light correction phrasing (e.g., “Can you check the multiplication again?”), which the model may under-interpret.

False Positives (Over-interpretation). Here, the model predicts “Yes” even when the tutor does not provide evidence of error recognition. This often results from over-interpreting generic encouragement (e.g., “Let’s try another one.”) or positive sentiment as pedagogical feedback.

Partial–Full Confusion. A frequent source of confusion is the distinction between full and partial identification or localization. Indirect language such as “You’re close, just verify your subtraction” may be intended as partial feedback, but the model may treat it as a complete identification.

Hedged Language Confusion. Tutors often use polite or indirect language (e.g., “Maybe revisit the earlier step?”), especially in educational settings. Such hedging may obscure intent, leading the model to underestimate the strength of the feedback signal.

Contextual Miss. Some misclassifications stem from failing to use conversational history. For instance, if a tutor’s comment refers to an earlier incorrect step, the model may mislabel it when that context is not incorporated effectively.

Template Bias. We also observed that the model sometimes over-relies on surface patterns seen during training. For example, statements like “Great work!” may be incorrectly classified as “Yes” due to template bias, even when no mistake is acknowledged.

These error categories offer valuable insight into the linguistic and contextual challenges of the task. They suggest that improvements in discourse modeling, uncertainty handling, and pragmatic language understanding could further enhance performance.

From the above taxonomy, we see that many of the model’s mistakes correspond to understandable difficulties. False negatives often involved indirect tutor feedback—the tutor recognized the mistake but phrased it as a question or hint, requiring inference to identify it as an acknowledgment of error. Our model sometimes took such tentative language at face value and labeled it as if the tutor did nothing. False positives, on the other hand, were cases where the tutor’s response had reassuring or neutral language that the model mistook for a sign of recognizing a mistake. For example, tutors might say “Let’s double-check that” even when the student was correct (encouraging the student, not pointing an error), and the model erroneously flagged it as identifying an error.

The partial vs. full confusion category was the most prevalent error type. This reflects the inherent ambiguity of the “To some extent” class—even human annotators might differ on these in some cases. Our model would sometimes collapse it into one of the binary decisions (“Yes” or “No”) depending on slight wording differences. In some cases, the model predicted “To some extent” when the tutor had actually pinpointed the error but perhaps in a subtle way; in others, it predicted “Yes” for a tutor response that was only hinting. This suggests that improving the model’s understanding of nuanced language (perhaps via better context usage or training on more examples of hedging) could help.

We also found that ambiguous wording and polite phrasing (common in educational settings) posed challenges. Phrases like “Maybe check that again” require contextual understanding—they might indicate an error without explicit wording. Our model did catch many of these, but not all. Some errors could be attributed to the model’s lack of world knowledge or reasoning; for example, if a tutor says “Remember the formula for area,” the model needs to infer that the student likely made a mistake related to area calculation and that the tutor is hinting at it—a level of reasoning beyond surface text.

In summary, the error analysis reveals that while our ensemble is effective, there is room for improvement in handling borderline cases and understanding implicit signals. These findings guided us in considering potential enhancements, as discussed next.

Error Type	Description	Example Scenario
False Negative (Missed Signal)	Tutor indicates or locates a mistake, but the model predicts “No” or “To some extent.”	<i>Tutor:</i> “Can you check the multiplication again?” <i>Gold:</i> Yes → <i>Pred:</i> To some extent
False Positive (Over-interpretation)	Model predicts “Yes” despite the tutor giving no error feedback.	<i>Tutor:</i> “Let’s try another one.” <i>Gold:</i> No → <i>Pred:</i> Yes
Partial–Full Confusion	Confuses indirect hints as full identification, or subtle localization as partial.	<i>Tutor:</i> “You’re close, just verify your subtraction.” <i>Gold:</i> To some extent → <i>Pred:</i> Yes
Hedged Language Confusion	Tutor’s suggestion is misread due to polite phrasing or indirect cues.	<i>Tutor:</i> “Maybe revisit the earlier step?” <i>Gold:</i> Yes → <i>Pred:</i> To some extent
Contextual Miss	Misclassification caused by ignoring or misusing multi-turn context.	<i>Tutor:</i> Feedback depends on an earlier step, but the model misses the reference.
Template Bias	Model favors phrases resembling training-time patterns, even when semantically incorrect.	<i>Tutor:</i> “Great work!” with no correction. Model assumes this implies error recognition.

Table 3: Taxonomy of common misclassification errors in both tasks, with representative examples.

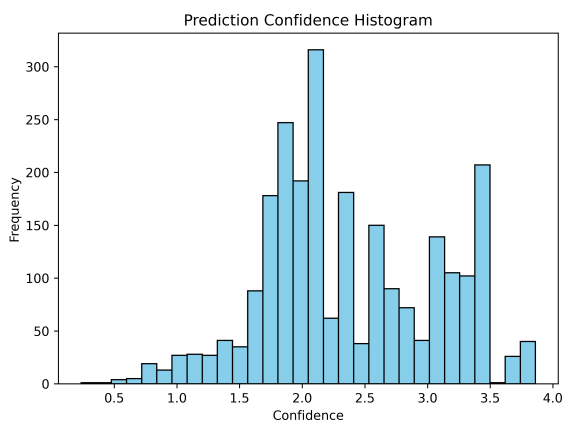


Figure 5: Histogram of prediction confidence values for Track 1 (Mistake Identification). Most predictions fall within a mid-confidence range.

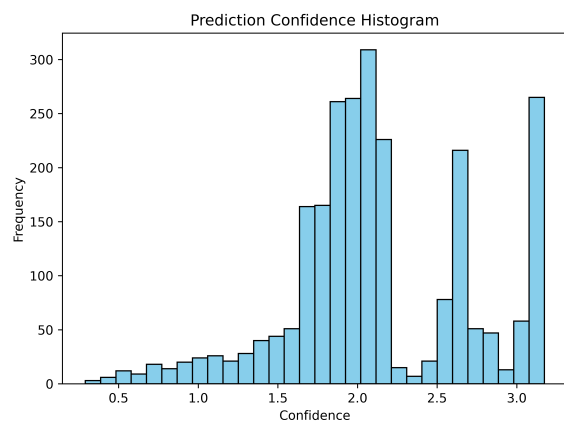


Figure 6: Histogram of prediction confidence values for Track 2 (Mistake Location). A similar mid-range clustering pattern is observed, with some extreme confidence peaks.

6.5 Confidence Distribution and Calibration

To further investigate the model’s decision-making behavior, we analyzed its prediction confidence across classes and tasks. Figures 5 and 6 present histograms of predicted confidence scores for Track 1 and Track 2, respectively. These reflect the model’s certainty in its predictions across the development set.

In both tasks, the confidence distribution is skewed toward the middle range (1.5–3.0), with multiple local peaks. This suggests that while the model often makes moderately confident predictions, it does not frequently commit to extremely low or high confidence outputs. The spiked clusters in Track 2 (Figure 6) hint at calibration artifacts possibly introduced by ensemble averaging. Despite ensemble smoothing, we still observe confidence saturation for some predictions near 3.5, particularly on easier instances.

To better understand class-specific behavior, we examined boxplots of prediction confidence grouped by predicted label (Figures 7 and 8). In both tasks, predictions labeled as “No” tend to have higher median confidence compared to “To some extent,” reflecting that the model is more certain when asserting a complete absence of error. Predictions for “To some extent” exhibit both lower median confidence and greater spread—supporting earlier findings that this category is harder to classify due to its inherent ambiguity. Interestingly, in Track 1, “Yes” predictions also show relatively high confidence, indicating that the model treats full error recognition as a more decisive signal than partial acknowledgment.

These confidence trends are broadly aligned with our confusion matrix analysis: “To some extent” is not only the most frequently confused class but also the one with the least confident predictions. This

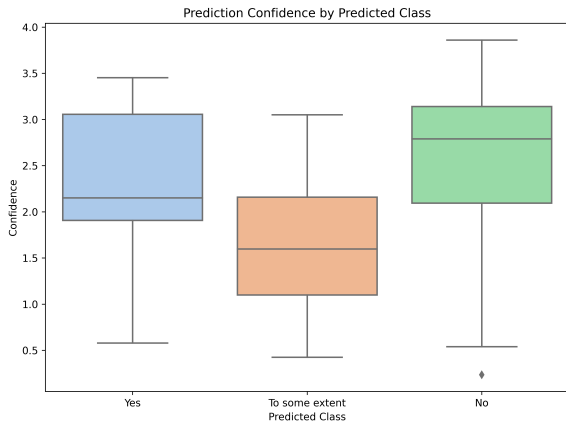


Figure 7: Boxplot of confidence by predicted class (Track 1). Predictions labeled “To some extent” tend to have lower median confidence.

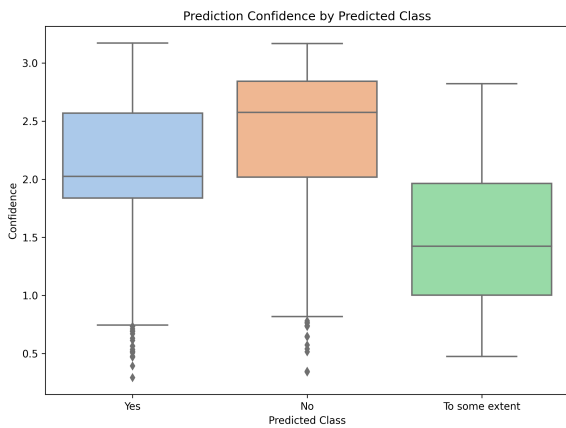


Figure 8: Boxplot of confidence by predicted class (Track 2). “No” and “Yes” predictions show higher confidence than “To some extent.”

highlights a key challenge in pedagogical feedback modeling—the need to model uncertainty explicitly, especially in borderline cases. Future work could explore temperature scaling or Bayesian ensembling to better calibrate prediction confidence, particularly for interpretability in high-stakes educational settings.

7 Conclusion

This paper presents Team BD’s ensemble-based MPNet system for the BEA 2025 Shared Task on Mistake Identification and Location in tutor responses. By fine-tuning MPNet with class-weighted loss and grouped cross-validation, we addressed data imbalance and maximized the use of training data, achieving high accuracy and macro-F1 scores on both Track 1 and Track 2. Extensive analyses show that, while the model reliably han-

dles clear-cut error recognition, it struggles with borderline cases involving partial acknowledgment, as evidenced by embedding-space visualizations and a taxonomy of common errors. Future work will explore multi-task learning across evaluation dimensions, leverage larger language models or adapter-based methods to incorporate LLM knowledge, and improve calibration and domain-specific contextual understanding to enhance system reliability and interpretability.

8 Limitations

Despite the strong results achieved by our ensemble MPNet-based system, several limitations warrant discussion:

Confidence Calibration: Our ensemble exhibits poor calibration, often assigning high confidence to incorrect predictions—problematic for intervention-triggering systems. We did not apply calibration methods due to time constraints. Adaptive Temperature Scaling (ATS), a recent post-hoc technique, improves token-level calibration by 10–50% across benchmarks (Xie et al., 2024), and merits future exploration.

Label Ambiguity: The line between “Yes” and “To some extent” is subjective, with some errors stemming from annotation uncertainty rather than model failure, thus limiting performance. Modeling the task as ordinal or probabilistic may better capture this continuum; ordinal methods have been proposed for similar label structures (Zhang et al., 2023).

Model Scope and Efficiency: MPNet-base lacks domain-specific specialization for educational dialogue, which may limit its ability to handle nuanced interactions. Exploring a larger, domain-adapted backbone or a multitask learning setup could enhance performance and is a promising direction for future work.

Acknowledgments

We thank the BEA 2025 Shared Task organizers for their efforts and Center for Computational & Data Sciences (CCDS) for supporting this work. Our code and models are available at: <https://github.com/ShadmanRohan/team-bd-bea25>

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. Association for Computational Linguistics.
- Nico Daheim, Jakub Macina, Tanmay Sinha, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. *arXiv preprint arXiv:2407.09136*.
- Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15. Springer.
- 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. Association for Computational Linguistics.
- Fredrik Gustafsson, Martin Danelljan, and Thomas B. Schön. 2020. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 169–170.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*, Vienna, Austria. Association for Computational Linguistics.
- 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. 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.
- 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. Association for Computational Linguistics.
- Yaniv Ovadia, Elad Fertig, Jae Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, and Balaji Lakshminarayanan. 2019. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*.
- Nils Reimers and Iryna Gurevych. 2020. sentence-transformers/all-mpnet-base-v2. <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>.
- SIGEDU. 2025. Be a 2025 shared task: Pedagogical ability assessment of ai-powered tutors. <https://sig-edu.org/sharedtask/2025>.
- 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)*.
- Anaïs Tack and Chris Piech. 2022a. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. In *Proceedings of the 15th International Conference on Educational Data Mining (EDM)*.
- Anaïs Tack and Chris Piech. 2022b. An evaluation taxonomy for pedagogical ability assessment of llm tutors. *arXiv preprint arXiv:2412.09416*.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.
- 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. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.
- Johnathan Xie, Annie S. Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. 2024. Calibrating language models with adaptive temperature scaling. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Yue Zhang, Wei Wang, and Xiaojun Wan. 2023. Boosting language-driven ordering alignment for ordinal classification. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*.

Appendix

A Software and Package Details

We conducted all experiments using Python 3.9, PyTorch 1.13, and the Hugging Face Transformers library (version 4.37.2) (Wolf et al., 2020). Specifically, we fine-tuned the sentence-transformers/all-mpnet-base-v2 model available on the Hugging Face Model Hub (Reimers and Gurevych, 2020). Tokenization was performed using MPNet’s tokenizer, with inputs truncated to a maximum length of 300 tokens.

All models were trained on a single NVIDIA RTX 3090 GPU (24 GB). Each fold took approximately 2–4 minutes per epoch to train, with convergence typically reached within 3 epochs (i.e., 6–12 minutes per model). Full ensemble training (10 models for Track 1 and 7 for Track 2) completed in under 3 hours. Despite the ensemble size, inference was efficient: classifying the entire test set (several hundred responses) took under 30 seconds.

B Training Configuration

Class Weights. To mitigate class imbalance, we applied inverse frequency class weighting in the cross-entropy loss function:

$$w_c = \frac{1}{\log(f_c + \epsilon)},$$

where f_c is the frequency of class c and $\epsilon = 1.05$.

Hyperparameter Search. We performed grid search over learning rates $\{1e-5, 2e-5, 3e-5\}$ and batch sizes $\{8, 16\}$. The best configuration was selected based on average macro-F1 over the cross-validation folds.

Reproducibility. We fixed all random seeds to 42 and set PyTorch to deterministic mode. Our code will be made publicly available upon publication.

C Preprocessing Frequency Across Models

Table 4 summarizes the frequency of manual cleanup operations required across models.

D Additional Training Results

Table 5 reports additional macro-F1 scores for Mistake Identification and Mistake Location tasks across various models. For non-Transformer models, we used TF-IDF representations as input features.

Category	Phi3	Mistral	Llama-3.1-8B	Llama-3.1-405B	GPT-4	Total
Extra Info	1	0	1	11	1	14
Appended Dialogue Trimming	19	0	0	0	0	19
Code Abstraction	2	0	0	0	0	2
Punctuation Cleanup	3	2	0	0	0	5
Totals	25	2	1	11	1	40

Table 4: Model-specific frequencies of manual cleanup operations on tutor responses.

Model	Mistake Identification	Mistake Location
BERT	0.8703	0.7025
RoBERTa	0.7816	0.6551
DeBERTa	0.8576	0.7025
ELECTRA	0.8513	0.6266
MPNet	0.8639	0.6203
NeoBERT	0.8513	0.6677
Logistic Regression	0.7880	0.6139
Random Forest	0.8260	0.6551
Gradient Boosting	0.8418	0.6519
SVM	0.7785	0.6110
LightGBM	0.8418	0.6551
XGBoost	0.8386	0.6646
CatBoost	0.8196	0.6582

Table 5: Macro-F1 scores for Mistake Identification and Mistake Location tasks across Transformer models and TF-IDF + traditional classifiers. Best results per column are bolded.

Thapar Titan/s : Fine-Tuning Pretrained Language Models with Contextual Augmentation for Mistake Identification in Tutor–Student Dialogues

Harsh Dadwal¹, Sparsh Rastogi¹, Jatin Bedi¹

¹Thapar Institute of Engineering and Technology, Patiala, India
hdadwal_be22@thapar.edu, srastogi_be22@thapar.edu, jatin.bedi@thapar.edu,

Abstract

This paper presents Thapar Titan/s’ submission to the BEA 2025 Shared Task on Pedagogical Ability Assessment of AI-powered Tutors (Kochmar et al., 2025). The shared task consists of five subtasks; our team ranked 18th in Mistake Identification, 15th in Mistake Location, and 18th in Actionability. However, in this paper, we focus exclusively on presenting results for Task 1: Mistake Identification, which evaluates a system’s ability to detect student mistakes.

Our approach employs contextual data augmentation using a RoBERTa based masked language model to mitigate class imbalance, supplemented by oversampling and weighted loss training. Subsequently, we fine-tune three separate classifiers: RoBERTa, BERT, and DeBERTa for three-way classification aligned with task-specific annotation schemas. This modular and scalable pipeline enables a comprehensive evaluation of tutor feedback quality in educational dialogues.

1 Introduction

With the rapid evolution of large language models (LLMs), their integration into the educational domain has expanded significantly. These models present a transformative opportunity to enhance equitable access to high-quality education, especially in remote or under-resourced areas where there is a persistent shortage of qualified educators. When implemented as AI-powered tutors, LLMs can facilitate interactive, human-like dialogues that potentially overcome the constraints of conventional educational tools and enable scalable, personalized learning experiences.

Nonetheless, despite their promise, current LLMs exhibit several notable limitations. They are susceptible to inherent biases derived from their training data, often display reduced reliability in solving mathematical problems requiring struc-

ture reasoning, and are prone to generating hallucinated or factually inaccurate responses. These deficiencies raise critical concerns about their dependability in educational settings where accuracy and clarity are paramount. Consequently, there is a growing imperative to establish rigorous and systematic frameworks for assessing the pedagogical efficacy of state-of-the-art generative models in the context of educational dialogues. Evaluating the pedagogical capabilities of generative models is crucial because AI tutors must do more than coherent dialogue generation, they need to provide accurate, constructive, and context-sensitive guidance that supports effective learning. This is especially important in mathematics and reasoning tasks, where precise problem-solving steps and logical explanations are essential. Without assessing these educational qualities, models may produce plausible but incorrect or misleading responses. Therefore, rigorous evaluation of pedagogical effectiveness is vital to ensure AI tutors genuinely enhance learning and meet educational standards.

Due to the absence of a unified evaluation framework, prior studies have adopted a variety of criteria to assess the effectiveness of AI tutoring systems. For instance, (Tack et al., 2023) and (Tack and Piech, 2022) focused on whether the model communicates like a teacher, understands student needs, and offers helpful guidance. (Macina et al., 2023) employed human evaluators to judge responses based on coherence, correctness, and fairness in tutoring. Meanwhile, (Wang et al., 2024) emphasized usefulness, empathy, and human-likeness, and (Daheim et al., 2024) assessed responses using targetedness, correctness, and actionability.

To address these challenges, this paper presents a classification approach based on fine tuning three pretrained language models RoBERTa (Zhuang et al., 2021), DeBERTa (He et al., 2021), and BERT (Devlin et al., 2019) designed to understand the

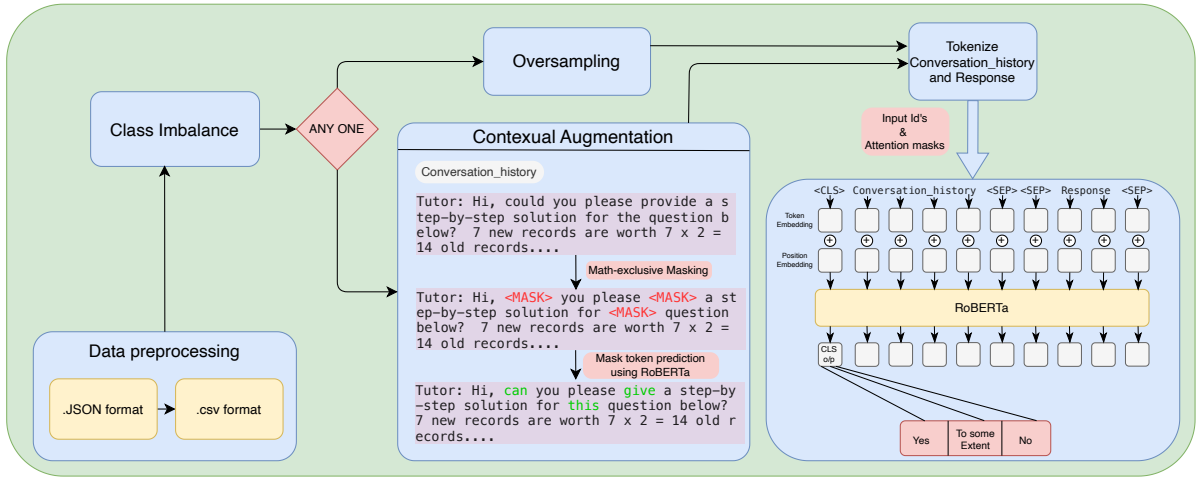


Figure 1: An schematic representation of the overall methodology

underlying context of educational dialogues and accurately identify student mistakes. To mitigate the inherent class imbalance in the dataset where certain types of errors are more frequent, our approach incorporates weighted training and contextual augmentation, ensuring the models do not develop internal biases toward specific mistake categories. Subsequent sections provide a detailed account of our methodology and findings.

2 Methodology

This work formulates the task of mistake classification in tutor–student dialogues as a multiclass classification problem. To address the pronounced class imbalance in the dataset, two complementary strategies were employed: conventional oversampling and contextual augmentation based on masked language modeling. The resulting balanced dataset was used to fine-tune transformer based models such as BERT, RoBERTa, and DeBERTa, with all layers unfrozen to facilitate effective weight optimization. The models were trained using categorical cross entropy loss and evaluated using macro F1 score and accuracy, with early stopping implemented based on macro F1. A detailed breakdown of this methodology is illustrated in Fig. 1 and further elaborated in the subsequent sections.

2.1 Dataset

We utilize the official dataset released as part of the BEA Shared Task 2025 (Maurya et al., 2025), comprising dialogues sourced from the MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024) datasets. The development set includes 300 dialogues, each consisting of several preceding tu-

tor–student turns where the student either makes a mistake or expresses confusion, followed by the student’s latest utterance and a set of tutor responses. These responses include those from human tutors extracted from the original datasets, as well as responses generated by seven LLMs-as-tutors, each identified by a unique model ID. In total, the development set contains over 2,480 tutor responses, each annotated for pedagogical quality. The annotations span three classes: yes, to some extent, and no, indicating whether the tutor successfully performs a given pedagogical function. However, the distribution is highly imbalanced, with approximately 78% of examples labeled as yes, 7% as to some extent, and only 14% as no. This skew poses a significant challenge, as it can lead to bias in model fine tuning if not properly addressed. The data is provided in JSON format with fields such as conversation id, conversation history, tutor responses, and annotations. The test set comprises 200 similarly structured dialogues from the same sources, containing unannotated responses from the same set of tutors, with tutor identities and pedagogical annotations withheld.

2.2 Data Augmentation

To address the severe class imbalance in the dataset, two complementary strategies were employed. The first involved conventional oversampling, in which the frequency of each example from the minority classes was increased by duplicating existing instances. Although this approach provided some improvement, it introduced a risk of overfitting due to repeated exposure to identical inputs. To mitigate this issue, contextual augmentation was also ap-

plied to generate diverse and meaningful examples for the underrepresented classes. A semantic masking approach was adopted, where selected words in the conversation history, which represents the student’s input to the model, were masked while preserving domain specific terms and mathematical symbols. These key terms were excluded because they carry essential meaning and detail, which are critical for accurately assessing a tutoring scenario. Irrelevant stopwords were also omitted, as they do not contribute significant semantic content and would not enhance the quality of augmentation. After masking, we applied masked language modeling using a pretrained RoBERTa based model. These models predicted and replaced the masked tokens based on their surrounding context, generating fluent and semantically consistent variations of the input. By leveraging multiple models, we introduced a rich set of plausible alternatives while preserving the original intent of the student’s question. Importantly, this augmentation was applied only to the input context and not to the tutor’s response. Altering the responses could distort the assessment of the model’s true predictive performance. This method allowed us to expand the dataset meaningfully, improve class balance, and maintain the authenticity of pedagogical evaluation.

2.3 Fine-Tuning for Classification

The overall problem was formulated as a multiclass classification task focused on identifying and localizing different types of mistakes within student-tutor dialogues. Three large language models, namely BERT large, RoBERTa, and DeBERTa, were chosen for fine-tuning due to their strong contextual understanding and performance in natural language tasks. The training was conducted using the final augmented dataset, which contained approximately 2,000 samples for each class to address the class imbalance and ensure balanced learning. To maximize performance, all layers of the models were unfrozen, allowing for comprehensive weight adjustment during training. The models were trained for up to 100 epochs on an Nvidia H100 GPU, with categorical cross entropy serving as the optimization loss function. Evaluation was performed using macro F1 score and accuracy metrics. Early stopping was applied based on the macro F1 score to prevent overfitting, and the best model weights were saved for subsequent evaluation.

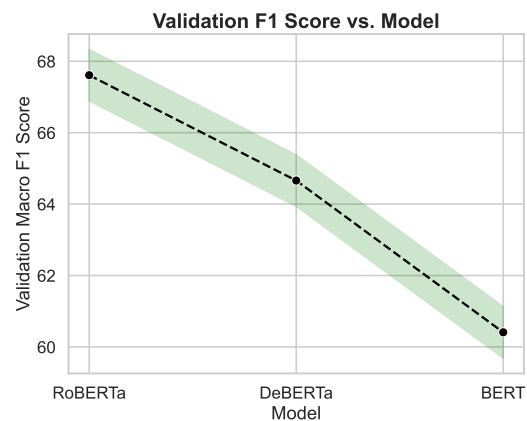


Figure 2: Model-wise Comparison of Validation Macro F1 Scores

3 Results and Discussion

Extensive experimentation was conducted across various hyperparameters and settings to assess their individual impact on model performance. RoBERTa was fixed as the baseline/default architecture for all experiments, and the mask ratio was set to a default of 15%, except where explicitly varied during the mask ratio ablation studies. The experiments focused on three key areas: evaluating different mask ratios during contextual masking (15%, 30%, and 50%), comparing transformer architectures (RoBERTa, BERT, and DeBERTa) at the default 15% mask ratio, and investigating two class imbalance handling techniques—contextual augmentation and conventional oversampling. We

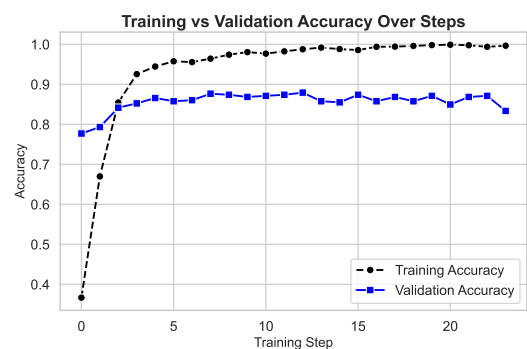


Figure 3: Training and Validation Accuracy over Optimization Steps

observed that the model achieved the highest performance with the default 15% mask ratio, yielding a training accuracy of 99.63%, validation accuracy of 87.9%, and validation F1 score of 67.61% (Fig. 3 and Fig. 4). Increasing the mask ratio to 30% and 50% led to a slight decrease in all performance metrics, with the lowest F1 scores observed at the

S. No.	Metric	Contextual Augmentation	Conventional Oversampling	Class Weights
1	Train Accuracy	99.63	100.00	99.77
2	Validation Accuracy	87.90	81.40	81.67
3	Validation F1 Score	67.61	63.75	63.48

Table 1: Performance comparison across different data augmentation and class imbalance handling techniques.

50% masking level (65.47%), as shown in Fig. 5. This indicates that excessive masking may hinder the model’s ability to learn meaningful contextual representations, while the 15% mask ratio strikes an effective balance between regularization and information retention, enhancing generalization on the validation set.

Using the fixed baseline RoBERTa model at the default mask ratio, we compared the performance of different transformer architectures. RoBERTa and DeBERTa demonstrated superior results, with validation accuracies of 87.9% and 87.1%, respectively. RoBERTa slightly outperformed DeBERTa in validation F1 score (67.61% vs. 64.66%). BERT lagged with a validation accuracy of 81.4% and an F1 score of 60.41%. The stronger performance of RoBERTa and DeBERTa is attributable to their improved pre-training methods and architectural enhancements compared to BERT, facilitating better contextual understanding (Fig. 2).

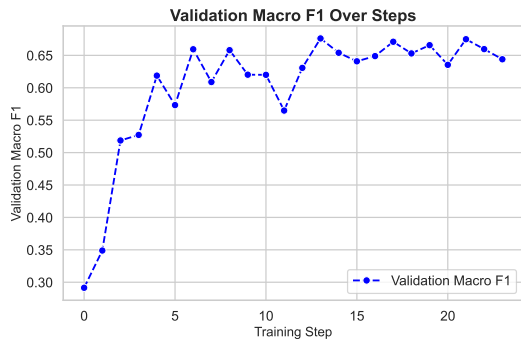


Figure 4: Validation Macro F1 Score Across Training Steps

For handling class imbalance, contextual augmentation and conventional oversampling were evaluated. While oversampling achieved perfect training accuracy (100%), it produced lower validation accuracy (81.4%) and F1 score (63.75%) compared to contextual augmentation (validation accuracy 87.9%, F1 67.61%), as shown in Table 1. This suggests that oversampling may lead to overfitting, whereas contextual augmentation, by generating semantically consistent synthetic samples, improves model generalization without overfitting.

Overall, these results emphasize the importance

of choosing an appropriate mask ratio, selecting advanced transformer architectures, and using semantically informed augmentation techniques for robust model performance. Fixing RoBERTa as the baseline and adopting a 15% mask ratio proved effective across experiments. The findings highlight the necessity of careful hyperparameter tuning and data augmentation strategies, especially when addressing class imbalance. Future research may explore integrating these techniques further and evaluating them on larger, more diverse datasets.

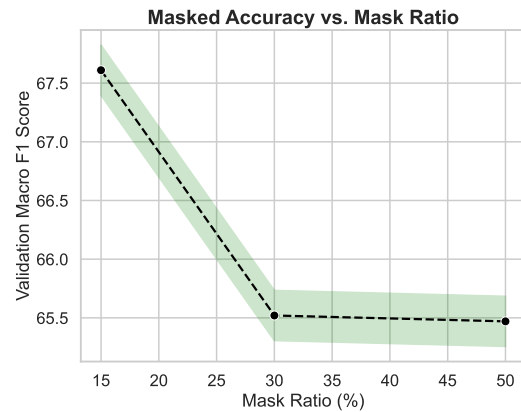


Figure 5: Effect of Masking Ratio on Validation Macro F1 Score

4 Conclusion

This study addresses the task of mistake classification in tutor–student dialogues by fine-tuning large pre-trained language models on a class-balanced dataset. To mitigate the issue of severe class imbalance, both conventional oversampling and contextual augmentation were employed, preserving the semantic integrity of student inputs. The use of BERT, RoBERTa, and DeBERTa enabled effective learning, and performance was evaluated using macro F1 and accuracy. Overall, the proposed framework enhances the reliability and generalizability of automated feedback systems. Future work may explore adaptive augmentation or dynamic feedback integration to further improve model robustness.

Limitations

This study is based on publicly available datasets, specifically MathDial and Bridge, which may not capture the full range of tutoring scenarios encountered in real-world educational settings. As a result, the model’s performance and generalizability could be limited when applied to more diverse or complex dialogues beyond these datasets. Furthermore, while contextual augmentation was effective in mitigating class imbalance by generating additional examples for minority classes, this approach may inadvertently introduce subtle biases or produce variations that are not entirely representative of natural student language. Such synthetic alterations, although contextually coherent, might affect the model’s robustness when faced with truly novel or unexpected inputs. Future studies could address these limitations by incorporating more diverse dialogue datasets and exploring augmentation strategies that more closely mimic real-world student behavior and language use.

References

- Nico Daheim, Nouha Dziri, Mrinmaya Sachan, Iryna Gurevych, and Edoardo Ponti. 2024. [Elastic weight removal for faithful and abstractive dialogue generation](#). 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 7096–7112, Mexico City, Mexico. 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.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [DeBERTa: Decoding-enhanced bert with disentangled attention](#). *Preprint*, arXiv:2006.03654.
- 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. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*, Online. Association for Computational Linguistics.
- 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.
- 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.
- 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.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. [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.
- Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. [A robustly optimized BERT pre-training approach with post-training](#). In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Author Index

- Aftahee, Sabik, 1127
Ahmad, Sarfraz, 1254
Ahsan, Momina, 1254
Akter, Syeda Sabrina, 460
Alfter, David, 186, 326
Alhafni, Bashar, 549
Ali, Amin, 1266
Allkivi, Kais, 953
Almasi, Mina, 70
AMIN, MD AL, 1127
Amjad, Maaz, 477
An, Jiyuan, 1084
Anand, Christopher, 582
Anastasopoulos, Antonios, 460
Andersen, Nico, 660
Antoun, Wissam, 1203
Araya, Roberto, 38
Arronte Alvarez, Aitor, 384
Asano, Yuya, 716
- Baier, Jorge, 1187
Baldwin, Peter, 830, 891
Bang, Jinhyun, 1049
Banno, Stefano, 632
Bantilan, Hans, 968
Barriere, Valentin, 38
Basem, Mohmaed, 1194
Bedi, Jatin, 1278
Beigman Klebanov, Beata, 716
Benedetto, Luca, 55
Bexte, Marie, 144, 225, 375
Bhatt, Krishnakant, 258
Bhattacharyya, Souvik, 1180
Bisliouk, Artem, 882
Bloch, Louise, 334
Bosselut, Antoine, 279
Bouwer, Renske, 535
Briscoe, Ted, 549
Brunato, Dominique, 708
Buc, Cristian, 38
Buca, Mihnea, 24
Bucur, Ana-Maria, 780
Butt, Sabur, 477
Buttery, Paula, 55
- Caines, Andrew, 213
Castañeda-Garza, Gerardo, 477
Cesaroni, Valeria, 505
- Chen, Gaowei, 1040
Chen, Lei, 1034
Chen, Longfeng, 1078
Chen, Nancy, 129
Chen, Ruishi, 908
Chen, Xiaobin, 978
Chifligarov, Mihail, 237
Chirkunov, Kirill, 549
Chiruzzo, Luis, 1135
Chitez, Madalina, 780
Choi, Gihyeon, 1145
Choi, Jinho D., 805
Chukharev, Evgeny, 841
Clauser, Brian, 830
Collins, Christopher, 446
Correa Busquets, Sofia, 1187
Correnti, Richard, 752
Csuros, Karla, 11
Córdova Véliz, Valentina, 1187
- D’Addario, Angelo, 891
Dadwal, Harsh, 1278
Daheim, Nico, 1, 100
Dainese, Nicola, 564
Dascalescu, Stefan, 89
de Chillaz, Aymeric, 279
De Kuthy, Kordula, 175
De Vrindt, Michiel, 535
Degraeuwe, Jasper, 312
DEKMAK, FATIMA, 1203
Desarkar, Maunendra Sankar, 1242
Dill, Alexander, 237
Ding, Yuning, 225, 345
Dinu, Liviu, 780
Drachsler, Hendrik, 1173
Drackert, Anastasia, 237
Dumitran, Marius, 24, 89
Dunn, Karen, 160
Dzienisiewicz, Daniel, 118
- Elaraby, Mohamed, 672
Elkordi, Hossam, 1121
- Fahim, Md, 1266
Fan, Hanghang, 1084
Fan, Yuming, 1073
Felice, Mariano, 160
Fernandez, Nigel, 294

Fishel, Mark, 953
 Flor, Michael, 398
 Frassinelli, Diego, 266
 Frederick Eneye, Tania Amanda Nkoyo, 477
 Friedrich, Christoph, 334
 Fu, Xiang, 1084

Gales, Mark, 632
 Galletti, Martina, 505
 Ganguli, Shrenik, 1242
 Gao, Qingyu, 202
 Gardner, Dominic, 375
 Geng, Tianyi, 186
 Girrback, Leander, 175
 Goh, Dion Hoe-Lian, 129
 Goldhammer, Frank, 660
 Gombert, Sebastian, 660, 1173
 Gonzales, Mark Edward, 1260
 Gralinski, Filip, 118
 Gu, Yang, 202
 Gulczyński, Michał, 794
 Gupta, Pranav, 1180
 Gupta, Vansh, 612
 Góngora, Santiago, 1135

Ha, Le An, 891
 Habash, Nizar, 549
 Hamdí, Ali, 1194
 Han, Junzhi, 805
 Hanif, Ikhlasul, 1212
 Harik, Polina, 830, 891
 Hellas, Arto, 564, 873
 Hikal, Baraa, 1194
 Horbach, Andrea, 225, 345, 660, 818
 Hou, Jue, 594, 737
 Houriet, Andrew, 891
 Hovy, Dirk, 356
 Huang, Zeyu, 1078
 Huang, Zihao, 202
 Hunter, Seth, 460
 Huovinen, Leo, 1002
 Hämäläinen, Mika, 1002

Ijezue, Chukwuebuka Fortunate, 477
 Ikram, Fareya, 765
 Imam Amjad, Ahmad, 477
 Iqbal, Hasan, 1254
 Ispas, Jany-Gabriel, 1224

Jain, Raunak, 1108
 Jansen, Thorben, 345

Jermann, Patrick, 279
 Junayed, Muhammad, 1127
 Jyothi, Preethi, 258

Kaivapalu, Annekatrin, 953
 Kajiwara, Tomoyuki, 499
 Kamarik, Taavi, 953
 Katinskaia, Anisia, 594, 737
 Kazemi Vanhari, Fatemeh, 582
 Kert, Karina, 953
 Kerzabi, Emily, 660
 Khairallah, Christian, 1203
 Khalafallah, Ayman, 1121
 Khalil, Salah, 920
 Kim, Euigyum, 920
 Kim, Harksoo, 1145
 King, Ann, 891
 Knill, Kate, 632
 Kochmar, Ekaterina, 1, 860, 988, 1011
 Kolagar, Zahra, 415
 Kong, Cunliang, 1084
 Koutcheme, Charles, 564
 Kristensen-McLachlan, Ross, 70
 Kubis, Marek, 794
 Kucharavy, Andrei, 248
 Kucheria, Aayush, 873
 Kurfali, Murathan, 213
 Käser, Tanja, 356

Laarmann-Quante, Ronja, 237
 Lan, Andrew, 294, 765
 Laâguidi, Jammila, 237
 Leite, Bernardo, 647
 Lesterhuis, Marije, 535
 Li, Seewoo, 920
 Li, Zelong, 202
 Liin, Krista, 953
 Lim, Lanz, 1260
 Litman, Diane, 672, 752
 Liu, Bo, 1084
 Liu, Shuliang, 1084
 LIU, SILIANG, 202
 Liu, Zhenghao, 1084
 Liu, Zhengyuan, 129
 Liu, Zoey, 687
 Lopes Cardoso, Henrique, 647
 Lyu, Zhihao, 1060

M, Niranjan, 1180
 Ma, Wanjing (Any), 398
 Macina, Jakub, 1, 100

Madnani, Nitin, 850
 Maine, Silvia, 953
 Mangalam, Karttikeya, 931
 Manlises, Maria Monica, 1260
 Mao, Zhenjiang, 882
 Marciniak, Jacek, 794
 Martynova, Daria, 100
 Matsumura, Lindsay Clare, 752
 Maurya, Kaushal, 988, 1011
 Meurers, Detmar, 175, 978
 Michael, Noah-Manuel, 818
 Micluta-Campeanu, Marius, 780
 Mikeska, Jamie, 716
 Mirabella, Adriana, 708
 Miyata, Rina, 499
 Moroianu, Theodor, 24
 Mulcaire, Phoebe, 850

 N J, Karthika, 258
 Naeem, Numaan, 1254
 Nama, Rohith, 882
 Nasyrova, Regina, 517
 Nebhi, Kamel, 968
 Nguyen, Tran Minh, 202
 Nisioi, Sergiu, 1224

 Oshallah, Islam, 1194
 Östling, Robert, 213

 Pal Chowdhury, Sankalan, 1, 356, 612
 Panchal, Deval, 446
 Panesar, Amrita, 968
 Parikh, Nisarg, 294
 Park, Geon, 1145
 Park, Jungyeul, 202
 Percia David, Dimitri, 248
 Petukhova, Kseniia, 860, 1011
 Pierce, Benjamin, 752
 Pit, Henry, 1164
 Plank, Barbara, 266

 Qiu, Mengyang, 202
 Qwaider, Chatrine, 549

 Rahman, AKM Mahbubur, 1266
 Rahman, Md Ashiqur, 1127
 Rahman, Md. Abdur, 1127
 Rahman, Mohammad, 1266
 Ramakrishnan, Ganesh, 258
 rastogi, pranshu, 1098
 Rastogi, Sparsh, 1278

 Remersaro, Ignacio, 1135
 Rengarajan, Srinivasan, 1108
 Rezayi, Saed, 830, 891
 Ribeiro-Flucht, Luisa, 978
 Robaina, Santiago, 1135
 Rogobete, Roxana, 780
 Roh, Jihyeon, 1049
 Rohan, Shadman, 1266
 Rooein, Donya, 1, 356, 612
 Rosá, Aiala, 1135
 Roy, Billodal, 1180
 Roşu, Ana, 1224
 Ruchkin, Ivan, 882
 Rückert, Johannes, 334

 Sachan, Mrinmaya, 1, 100, 356, 612
 Saeed, Mariam, 1121
 Saha, Trishita, 1242
 Sakunkoo, Annabella, 697
 Sakunkoo, Jonathan, 697
 Sastre, Ignacio, 1135
 Sawhney, Nitin, 873
 Scarlatos, Alexander, 294, 765
 Schaller, Nils-Jonathan, 345
 Schellenberg, Max, 237
 Schmalz, Veronica, 937
 Sharma, Mayank, 898
 Sheu, Ching-Fan, 737
 Shi, Kevin, 931
 Shimabukuro, Mariana, 446
 Shin, Hyo Jeong, 660, 920
 Shochcho, Muhtasim, 1266
 Singh, Astha, 841
 Skidmore, Lucy, 160
 Song, Jiwoo, 1145
 Song, Wenyu, 1073
 Sonkar, Shashank, 1
 Sorokin, Alexey, 517
 Sotnikova, Anna, 279
 Srivatsa, KV Aditya, 988, 1011
 Staruch, Ryszard, 118
 Sun, Juoh, 1145
 Sung, Hakyung, 11
 Sung, Min-Chang, 11
 Sur Apan, Ishita, 1266
 Szczepański, Marcin, 794
 Szpilkowski, Adam, 794
 Säuberli, Andreas, 266

 Tack, Anaïs, 535, 937, 1011
 Tamori, Hideaki, 499

Tan, Chuangchuang, 1073
Taslimipoor, Shiva, 55
Timukova, Anna, 237
Tiwari, Rajneesh, 1098
Tjitrahardja, Eduardus, 1212
Torrance, Mark, 841
Tran, Nhat, 752

Urakawa, Toru, 499
Urrutia, Felipe, 38

Vainikko, Martin, 953
Vallez, Cyril, 248
Van Den Noortgate, Wim, 535
Vanhatalo, Ulla, 594
Vasiluta, Mihai Alexandru, 89
Vasselli, Justin, 1011
Vu, Anh-Duc, 594, 737

Wang, Deliang, 1040
Wang, Shuo, 1084
Wang, Zuowei, 398
Welch, Charles, 582
Wieczarek, Adam, 794
Woo, David, 460
Woodhead, Simon, 294
Wu, Yiheng, 594
Wulff, Stefanie, 687

Xiao, Zheng, 1078
Xie Fincham, Naiyi, 384
Xu, Jin, 1078

Yalcin, Nilay, 100
Yaneva, Victoria, 830, 891
Yang, Chao, 1040
Yang, Erhong, 1084
Yang, Haiyin, 687
Yang, Liner, 1084
Yangarber, Roman, 594, 737
Yarmohammadtoosky, Sahar, 830
Yasser, Mazen, 1121
Yin, Stella Xin, 129

Zalkow, Frank, 415
Zarcone, Alessandra, 415
Zehner, Fabian, 660, 1173
Zeng, Yawen, 1078
Zesch, Torsten, 144, 375, 660
Zhang, Jason, 898
Zhang, Terry Jingchen, 356
Zhang, Xiaoyu, 100
Zhao, Yiling, 908
Zhou, Yiyun, 830, 891
Zong, Xuquan, 1084
Zouhar, Vilém, 612