AIME-Con 2025

Artificial Intelligence in Measurement and Education Conference (AIME-Con)

Volume 3: Coordinated Session Papers

October 27-29, 2025

The AIME-Con organizers gratefully acknowledge the support from the following sponsors.

Platinum

Gold

Silver

Supporters


FOCALPOINT
Teaching & Learning. Measured. Managed


MEASUREMENT
INCORPORATED


University of
Pittsburgh
Learning Research and
Development Center


WestEd
WestEd.org

duolingo english test

The future of language assessment is here

The Duolingo English Test is a computer adaptive test powered human-in-the-loop AI and supported by rigorous validity research. The test measures speaking, writing, reading, and listening skills, providing a deeper insight into English proficiency.

Built on the latest language assessment science

- Accessible by design, supporting test takers wherever they are for just $70
- Built on rigorous research and industry- leading security
- Integrates the latest assessment science and AI for accurate results
- Accepted by over 5,800 programs worldwide

englishtest.duolingo.com

Evidence-based approach to AI in Measurement & Learning

At the intersection of artificial intelligence and educational measurement, Pearson stands as your trusted partner—delivering clarity, confidence, and innovation in every assessment moment.

Why Pearson?

- AI-Enhanced Accuracy: Using automated scoring and predictive analytics to provide insights that are accurate, fair, and timely.
- Future-Ready Solutions: Platforms that evolve with policy, pedagogy, and technology.
- Personalized Learning Journeys: Multi-lingual access and adaptive item generation to support each student's unique growth trajectory.
- Ethical AI Practices: Commitment to data security, transparency, explainability, and bias mitigation.
- Collaborative Innovation: Partnering with educators, researchers, and technologists to shape the future of assessment.

Human-Centric AI	Pearson believes AI's highest purpose is to elevate and empower human capabilities.
Assessment as a Learning Continuum	We reimagine assessments not as endpoints, but as integral parts of the learning journey.
AI as an Environment	Pearson is exploring how this shift impacts our approach to assessment—ensuring our tools are adaptive and future-ready.
Balancing Vision and Capabilities	We deliver reliable solutions today while building toward the future of AI in education.

Pearson

Order copies of this and other NCME proceedings from:

National Council on Measurement in Education (NCME)
520 S. Walnut St. Box 2388
Bloomington, IN 47402
USA
Tel: +1-812-245-8096
ncme@ncme.org

ISBN 979-8-218-84230-7

Preface

Introduction

The inaugural NCME-sponsored Artificial Intelligence in Measurement and Education Conference (AIME-Con) brought together an interdisciplinary community of experts working at the intersection of artificial intelligence (AI), educational measurement, assessment, natural language processing, learning analytics, and technological development. As AI continues to transform education and assessment practices, this conference provided a critical platform for fostering cross-disciplinary dialogue, sharing cutting-edge research, and exploring the technical, ethical, and practical implications of AI-driven innovations in measurement and education. By bringing together experts from varied domains, the conference fostered a rich exchange of knowledge to enhance the collective understanding of AI's impact on educational measurement and evaluation.

Conference Theme - Innovation and Evidence: Shaping the Future of AI in Educational Measurement

The NCME-Sponsored AIME-Con focused on how rigorous measurement standards and innovative AI applications can work together to transform education. With sessions spanning summative large-scale assessment, formative classroom assessment, automated feedback, and informal learning tools, this conference fostered both the advancement and evaluation of AI technologies that are effective, reliable, and fair.

The National Council on Measurement in Education

The National Council on Measurement in Education is a community of measurement scientists and practitioners who work together to advance theory and applications of educational measurement to benefit society. A professional organization for individuals involved in assessment, evaluation, testing, and other aspects of educational measurement, our members are involved in the construction and use of standardized tests; new forms of assessment, including performance-based assessment; program design; and program evaluation. Learn more about NCME, including our goals and our leadership, at www.ncme.org. We are grateful to the NCME.

NCME Special Interest Group on Artificial Intelligence in Measurement and Education

The AIME SIGIMIE seeks to advance the theoretical and applied research into AI of educational measurement by bringing together data scientists, psychometricians, education researchers, and other interested stakeholders. The SIGIMIE will discuss current practices in using Generative AI, approaches to evaluate their precision/accuracy, and areas where more foundational research is required into the way we test and measure educational outcomes. This group seeks to create a strong professional identity and intellectual home for those interested in the use of AI in many areas, including automated scoring, item evaluation, validity studies, formative feedback, and generative AI for automated item generation.

Proposal Requirements and Review Process for Coordinated Paper Sessions

AIME-Con invited submissions of coordinated paper sessions, which brought together 4–5 papers on a common theme within a 90-minute session. Each proposal included both session-level information (title, abstract, keywords, chair/moderator, and discussant where applicable) and paper-level details (title, short abstract, topic of interest, and either a 1,000-word structured summary or a six-page paper). **All contributors were identified at submission, as the review process was not blind.**

Submissions were evaluated by members of the review committee using a rubric that evaluated the following dimensions:

- **Relevance and community impact:** pertinence to the AI in measurement and education community, and potential contribution to current discussions and challenges in the field
- **Significance and value:** scholarly merit or practical importance of the work, and potential impact on theory, practice, or policy
- **Methodological rigor:** coherence and appropriateness of the proposed methods, techniques, and approaches; and soundness of the overall research design
- **Quality of expected outcomes:** whether the proposed analysis and interpretation methods are appropriate, and the potential contribution to knowledge in the field
- **Feasibility and timeline:** the realistic likelihood that the proposed work can be completed by the conference date

For the purposes of this conference, "AI" was defined broadly to include rule-based methods, machine learning, natural language processing, and generative AI/large language models. Reviewers provided constructive feedback and overall recommendations to ensure that accepted sessions reflected both scholarly merit and practical value to the AI in measurement and education community.

Organizing Committee

NCME Leadership

Amy Hendrickson, Ph.D. (President)
Rich Patz, Ph.D. (Executive Director)

Conference Chairs

Joshua Wilson, University of Delaware
Christopher Ormerod, Cambium Assessment
Magdalen Beiting Parrish, Federation of American Scientists

Proceedings Chair

Nitin Madnani, Duolingo

Proceedings Committee

Jill Burstein, Duolingo
Polina Harik, NBME

Program Committee

Conference Chairs

Joshua Wilson, University of Delaware
Christopher Ormerod, Cambium Assessment
Magdalen Beiting Parrish, Federation of American Scientists

Reviewers

Hope Adegoke, University of North Carolina, Greensboro
Magdalen Beiting-Parrish, Federation of American Scientists
Peter Foltz, University of Colorado, Boulder
Hudson Golino, University of Virginia
Hongli Li, Georgia State University
Sheng Li, University of Virginia
Jianyuan Ni, Juniata College
Christopher Ormerod, Cambium Assessment
Corey Palermo, Measurement Incorporated
Shaila Quazi, Drexel University
Andrew Runge, Duolingo
Christopher Runyon, National Board of Medical Examiners
Raashi Sangwan, Miller School of Medicine, University of Miami
Khem Sedhai, University of Albany
Mark Shermis, Performance Assessment Analytics, LLC
Kimberly Swygert, National Board of Medical Examiners
Joshua Wilson, University of Delaware
Jiawei Xiong, University of Georgia
Xinhui Maggie Xiong, ExamRoom AI

Table of Contents

When Does Active Learning Actually Help? Empirical Insights with Transformer-based Automated Scoring

Justin O. Barber **Michael P. Hemenway** **Edward W. Wolfe**
Pearson Education
{justin.barber, michael.hemenway, ed.wolfe}@pearson.com

Abstract

Developing automated essay scoring (AES) systems typically demands extensive human annotation, incurring significant costs and requiring considerable time. Active learning (AL) methods aim to alleviate this challenge by strategically selecting the most informative essays for scoring, thereby potentially reducing annotation requirements without compromising model accuracy. This study systematically evaluates four prominent AL strategies—uncertainty sampling, BatchBALD, BADGE, and a novel GenAI-based uncertainty approach—against a random sampling baseline, using DeBERTa-based regression models across multiple assessment prompts exhibiting varying degrees of human scorer agreement. Contrary to initial expectations, we found that AL methods provided modest but meaningful improvements only for prompts characterized by poor scorer reliability (<60% agreement per score point). Notably, extensive hyperparameter optimization alone substantially reduced the annotation budget required to achieve near-optimal scoring performance, even with random sampling. Our findings underscore that while targeted AL methods can be beneficial in contexts of low scorer reliability, rigorous hyperparameter tuning remains a foundational and highly effective strategy for minimizing annotation costs in AES system development.

1 Introduction

Automated Essay Scoring (AES) systems have become integral to educational assessments by providing efficient, reliable, and scalable evaluation of student writing. State-of-the-art AES approaches typically utilize medium- to large-size pretrained transformer-based language models such as BERT (Devlin et al., 2019), ELECTRA (Clark et al., 2020), and DeBERTa (He et al., 2021), fine-tuned on datasets of human-scored essays to produce scoring models aligned closely with human judgment. The development of robust AES models, however, usually requires extensive annotation efforts—often involving thousands of essays per prompt—posing significant practical limitations in terms of cost, time, and resources.

Active Learning (AL) mitigates these annotation burdens by scoring only the most informative essays. Although AL is well studied in NLP (Zhang et al., 2023; Li et al., 2024), few works test multiple strategies in AES or examine how scorer agreement moderates AL gains (Firoozi et al., 2023; Hellman et al., 2019). We compare four AL methods with a random sampling baseline across prompts of differing reliability.

Addressing this critical gap, our study evaluates four prominent AL strategies—uncertainty sampling, BatchBALD (Bayesian Active Learning by Disagreement), BADGE (Batch Active Learning by Diverse Gradient Embeddings), and a novel GenAI-based uncertainty sampling approach—across multiple writing and reading assessment prompts. These AL methods are benchmarked against random sampling as a baseline, examining their efficacy at annotation budgets ranging from 32 to 1,024 essays.

1.1 Research Questions

This study specifically investigates three research questions:

1. Which AL strategies yield the highest scoring agreement (measured via Quadratic Weighted Kappa [QWK]) with the minimal number of human-scored training examples, particularly across varying degrees of human inter-rater agreement?

2. Can a novel GenAI-guided AL approach effectively identify especially challenging-to-score essays, thereby enhancing the efficiency and quality of AES model training?

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 1–8

3. To what extent does comprehensive hyperparameter optimization alone (even with random sampling) significantly reduce the number of training essays required to achieve acceptable scoring accuracy across various prompts?

1.2 Contributions

Our contributions are:

- A four-way comparison of AL strategies versus a random baseline on prompts spanning different human scorer agreement;
- A novel GenAI sampler for small budgets;
- Evidence that hyperparameter tuning alone rivals AL when scorer reliability is moderate–high;
- Practical guidance for where AL is (and is not) worth the cost.

These findings hold significant implications for educational assessment organizations aiming to develop AES systems more efficiently. By understanding the nuanced contexts in which AL methods excel and the powerful impact of systematic hyperparameter tuning, stakeholders can better allocate annotation resources, enabling broader and more cost-effective application of automated scoring systems across diverse educational contexts.

2 Related Work

2.1 Active Learning in NLP

Active Learning (AL) reduces annotation costs by selecting the most informative unlabeled samples for labeling, enhancing model performance with fewer annotations (Settles, 2009; Zhang et al., 2023; Li et al., 2024). In NLP, prominent AL strategies include uncertainty-based, Bayesian, diversity-based, and hybrid approaches, which we adapt for Automated Essay Scoring (AES).

2.1.1 Uncertainty Sampling

Uncertainty-based sampling selects samples where models exhibit the highest uncertainty. Lewis and Gale (1994) introduced entropy-based selection, while Gal et al. (2017) popularized Monte Carlo dropout to estimate uncertainty in deep learning models. Margin-based methods, recently highlighted by Doucet et al. (2024), select samples with minimal differences between top class probabilities and have frequently outperformed random sampling in NLP tasks.

2.1.2 Bayesian Active Learning

Bayesian Active Learning (BAL) focuses explicitly on maximizing information gain regarding model parameters (Siddhant and Lipton, 2018). Bayesian Active Learning by Disagreement (BALD) selects samples based on uncertainty across posterior predictions (Houlsby et al., 2011). BatchBALD (Kirsch et al., 2019) extends this to batch selection, reducing redundancy by jointly evaluating batch informativeness at increased computational cost.

2.1.3 Diversity-Based and Hybrid Sampling

Diversity-based methods select samples that represent diverse regions of input space, ensuring robust generalization. Hybrid strategies like BADGE (Ash et al., 2020) combine uncertainty and diversity by clustering gradient embeddings to identify diverse yet informative samples, demonstrating strong performance in various classification tasks.

2.1.4 LLM-Guided Active Learning

Emerging approaches integrate Large Language Models (LLMs) into AL for nuanced semantic evaluation of samples. Methods such as ActiveLLM (Bayer and Reuter, 2024), ActivePrune (Azeemi et al., 2024), SelectLLM (Parkar et al., 2024), and ranking-based approaches (Jeong et al., 2025) have shown promise in identifying linguistically complex or ambiguous samples relevant for AES.

2.2 Active Learning for Automated Scoring

Research explicitly addressing AL in automated scoring contexts remains sparse. Horbach and Palmer (2016) compared AL strategies on short-answer scoring, noting significant variability across prompts. Hellman et al. (2019) demonstrated batch-mode AL effectiveness in instructor-driven contexts. Firoozi et al. (2023) highlighted uncertainty sampling's efficiency in AES, although their work focused exclusively on shallow models without exploring transformer-based methods or comprehensive comparisons.

2.2.1 Our Study in Context

Existing AES-focused AL studies have not systematically evaluated how scorer reliability impacts AL strategy efficacy nor have they fully explored the independent impact of extensive hyperparameter optimization. Our study addresses these gaps by rigorously comparing multiple AL strategies, explicitly considering varying scorer reliability levels, and demonstrating the substantial efficiency

gains achievable through hyperparameter optimization alone. These insights inform best practices for practical AES deployment.

3 Methods

3.1 Problem Formulation

Given an unlabeled pool $\mathcal{U}$ and budget B, we run four AL rounds. Each round (i) trains on current labels, (ii) selects $\lfloor B/4 \rfloor$ essays via an acquisition function, (iii) obtains scores, and (iv) updates the model. A final training pass with tuned hyperparameters follows. Performance is reported on a held-out validation set.

3.2 Model Architecture

We fine-tune DeBERTaV3-base (He et al., 2021) as a regression model by adding a single linear head on the [CLS] embedding and optimizing Mean-Squared-Error loss weighted by inverse score frequency. Essays are tokenized with the DeBERTaV2 tokenizer (512-token limit) and trained using AdamW with linear warm-up and gradient clipping.

3.3 Active Learning Strategies

We evaluate four AL strategies:

Uncertainty Sampling: Selects essays with highest predictive entropy based on Gaussian-derived probability distributions from regression outputs.

BatchBALD (Kirsch et al., 2019): Maximizes batch mutual information using Monte Carlo dropout, first filtering the unlabeled pool by predictive entropy to enhance computational efficiency.

BADGE (Ash et al., 2020): Combines uncertainty and diversity by clustering gradient embeddings derived from a temporary classification head on the model encoder.

GenAI-Uncertainty Sampling (novel approach): Uses large language models (LLMs) to identify challenging essays (rated 1–5 on scoring difficulty). Essays rated highly challenging (5) are prioritized, selecting diverse examples within difficulty strata using k-means clustering.

3.4 Multi-Round Active Learning Framework

Our AL approach includes:

- Initial seed of 16 essays.
- Four AL rounds (one for GenAI), evenly dividing annotation budgets.
- Each round selects essays for scoring, expands the labeled set, and retrains the model.

3.5 Hyperparameter Optimization

Given its significant impact, we rigorously optimize hyperparameters using Optuna (Akiba et al., 2019):

Search Space:

- Learning rate: [1e-5 to 2e-5]
- Weight decay: [1e-3 to 1e-1]
- Batch size: [4, 8]

Optimization Approach:

1. **Discovery Phase**: Perform 40-trial hyperparameter optimization using random sampling at each annotation budget.
2. **Evaluation Phase**: Evaluate the top 16 discovered hyperparameter configurations across all AL strategies, limiting computationally intensive strategies (BatchBALD, BADGE) to budgets <= 384.

Final models train for up to 30 epochs with early stopping (patience=5) based on validation loss.

4 Experiments

4.1 Data Sources

Operational corpus. Our experiments utilize operational student response data from a large-scale summative K–12 assessment administered across multiple U.S. states. The dataset comprises both short constructed-response reading items and full-length essay prompts, capturing diverse aspects of student writing performance.

Prompt Selection Criteria. To establish a balanced and robust evaluation framework, prompts were selected based on sufficient availability of double-scored responses. This resulted in a set of eight suitable prompts: five reading items and three writing prompts.

Reading Tasks. The reading task subset consists of three Grade-8 items (R-8A, R-8B, R-8C) and two Grade-10 items (R-10A, R-10B). Reading responses were holistically scored on a three-point ordinal scale (0–2) or a five-point ordinal scale (0-4), each assessing a single construct.

Task	Grade	Genre / Trait	Scale	N
R-8A	8	Reading	0–2	5,305
R-8B	8	Reading	0–4	4,575
R-8C	8	Reading	0–2	3,911
R-10A	10	Reading	0–2	5,931
R-10B	10	Reading	0–4	4,987
W-5	5	Argumentative, Content	0–3	11,088
W-8	8	Informative, Content	0–3	10,754
W-11	11	Narrative, Content	0–3	10,416

Table 1: Descriptive statistics for the experimental corpus.

Writing Tasks. The writing tasks cover Grades 5 through 11, balanced across genres: W-5 (argumentative), W-8 (informative/explanatory), and W-11 (narrative). Although each essay includes multiple trait scores, we focus specifically on the *Content* trait, given its strong alignment with textual evidence and minimal confounding by surface-level mechanical features. Content scores range from 0 to 3.

Sample Sizes. Usable responses per task range from 3,911 to 11,088. Table 1 summarizes detailed counts.

Train–Validation Protocol. For each prompt–trait pair, we hold out a stratified sample of 500 responses as a validation set, preserving the marginal score distribution. This validation set is used exclusively for model checkpoint selection and hyperparameter optimization. Consequently, although it remains distinct from the training data used directly for gradient updates, it is not strictly unseen. This methodological choice may slightly overestimate absolute model performance but does not affect our comparative analysis of active learning strategies.

4.2 Evaluation Metrics

Quadratic Weighted Kappa (QWK): Our primary evaluation criterion measures the degree of agreement between model predictions and human raters and accounts explicitly for varying degrees of scoring discrepancy. We calculate QWK using Cohen's quadratic weighted kappa implementation from scikit-learn.

Metrics are calculated after rounding and clipping predictions: $\hat{y} = \text{clip}(\text{round}(f_\theta(x)), y_{\min}, y_{\max})$, where $y_{\min}$ and $y_{\max}$ represent score boundaries.

Model selection during training employs early stopping (patience=5) based on validation loss, with the best-performing model checkpoint saved according to QWK scores. Hyperparameter optimization also prioritizes QWK.

4.3 Implementation Details

Models are trained in PyTorch with Hugging Face Transformers on NVIDIA A10 GPUs. We use AdamW with 10% warm-up, gradient clipping (1.0), mixed precision, and smoothed inverse-frequency class weights (70% empirical frequency + 30% uniform distribution). Hyperparameter searches run in parallel round-robin across GPUs. For efficiency we drop BatchBALD and BADGE when budgets exceed 384 essays and subsample 500–2,560 essays for GenAI.

Strategy-specific details:

- **BatchBALD**: 10 Monte Carlo dropout passes with initial entropy-based filtering (top 10%, minimum 2,000 essays).
- **BADGE**: Temporary classification head derived from the regression model to compute gradient embeddings.

All experiments utilize fixed random seeds for reproducibility across NumPy, PyTorch, and strategy-specific operations.

Sample Size	Random	Uncertainty	BatchBALD	GenAI	BADGE
32	**0.79**	**0.79**	**0.79**	0.75	0.78
64	**0.81**	0.80	0.78	0.77	0.77
96	**0.81**	**0.81**	0.78	**0.81**	0.80
128	0.80	**0.81**	0.79	0.80	**0.81**
192	**0.82**	0.80	0.80	0.78	**0.82**
256	**0.83**	0.82	0.80	0.82	0.81
384	**0.83**	0.82	0.81	0.80	0.82
1024	**0.84**	–	–	–	–

Table 2: QWK results for prompts with good scorer agreement. Bold indicates the highest score(s) per row.

Sample Size	Random	Uncertainty	BatchBALD	GenAI	BADGE
32	**0.77**	0.70	0.74	0.75	0.73
64	**0.79**	0.73	0.77	0.71	0.77
96	**0.79**	0.76	0.75	0.76	0.75
128	**0.80**	0.76	0.78	0.70	0.78
192	**0.81**	0.75	0.77	0.78	0.76
256	**0.82**	0.78	0.79	0.79	0.78
384	**0.81**	0.76	0.80	0.79	0.80
1024	**0.82**	–	–	–	–

Table 3: QWK results for prompts with acceptable scorer agreement. Bold indicates the highest score(s) per row.

Sample Size	Random	Uncertainty	BatchBALD	GenAI	BADGE
32	0.69	0.67	0.66	**0.70**	0.65
64	0.67	0.71	0.69	**0.73**	0.69
96	0.71	0.66	0.70	**0.72**	0.71
128	0.71	0.70	0.71	0.72	**0.73**
192	**0.76**	0.71	0.75	0.73	0.73
256	0.73	0.71	**0.74**	0.73	**0.74**
384	**0.75**	0.74	**0.75**	0.74	**0.75**
1024	**0.77**	–	–	–	–

Table 4: QWK results for prompts with poor scorer agreement. Bold indicates the highest score(s) per row. Dashes indicate unavailable or omitted results.

5 Results

5.1 Performance of Active Learning Strategies by Scoring Quality

Tables 2, 3, and 4 present the Quadratic Weighted Kappa (QWK) performance of Active Learning (AL) strategies across three contexts of scorer reliability: good, acceptable, and poor. These tables explicitly compare random sampling against four AL methods (Uncertainty, BatchBALD, GenAI, and BADGE).

Table 2 highlights the scenario of good scorer agreement (approximately 80% agreement). Here, AL methods exhibit little advantage over random sampling. Even at small annotation budgets (e.g., $n = 32$ or 64), random sampling matches or surpasses AL approaches. For example, at $n = 256$, random sampling (QWK=0.83), uncertainty sampling (0.82), and GenAI (0.82) demonstrate similar effectiveness, but no AL method exceeds random sampling substantially.

Table 3 shows analogous results for acceptable scorer agreement contexts (about 60% agreement). Again, random sampling typically achieves a slightly higher or equal QWK compared to AL strategies across most sample sizes, though the GenAI method achieves competitive performance at several points. Notably, at $n = 256$ annotations, random sampling still yields the top performance (QWK=0.82), followed closely by BatchBALD, GenAI, and BADGE strategies, each achieving scores of at least 0.78.

In contrast, for prompts with poor scorer agreement (<60%), AL methods show clearer advantages over random sampling (Table 4). Particularly at lower annotation budgets, uncertainty-based strategies, including the GenAI and BADGE methods, consistently outperform random selection. For instance, at $n = 64$, the GenAI method (0.73) significantly surpasses random sampling (0.67).

Training Sample Size	QWK (Random Sampling)
32	0.77
64	0.78
96	0.78
128	0.80
192	0.81
256	0.82
384	0.82
1024	0.82

Table 5: Impact of 40-trial hyperparameter optimization on QWK using random sampling across sample sizes for all prompts.

Similarly, uncertainty-based AL strategies continue to show small but consistent advantages at larger annotation sizes (e.g., $n = 128$ through $n = 384$), reflecting their capacity to effectively select informative and potentially challenging essays for model training.

Finally, in table 5 we explicitly examine how extensive hyperparameter optimization alone influences AES performance (Table 5). With careful tuning, random sampling swiftly achieves strong performance and approaches saturation quickly (QWK=0.81 at $n = 192$ annotations), demonstrating the significant impact of optimization without specialized AL. Indeed, this tuning reduces required annotation counts substantially, effectively narrowing the advantage that sophisticated AL methods could achieve in many practical scoring scenarios.

6 Discussion

6.1 Effectiveness of Active Learning for AES

Our findings indicate that active learning (AL) methods provide modest yet meaningful benefits specifically for prompts characterized by low scorer agreement (<60% agreement per score point). In these challenging scoring contexts, uncertainty-based methods, including BatchBALD and our novel GenAI-based approach, consistently yielded slight improvements over random sampling at smaller annotation budgets. This aligns with the intuition that uncertain, borderline scoring cases are particularly informative for model calibration, and extends prior findings by Firoozi et al. (2023), emphasizing AL's specific utility in challenging scoring scenarios.

However, contrary to initial expectations, AL methods provided no substantial advantage over random sampling in contexts with moderate to high scorer reliability (approximately 60–80% agreement). This lack of improvement can largely be attributed to our extensive hyperparameter optimization process, which significantly boosted the performance of random sampling, leaving limited room for AL methods to offer additional benefits.

Additionally, our GenAI-based approach demonstrated encouraging results in identifying challenging essays early in the annotation process, highlighting the potential of leveraging large language models to enhance targeted sampling. Although the overall improvement was modest, the interpretability and targeted nature of the GenAI sampling suggest potential future avenues for improving essay scoring models, especially in highly ambiguous scoring contexts.

6.2 Impact of Hyperparameter Optimization

A critical secondary finding of our study is the pronounced effectiveness of extensive hyperparameter optimization—even when employing random sampling. Our rigorous hyperparameter tuning approach (40 trials using Optuna) substantially reduced the annotation budget required to achieve robust model performance. This suggests that, in many practical AES contexts, careful model optimization can significantly improve annotation efficiency, often exceeding the marginal gains offered by more complex sampling strategies.

6.3 Practical Implications

The findings reported here offer important insights for the practical development and operational management of AES systems:

- **When to use active learning (AL):** Our findings suggest that AL methods demonstrate the strongest benefits in low-reliability scoring contexts. When scoring reliability is low and essays are challenging to rate, AL techniques—such as uncertainty-based sampling and GenAI methods—systematically identify the most informative instances, thus effectively improving model quality and calibration.

- **Tune first, apply AL second:** Extensive hyperparameter optimization alone produces highly competitive AES models, especially for scoring contexts with scorer reliability at or above 60%. Model builders should, therefore, devote significant attention initially to optimizing hyperparameters before turning to AL methods.

We thus propose the following operational framework for AES implementations based on these insights:

1. Begin with a modest-sized randomly sampled initial set (e.g., 16–32 essays), ensuring sufficient prompt coverage.
2. Immediately prioritize extensive hyperparameter optimization early in the model-development process.
3. After initial tuning, selectively apply uncertainty-based AL (particularly GenAI-driven sampling) as annotation proceeds, especially in cases of lower scoring reliability.
4. As more responses are collected, continuously revisit and adjust hyperparameters, since optimal settings may evolve with increasing data.

6.4 Limitations and Future Work

Our study offers valuable insights but has several limitations indicating promising directions for future research:

- **Prompt and Context Diversity**: Our analysis was limited to eight prompts from a single assessment context. Future work should explore broader prompt variability, scoring traits, and educational contexts.
- **Human-in-the-loop Validation**: Real-world AL implementations involve iterative human scoring. Future research should directly assess AL's practical implications within live annotation workflows.
- **Hyperparameter Exploration**: This work has highlighted the importance of hyperparameter optimization in model performance. Future experiments will consider an even wider hyperparameter space and optimization techniques that would be robust in operational contexts.
- **Fairness Considerations**: Further research could investigate how AL and targeted sampling methods, including GenAI, influence scoring fairness and demographic representation, potentially integrating fairness-aware constraints or regularizations.
- **Semi-Supervised Approaches**: Leveraging unlabeled data via semi-supervised or self-supervised learning methods (e.g., consistency regularization, pseudo-labeling, contrastive learning) may further enhance AES efficiency and warrants exploration.

Overall, our results highlight both the nuanced effectiveness of active learning methods under specific conditions and the crucial foundational role of rigorous hyperparameter optimization. These insights provide clear guidance for enhancing annotation efficiency and scoring reliability within AES deployments.

7 Conclusion

This study highlights two key findings for automated essay scoring (AES): First, active learning (AL) offers modest improvements over random sampling primarily in low-reliability scoring contexts. In prompts with higher scorer agreement, random sampling—when paired with wide hyperparameter sweeps—achieves near-optimal performance, often matching or exceeding AL strategies. Second, our novel GenAI-based sampling approach shows promise in identifying challenging essays early, but its benefits diminish as budgets increase.

These results suggest that rigorous hyperparameter optimization may be more impactful than AL in many AES scenarios. For practical deployment, AL may still provide value in identifying difficult examples and supporting scorer calibration in ambiguous contexts. Future research should explore how AL interacts with fairness, human-in-the-loop scoring, and hybrid semi-supervised learning strategies to further improve scoring efficiency and transparency.

Acknowledgments

We thank our colleagues at Pearson for their support and feedback on this research. We also acknowledge the computational resources provided by Pearson's technology team.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. *arXiv preprint*. ArXiv:1907.10902 [cs].

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. *arXiv preprint*. ArXiv:1906.03671 [cs].

Abdul Hameed Azeemi, Ihsan Ayyub Qazi, and Agha Ali Raza. 2024. Language Model-Driven Data Pruning Enables Efficient Active Learning. *arXiv preprint*. ArXiv:2410.04275 [cs].

Markus Bayer and Christian Reuter. 2024. ActiveLLM: Large Language Model-based Active Learning for Textual Few-Shot Scenarios. *arXiv preprint*. ArXiv:2405.10808 [cs] version: 1.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

Paul Doucet, Benjamin Estermann, Till Aczel, and Roger Wattenhofer. 2024. Bridging Diversity and Uncertainty in Active learning with Self-Supervised Pre-Training. *arXiv preprint*. ArXiv:2403.03728 [cs] version: 1.

Tahereh Firoozi, Hamid Mohammadi, and Mark J. Gierl. 2023. Using Active Learning Methods to Strategically Select Essays for Automated Scoring. *Educational Measurement*, 42(1):34–43. ArXiv:2301.00628 [cs].

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian Active Learning with Image Data. *arXiv preprint*. ArXiv:1703.02910 [cs].

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*.

Scott Hellman, Mark Rosenstein, Andrew Gorman, William Murray, Lee Becker, Alok Baikadi, Jill Budden, and Peter W. Foltz. 2019. Scaling Up Writing in the Curriculum: Batch Mode Active Learning for Automated Essay Scoring. In *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale*, pages 1–10, Chicago IL USA. ACM.

Andrea Horbach and Alexis Palmer. 2016. Investigating Active Learning for Short-Answer Scoring. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 301–311, San Diego, CA. Association for Computational Linguistics.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian Active Learning for Classification and Preference Learning. *arXiv preprint*. ArXiv:1112.5745 [stat].

Daniel P. Jeong, Zachary C. Lipton, and Pradeep Ravikumar. 2025. LLM-Select: Feature Selection with Large Language Models. *arXiv preprint*. ArXiv:2407.02694 [cs].

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. 2019. BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. *arXiv preprint*. ArXiv:1906.08158 [cs].

David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. *arXiv preprint*. ArXiv:cmp-lg/9407020.

Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. 2024. A Survey on Deep Active Learning: Recent Advances and New Frontiers. *arXiv preprint*. ArXiv:2405.00334 [cs].

Ritik Sachin Parkar, Jaehyung Kim, Jong Inn Park, and Dongyeop Kang. 2024. SelectLLM: Can LLMs Select Important Instructions to Annotate? *arXiv preprint*. ArXiv:2401.16553 [cs].

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study. *arXiv preprint*. ArXiv:1808.05697 [cs].

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2023. A Survey of Active Learning for Natural Language Processing. *arXiv preprint*. ArXiv:2210.10109 [cs].

Automated Essay Scoring Incorporating Annotations from Automated Feedback Systems

Christopher Ormerod
Cambium Assessment Inc.
christopher.ormerod@cambiumassessment.com

Abstract

This study illustrates how incorporating feedback-oriented annotations into the scoring pipeline can enhance the accuracy of automated essay scoring (AES). This approach is demonstrated with the Persuasive Essays for Rating, Selecting, and Understanding Argumentative and Discourse Elements (PERSUADE) corpus. We integrate two types of feedback-driven annotations: those that identify spelling and grammatical errors, and those that highlight argumentative components. To illustrate how this method could be applied in real-world scenarios, we employ two LLMs to generate annotations – a generative language model used for spell correction and an encoder-based token-classifier trained to identify and mark argumentative elements. By incorporating annotations into the scoring process, we demonstrate improvements in performance using encoder-based large language models fine-tuned as classifiers.

1 Introduction

Automated Essay Scoring (AES) uses statistical models to assign grades to essays that approximate hand-scoring (Shermis and Hamner, 2013). Automated Writing Evaluation (AWE) is the provision of automated feedback designed to help students iteratively improve their essays (Huawei and Aryadoust, 2023). Initial attempts at AES and AWE were based on Bag-of-Words (BoW) models that combine frequency-based and hand-crafted features (Attali and Burstein, 2006; Page, 2003). Well-designed features can serve two purposes: to improve scoring accuracy and provide feedback to students to improve their essays. These features tend to be global features, such as the number of words, sentence length, or readability metrics, and are not based on fine-grained semantics or the organizational structure of essays.

Many modern AES engines employ transformer-based Large Language Models (LLM)s (Rodriguez et al., 2019), which offer improved accuracy over bag-of-words models. However, this comes at the expense of reduced interpretability due to implicit feature definition. Some researchers have sought to combine LLM-derived features with traditional hand-crafted features to enhance accuracy and provide some level of interpretability (Uto and Uchida, 2020). LLMs also offer the ability to provide semantically rich feedback, such as key phrases from explainable AI (Boulanger and Kumar, 2020), annotation schemas for automated writing evaluation systems (Crossley et al., 2022; Lottridge et al., 2024), and the generation of detailed feedback through LLM prompting techniques (Lee et al., 2024). This study considers how these semantically rich features, designed primarily for providing feedback, can also enhance scoring accuracy.

We demonstrate our approach using the Persuasive Essays for Rating, Selecting, and Understanding Argumentative and Discourse Elements (PERSUADE) corpus, which is a dataset of essays in which the argumentative components of the essays were annotated (Crossley et al., 2022). This dataset also contains scores assigned against an openly available holistic rubric [1] and demographic data, which allows us to test any AES system with respect to operational standards, including the addition of potential bias (Williamson et al., 2012).

The two classes of features we consider are derived from Grammatical Error Correction (GEC) and Computational Argumentation. The goal of GEC is to provide a mapping from a sentence that may or may not contain errors in language, to a version with the same meaning with fewer language errors (Martynov et al., 2023), while computational argumentation seeks to isolate and analyze the set of argumentative components of an essay (Stab and Gurevych, 2014a). For these features to be incorporated into an AES pipeline, we leverage

[1] https://github.com/scrosseye/persuade_corpus_2.0

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 9–18

the ability of language models to accurately annotate argumentative components (Ormerod et al., 2023) and correct spelling and grammatical errors (Rothe et al., 2021). These spelling and grammatical corrections can then be annotated and classified to provide locally defined information on conventions (Korre and Pavlopoulos, 2020). Once these annotations have been derived, we incorporate the annotations using Extensible Markup Language (XML) for easy parsing, which facilitates natural integration into an AWE system based on essays encoded in HTML.

We must also be cognizant that increased automation can exacerbate biases, especially for English Language Learners (Ormerod, 2022a). For this reason, we also examine whether this pipeline leads to greater bias by examining the standardized mean difference for the relevant subgroups.

In §2, we describe our methods, including the data used, the models, and the approach. The performance of the annotation models and the scoring models are presented in §3. We will discuss the findings and suggest future directions in §4.

2 Method

2.1 Data

The PERSUADE corpus is an openly available dataset of 25,996 argumentative essays between grades 6 and 12 on a range of 15 different topics (Crossley et al., 2022). The essays are responses to prompts that are either dependent on source material, or independent of source material. The set has been divided into a training set and a test set by the original authors. An outline of the composition of these two sets is presented in Table 1.

2.1.1 Annotations

A key characteristic of the corpus that makes it useful from the standpoint of computational argumentation is the annotations. The argumentative clauses of each essay were identified and classified into one of seven classes:

1. **Lead (L):** An introduction that begins with a statistic, a quotation, a description, or some other device to grab the reader's attention and point toward the thesis.
2. **Position (P):** An opinion or conclusion on the main question
3. **Claim (C1)**: A claim that supports the position
4. **Counterclaim (C2)**: A claim that refutes another claim or gives an opposing reason to the position
5. **Rebuttal (R)**: A claim that refutes a counterclaim
6. **Evidence (E)**: Ideas or examples that support claims, counterclaims, rebuttals, or the position
7. **Concluding Statement (C3)**: A concluding statement that restates the position and claims

Grade	Train	Test	Total	Avg. Len.
6	688	684	1372	294.6
8	5614	4015	9629	374.9
9	1831	235	2066	426.6
10	4654	3620	8274	407.6
11	1863	1220	3083	610.9
12	243	161	404	469.0
Unk.	701	467	1168	452.5
Total	15594	10402	25996	418.1

Table 1: The grade level and length statistics for the training and testing splits for the PERSUADE corpus, and the counts of essays that are responses to prompts that are dependent (Dep.) on source material and independent (Ind..) of source material.

There are an average of 11.0 annotated components in each essay. Some descriptive statistics on the distribution of applied labels can be found in Table 2. Among the labels, the most frequently applied labels are Claims and Evidence and the least frequently applied labels are Counterclaims and Rebuttals. In accordance with state standards, the development of a counter-argument in persuasive essay writing is developed at grades eight and beyond, hence, Counterclaims and Rebuttals are rarely applied at the sixth-grade level.

	6	8	9	10	11	12
L	4.3	4.5	4.7	5.9	6.0	5.0
P	9.8	9.0	9.1	9.9	7.0	8.3
C1	27.2	27.7	27.6	29.8	31.2	34.3
C2	2.1	2.8	5.2	2.5	5.3	5.0
R	1.5	2.2	3.7	1.7	4.3	2.0
E	27.3	25.7	26.8	28.7	23.4	27.5
C3	8.0	7.5	7.6	8.7	6.9	7.5

Table 2: Some descriptive statistics regarding the distribution of annotations with respect to the various grade levels.

We did not use the effectiveness scores for the discourse elements in this study. This was a conscious choice due to the fairly low agreement between the effectiveness scores assigned by human raters ($\kappa = 0.316$). Similar attempts at judging the quality of arguments have also yielded low agreement rates (Gretz et al., 2019; Toledo et al., 2019).

2.1.2 Scores

Each essay was graded against a standardized SAT holistic essay scoring rubric, which was slightly modified for the source-based essays [2]. Based on the rubric, a high-scoring essay (5-6) demonstrates mastery through effective development of a clear point of view, strong critical thinking with appropriate supporting evidence, well-organized structure with coherent progression of ideas, skillful language use with varied vocabulary and sentence structure, and minimal grammatical errors. In contrast, a lower-scoring essay (1-3) exhibits significant weaknesses: vague or limited viewpoint, weak critical thinking with insufficient evidence, poor organization resulting in disjointed presentation, limited vocabulary with incorrect word choices, frequent sentence structure problems, and numerous grammatical errors that interfere with meaning.

The score distribution is fairly regular, with the highest and lowest scores being the rarest. The full score distribution can be found in Table 3. The reported inter-rated reliability, as reported in (Crossley et al., 2022), is given by $\kappa = 0.745$ (see (6)).

[2] https://www.kaggle.com/datasets/davidspencer/persuade-rubric-holistic-essay-scoring

Score	1	2	3	4	5	6
%	4.0	21.9	32.2	25.9	12.7	3.4

Table 3: The score distribution for the PERSUADE dataset.

The key differentiators in the rubric between high and low scoring essays are the clarity of thought, quality of supporting evidence, organizational coherence, and technical proficiency in language use. The premise behind the approach is that organizational coherence and technical proficiency in language are both made clearer by highlighting the argumentative components and conventions-based errors. Provided our pipeline for annotating essays is sufficiently accurate, these annotations should help the engine align scores with the rubric.

2.1.3 Augmented Data

The input into the scoring model was augmented to use the annotation information using Extensible Markup Language (XML). We have an XML tag per argumentative component type. To annotate conventions errors, we used the ERRANT tool (Bryant et al., 2017). The ERRANT tool classifies errors into 25 different main types, with many of these categories appearing with three different subtypes: "R" for replace, "M" for missing, and "U" for Unnecessary. For example, one category, "PUNCT", refers to a punctuation error. We can either replace, remove, or add punctuation to a sentence to make it correct, corresponding to "R:PUNCT", "U:PUNCT", and "M:PUNCT", respectively. We refer to (Bryant et al., 2017) for a full explanation of the categories.

To simplify the categories for annotation purposes, we divide all possible ERRANT annotations into three labels: `<Spelling >`, `<PunctOrth >`, and `<Grammar >`. The `<Spelling >` label is applied to the subcategories of "SPELL", the `<PunctOrth >` is applied to the subcategories of "PUNCT" and "ORTH", while all other categories are designated as having labels of `<Grammar >`. This means that we have a total of 10 annotation labels: 7 associated with argumentative components and 3 for convention errors. An example of the input into the model is shown in Figure 1. Since we do not have human-annotated data, the augmented data relies on the output from an annotation model and a spell-correction model, both of which can contribute to annotations that are less accurate.

```
<Lead>There can be many <Spelling>
advanteges</Spelling> and <Spelling>
disadvanteges</Spelling> to having a
car but the <Spelling>advanteges
<Spelling> to not having a <Grammar>card
</Grammar> greatly outweighs having one.
</Lead>There can be many reasons why not
having a car is great but the main three
are <Claim>it reduces pollution reduces
stress</Claim> and <Claim>having less
cars reduces the <Spelling>noice
<Spelling> pollution in a city. </Claim>
Can you imagine a place with no cars?
```

Figure 1: An example of the model input using an excerpt of an annotated essay.

Demographics: The last detail of this dataset, which makes it exceptionally well-suited to the investigation of AES in an operational setting, is the accompanying demographic information. This allows us to investigate any additional potential bias introduced in modeling. We measure bias by the original operational standards defined by Williamson et al. (Williamson et al., 2012). While this standard is important for many reasons, there have been numerous alternative approaches to bias (Ormerod et al., 2022).

key	Subgroup	Train	Test
WC	White/Caucasian	7012	4559
HL	Hispanic/Latino	3869	2691
BA	Black/African American	2975	1984
AP	Asian/Pacific Islander	1072	671
Mix	Two or more	598	424
Nat	American Indian Alaskan Native	68	73
ELL	English Language Learner	1330	914
DE	Disadvantaged Economically	5391	4252
ID	Identified Disability	1516	1172

Table 4: The main subgroup populations in the train and test set.

The population of various subgroups in the train and test split, as presented in the data, have been outlined in Table 4.

2.2 Modeling details

Since the advantages of the transformer were first celebrated (Vaswani et al., 2017) and BERT was trained (Devlin et al., 2018), many of the state-of-the-art results can be attributed to transformer-based LLMs (Wang et al., 2019). For this reason, we restrict our attention to fine-tuned transformer-based LLMs, whose architectures can be described as encoder, decoder, or encoder-decoder models (Vaswani et al., 2017). As a general rule, encoder models excel in natural language inference tasks (Devlin et al., 2018), decoder models excel in generative tasks (Radford et al., 2018), and encoder-decoder models excel in translation, where the task benefits from representation learning (Raffel et al., 2020).

2.2.1 Annotation model

The task of annotations can be framed as a token-classification task, where each token is classified into one of eight possible labels, one for each possible argumentative component in addition to one extra label for unannotated regions. This task lends itself to an encoder-based model trained as a masked language model like BERT (Devlin et al., 2018). Since BERT, arguably the best performing series of models are Microsofts' DeBERTa model series (He et al., 2021). The problem with these models is that many of the essays exceed the 512 token limit after tokenization. For this reason, we turn to a newly developed long context model known as ModernBERT (Warner et al., 2024).

Aside from some differences in the choices of normalization layers (Xiong et al., 2020), the use of gated activation functions (Shazeer, 2020), and more extensive pretaining, the biggest difference in the architecture is the use of Rotational Positional Embeddings (RoPE) (Su et al., 2024). To understand how RoPE works, in the original implementation of attention, the output of attention is given as a function of the key vectors, k_i, query vectors, q_j, and value vectors, v_l, given by

$$a_{m,n} = \frac{\exp(q_m^T k_n/\sqrt{d})}{\sum_j \exp(q_m^T k_j/\sqrt{d})}. \quad (1)$$

where key, query, and value vectors are functions of the embedding vectors at the first attention layer. The standard construction is that these functions be affine linear functions (linear with a bias term), where the positional embedding is the addition of token embeddings and some learnable positional

embedding terms. Instead of adding a vector, we define the query and key vectors using

$$q_m = R^d_{\Theta,m} W_q x_m, \qquad k_m = R^d_{\Theta,m} W_k x_m \tag{2}$$

where $R^d_{\Theta,m}$ a block-diagonal matrix of 2-dimensional rotation matrices, r_ϕ, given by

$$R^d_{\Theta,m} = \text{diag}(r_{m\theta_1}, \ldots, r_{m\theta_{d/2}}), \tag{3}$$

$$r_\phi = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}, \tag{4}$$

with $\theta_i = \tilde{\vartheta}^{-2(i-1)/d}$. The term ϑ is called the RoPE Theta. The key idea is that this transformation encodes positional information by rotating token embeddings in a particular manner that allows the model to understand relative distances between tokens rather than just absolute positions.

The ModernBERT model, like BERT, is trained as a masked-language model with an encoder and a token classification head. The structure of the encoder includes multiple layers of self-attention in which the attention layers alternate between rotary embeddings with base $\vartheta = 1 \times 10^4$ and $\vartheta = 1.6 \times 10^5$. However, the training was done in two phases. By adjusting the value of ϑ, researchers developed a way to scale the context length of the RoPE embedding (Fu et al., 2024), which has been applied to many other models (AI@Meta, 2024). Using this technique, the ModernBERT model was pretrained on 1.7 trillion tokens with a context length of 1024 with a $\vartheta = 10^{-4}$, which was extended to 8196 by additional training with altered layers in which $\vartheta = 1.6 \times 10^5$. In this manner, one way of thinking of the change in ϑ values in the encoder is that the attention mechanism alternates between global and local attention.

It should be noted that many architectures have attempted to circumvent the context limitation, such as the Reformer (Kitaev et al., 2020), Longformer (Beltagy et al., 2020), Transformer-XL (Dai et al., 2019), and XLNet (Yang et al., 2019), to name a few. Many of these solutions use some sort of sliding context window and/or a recurrent adaptation of the transformer architecture. Extending the context using rotary positional embeddings is more computationally efficient and effective at scaling to large context lengths, making ModernBERT a more appropriate choice in this context.

The pretrained ModernBERT model was modified for annotation by adding a classification head to the encoder. The annotator model possesses a classification head with 9 output dimensions: 7 for argumentative component labels, 1 for unannotated text, and 1 for padded variables (which can be disregarded). The output represents log probabilities for each label. This model was trained to predict the annotations for each token in the training set using the cross-entropy loss function and the Adam optimizer with a learning rate of 1e-6 over 10 epochs. We simplified the training by not having a development set.

2.2.2 Spelling and Grammar

The most successful and accurate way to perform GEC has been to utilize representation learning, hence, we seek an encoder-decoder model, which is a sequence-to-sequence model (Sutskever et al., 2014). The premise behind this method is that we are able to use the encoder to map sentences to a vector space that encodes the semantic information, while a decoder maps from the vector space to grammatically correct text (Rothe et al., 2021). This suggests we use a T5 model, pretrained as a text-to-text transformer and fine-tuned to perform grammatical error correction (GEC) (Martynov et al., 2023). Once the correction is defined, the original sentence and the correction are used as input into the ERRANT tool to produce a classified correction (Korre and Pavlopoulos, 2020).

2.2.3 Scoring Models

Given the model input includes both the essay and any annotations, we require long-context models. While we have experimented with the use of QLoRA-trained generative Models (Ormerod and Kwako, 2024), to simplify the presentation, we use the ModernBERT model for scoring in addition to annotation. In this case, the scoring models were constructed by appending a linear classification head to the ModernBERT model with 6 targets, one for each score point.

2.3 Evaluation

We have two models to evaluate: an annotation model and a classification model. There are many challenges to assessing annotations for argumentative clauses. We need to carefully define what it means for a particular clause to be identified and correctly classified, given that certain identifications may not perfectly align with the predicted components. For holistic scoring, there are many more well-defined and accepted standards presented by Williamson et al. (Williamson et al.,

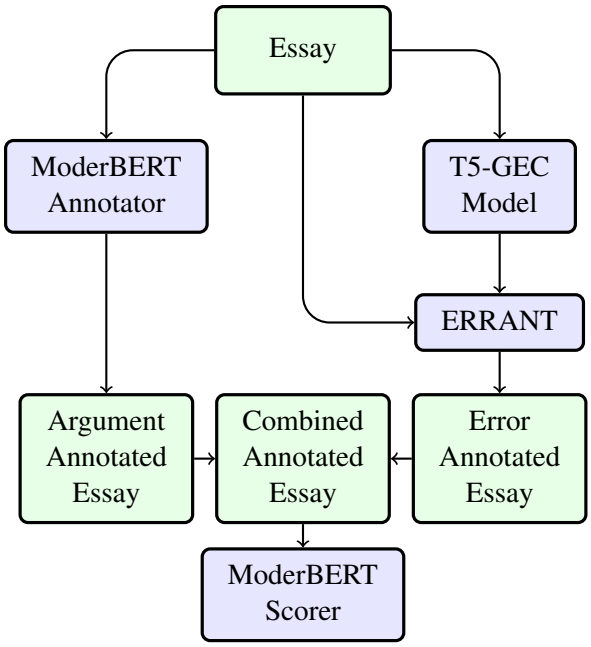

Figure 2: A diagram representing the scoring pipeline.

2012).

2.3.1 Annotator Evaluations

The standard described by the annotated argumentative essay dataset (Stab and Gurevych, 2014a,b) is reduced to a classification of IOB tags, where the governing statistic is an F1 score. Our guiding principle is the original rules for the competition[3], in which we use the ground truth and consider a match if there is over 50% overlap between the two identified components. Given a match using this rule, matching the argumentative type is considered a true positive (TP), unmatched components are considered false negatives (FN), while predicted label mismatches are considered false positives (FP). The final reported value is the F1 score, given by the familiar formula

$$F1 = \frac{2TP}{2TP + FP + FN}. \tag{5}$$

We can compare agreements for each label applied based on the ground truth. In this way, for each type of argumentative component type, we have a corresponding F1 score. In accordance with the rules of the competition, the final F1 statistic of interest is the macro average, given by the unweighted average over all the classes.

[3]https://www.kaggle.com/c/feedback-prize-2021/overview

2.3.2 Error Annotations

When it comes to annotating errors in the use of language, since the errors in the essays were not explicitly annotated by hand, we have no direct way of evaluating the accuracy of any annotations. We can only rely on the accuracy of the individual components. The T5 model we used has been evaluated in (Martynov et al., 2023) with respect to the JFLEG dataset (Napoles et al., 2017) and the BEA60k dataset (Jayanthi et al., 2020). According to those benchmarks, the accuracy of the model used is comparable to ChatGPT and GPT-4.

2.3.3 Automated Scoring Evaluations

For the scoring model, we use the standards for agreement specified by Williamson et al. (Williamson et al., 2012). The first and primary statistic used to describe agreement is Cohen's quadratic weighted kappa (QWK) (Cohen, 1960). Given scores between 1 and N, we define the weighted kappa statistic by the formula

$$\kappa = 1 - \frac{\sum w_{ij} O_{ij}}{\sum w_{ij} E_{ij}} \tag{6}$$

where O_{ij} is the number of observed instances where the first rater assigns a score of i and the second rater assigns a score of j, and E_{ij} are the expected number of instances that first rater assigns a score of i and the second rater assigns a score of j based purely on the random assignment of scores given the two rater's score distribution. This becomes the QWK when we apply the quadratic weighting:

$$w_{ij} = \frac{(i-j)^2}{(N-1)^2}. \tag{7}$$

The QWK takes values from -1 and 1, indicating perfect disagreement and agreement, respectively. It is often interpreted as the probability of agreement beyond random chance. The second statistic used is exact agreement, which is viewed as less reliable since uneven score distributions can skew it.

The last statistic used is the standardized mean difference (SMD). If y_t represents the true score and y_p represents the predicted score, then the SMD is given by

$$SMD(y_t, y_p) = \frac{\overline{y_p} - \overline{y_t}}{\sqrt{(\sigma(y_p)^2 + \sigma(y_t)^2)/2}}. \tag{8}$$

This statistic can be interpreted as a standardized relative bias. A positive or negative value indicates that the model is introducing some positive or

negative bias in the modeling process, respectively. Furthermore, when we restrict this calculation to scores for a specific demographic, assuming that demographic is sufficiently well-represented, the SMD is considered a gauge of the bias associated with the modeling for that subgroup.

3 Results

3.1 Annotation Accuracy

Using the 50% overlap rule, we present the number of true positives, false positives, and false negatives, and the resulting F1 score for each of the component types, excluding unannotated. These results are presented in Table 5.

	TP	FP	FN	F1
L	4332	270	95	0.960
P	6359	832	205	0.924
C1	20764	2057	1094	0.929
C2	1839	654	164	0.818
R	1443	495	106	0.827
E	18838	1653	1299	0.927
C3	5874	321	321	0.950

Table 5: The number of true positives (matched components and labels), false positives (matched components, unmatched labels), and false negatives (unmatched components) between the true annotations and the predicted annotations by component type.

These scores are exceptionally high, with the lowest performance given by the annotator's ability to discern counterclaims and rebuttals. As an indication of the annotated errors in language, the pipeline highlighted 2795 spelling errors, 1401 grammatical errors, and 201 punctuation or orthography errors. The pipeline used does not seem to be uncovering as many errors as expected.

3.2 Scoring Accuracy

Given that the classification head is typically randomly initialized, we were also interested in whether these results were stable. We trained the scorer 10 times and reported the average, minimum, and maximum agreement levels for each agreement statistic we listed above. These statistics can be found in Table 6.

All the models performed well above the human baseline. Out of the 10 separate trials, no model trained on the full-text alone scored as accurately as any of the models trained on the component annotated data or the data with both argumentative components and language errors annotated. An interesting observation is that the SMD, as calculated by (8), is only positive for models trained on the combined annotations, and that the models trained on component annotated text showed the most controlled SMDs.

3.3 Potential bias

To investigate the possibility of potential bias, we consider the SMD defined on subgroups. These SMDs are presented in Table 7.

Key	Orig.	Comp.	Error	Comb.
Female	0.12	0.11	0.12	0.06
WC	0.09	0.11	0.08	0.15
HL	-0.25	-0.25	-0.24	-0.19
BA	-0.17	-0.16	-0.20	-0.17
AP	0.44	0.45	0.53	0.52
Nat	-0.43	-0.41	-0.44	-0.21
Mix	0.08	0.07	0.12	0.01
ELL	-0.59	-0.60	-0.62	-0.54
DE	-0.36	-0.35	-0.37	-0.32
ID	-0.51	-0.48	-0.49	-0.42

Table 7: The bias in the various subgroups as measured by SMD for the particular subgroup.

While the resulting bias was higher than expected, a cursory look seems to suggest that the use of combined annotations is mitigating some of the biases rather than exacerbating them.

This work is one of a number of works that highlight the growing need to address the bias introduced by automation, especially for ELL students (Ormerod et al., 2022). This suggests we need to apply bias mitigation, which could be of the form of a regression-based system with adjusted cut-off points (Ormerod, 2022b), or some sort of reinforcement learning mechanism.

4 Discussion

The results of this study demonstrate that incorporating feedback-oriented annotations into automated essay scoring (AES) pipelines can significantly improve scoring accuracy and provide meaningful, interpretable insights for students. The work of Uto and Uchida (2020) suggests this is also true for traditional global features. What we propose

	QWK			Exa			SMD		
	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
Full Text Only	0.860	0.859	0.862	67.2	66.9	67.5	-0.023	-0.027	-0.018
Component Annotated	0.868	0.867	0.870	68.9	68.6	69.1	-0.013	-0.017	-0.009
Error Annotated	0.858	0.856	0.859	67.1	66.9	67.5	-0.025	-0.028	-0.022
Combined Annotations	0.866	0.867	0.870	68.4	68.0	68.9	0.021	0.016	0.025
Human Baseline	0.745								

Table 6: The average QWK, Exa, and SMD results on the test set for over 10 trials of training the scoring model.

is a realignment of AES to incorporate AWE elements so that we can provide students with more than just a score.

One of the most critical findings from this study concerns the potential for introduced bias, particularly among subgroups such as English Language Learners (ELLs). Our SMD analysis revealed notable disparities across demographic groups, echoing previous research on the disproportionate impact of automated systems on linguistically diverse populations. While the current pipeline demonstrates strong overall performance, these disparities underscore the importance of ongoing bias mitigation strategies. However, we know that SMDs can be unreliable, especially for subgroups with smaller populations. Future work should explore methods such as regression-based adjustments, fairness-aware training techniques, or reinforcement learning approaches that explicitly account for subgroup characteristics.

References

AI@Meta. 2024. Llama 3 Model Card.

Yigal Attali and Jill Burstein. 2006. Automated Essay Scoring With e-rater® V.2. *The Journal of Technology, Learning and Assessment*, 4(3). Number: 3.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv preprint*. ArXiv:2004.05150 [cs].

David Boulanger and Vivekanandan Kumar. 2020. SHAPed Automated Essay Scoring: Explaining Writing Features' Contributions to English Writing Organization. In *Intelligent Tutoring Systems*, pages 68–78, Cham. Springer International Publishing.

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.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46. Publisher: SAGE Publications Inc.

Scott A. Crossley, Perpetual Baffour, Yu Tian, Aigner Picou, Meg Benner, and Ulrich Boser. 2022. The persuasive essays for rating, selecting, and understanding argumentative and discourse elements (PERSUADE) corpus 1.0. *Assessing Writing*, 54:100667.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv preprint*. ArXiv:1901.02860 [cs, stat].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Technical Report arXiv:1810.04805, arXiv. ArXiv:1810.04805 [cs] type: article.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data Engineering for Scaling Language Models to 128K Context. *arXiv preprint*. ArXiv:2402.10171 [cs].

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2019. A Large-scale Dataset for Argument Quality Ranking: Construction and Analysis. *arXiv preprint*. ArXiv:1911.11408 [cs].

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv preprint*. Number: arXiv:2006.03654 arXiv:2006.03654 [cs].

Shi Huawei and Vahid Aryadoust. 2023. A systematic review of automated writing evaluation systems. *Education and Information Technologies*, 28(1):771–795.

Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. NeuSpell: A Neural Spelling Correction Toolkit. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 158–164, Online. Association for Computational Linguistics.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. Technical Report arXiv:2001.04451, arXiv. ArXiv:2001.04451 [cs, stat] type: article.

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.

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.

Susan Lottridge, Amy Burkhardt, Christopher Ormerod, Sherri Woolf, Mackenzie Young, Milan Patel, Harry Wang, Julius Frost, Kevin McBeth, and Julie Benson. 2024. Write On with Cambi: The development of an argumentative writing feedback tool.

Nikita Martynov, Mark Baushenko, Anastasia Kozlova, Katerina Kolomeytseva, Aleksandr Abramov, and Alena Fenogenova. 2023. A Methodology for Generative Spelling Correction via Natural Spelling Errors Emulation across Multiple Domains and Languages. *arXiv preprint*. ArXiv:2308.09435 [cs].

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.

Christopher Ormerod. 2022a. Short-answer scoring with ensembles of pretrained language models. *arXiv preprint*. ArXiv:2202.11558 [cs].

Christopher Ormerod, Amy Burkhardt, Mackenzie Young, and Sue Lottridge. 2023. Argumentation Element Annotation Modeling using XLNet. *arXiv preprint*. ArXiv:2311.06239 [cs].

Christopher Ormerod, Susan Lottridge, Amy E. Harris, Milan Patel, Paul van Wamelen, Balaji Kodeswaran, Sharon Woolf, and Mackenzie Young. 2022. Automated Short Answer Scoring Using an Ensemble of Neural Networks and Latent Semantic Analysis Classifiers. *International Journal of Artificial Intelligence in Education*.

Christopher Michael Ormerod. 2022b. Mapping Between Hidden States and Features to Validate Automated Essay Scoring Using DeBERTa Models. *Psychological Test and Assessment Modeling*, 64(4):495–526.

Christopher Michael Ormerod and Alexander Kwako. 2024. Automated Text Scoring in the Age of Generative AI for the GPU-poor. *arXiv preprint*. ArXiv:2407.01873 [cs].

Ellis Batten Page. 2003. Project Essay Grade: PEG. In *Automated essay scoring: A cross-disciplinary perspective*, pages 43–54. Lawrence Erlbaum Associates Publishers, Mahwah, NJ, US.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-training.

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. Technical Report arXiv:1910.10683, arXiv. ArXiv:1910.10683 [cs, stat] type: article.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. Language models and Automated Essay Scoring. *arXiv preprint*. Number: arXiv:1909.09482 arXiv:1909.09482 [cs, stat].

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A Simple Recipe for Multilingual Grammatical Error Correction. *arXiv preprint*. Number: arXiv:2106.03830 arXiv:2106.03830 [cs].

Noam Shazeer. 2020. GLU Variants Improve Transformer. *arXiv preprint*. ArXiv:2002.05202 [cs].

Mark D. Shermis and Ben Hamner. 2013. Contrasting State-of-the-Art Automated Scoring of Essays. pages 335–368. Publisher: Routledge Handbooks Online.

Christian Stab and Iryna Gurevych. 2014a. Annotating Argument Components and Relations in Persuasive Essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014b. Identifying Argumentative Discourse Structures in Persuasive Essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Assaf Toledo, Shai Gretz, Edo Cohen-Karlik, Roni Friedman, Elad Venezian, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. 2019. Automatic Argument Quality Assessment - New Datasets

and Methods. 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 5625–5635, Hong Kong, China. Association for Computational Linguistics.

Masaki Uto and Yuto Uchida. 2020. Automated Short-Answer Grading Using Deep Neural Networks and Item Response Theory. *AIED*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. Technical Report arXiv:1804.07461, arXiv. ArXiv:1804.07461 [cs] type: article.

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. *arXiv preprint*. ArXiv:2412.13663 [cs].

David M. Williamson, Xiaoming Xi, and F. Jay Breyer. 2012. A Framework for Evaluation and Use of Automated Scoring. *Educational Measurement: Issues and Practice*, 31(1):2–13. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1745-3992.2011.00223.x.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On Layer Normalization in the Transformer Architecture. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10524–10533. PMLR. ISSN: 2640-3498.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Text-Based Approaches to Item Alignment to Content Standards in Reading & Writing Tests

Yanbin Fu, Hong Jiao, Tianyi Zhou, Nan Zhang, Ming Li, Qingshu Xu, Sydney Peters, Robert W. Lissitz
University of Maryland, College Park

Abstract

Aligning test items to content standards is a critical step in test development to collect validity evidence based on content. Item alignment has typically been conducted by human experts, but this judgmental process can be subjective and time-consuming. This study investigated the performance of fine-tuned small language models (SLMs) for automated item alignment using data from a large-scale standardized reading and writing test for college admissions. Different SLMs were trained for both domain and skill alignment. The model performance was evaluated using precision, recall, accuracy, weighted F1 score, and Cohen's kappa on two test sets. The impact of input data types and training sample sizes was also explored. Results showed that including more textual inputs led to better performance gains than increasing sample size. For comparison, classic suprvised machine learning classifiers were trained on multilingual-E5 embeddings. Fine-tuned SLMs consistently outperformed these models, particularly for fine-grained skill alignment. To better understand model classifications, semantic similarity analyses including cosine similarity, Kullback-Leibler divergence of embedding distributions, and two-dimension projections of item embeddings revealed that certain skills in the two test datasets were semantically too close, providing evidence for the observed misclassification patterns.

1. Introduction

Item alignment is part of alignment defined as the consistency among assessments, content standards, and instructional practices (Smith & O'Day, 1990; Webb, 1997). The degree of item alignment to content standards is critical evidence for validity based on content. Item alignment is typically conducted manually by content experts. The process involves reviewing test items one by one and determining which content standards each item aims to measure. Experts rely on their subject-matter expertise and professional judgement to assess alignment. Thus, this approach has clear limitations. First, manual alignment is time-consuming and labor-intensive especially for large-scale assessments (Bier et al., 2019; Ding et al., 2025; Zhou & Ostrow, 2022). Second, reliance on expert judgement introduces subjectivity (Camilli, 2024; Khan et al., 2021). Third, as test items are designed to measure more complex domains and skills, incorporating multiple skills, domains or hierarchical label structures makes manual methods increasingly insufficient (Li et al., 2024).

To address these limitations, researchers started exploring using machine learning and natural language processing (NLP) techniques. These approaches aim to enhance consistency, reduce labor, and enable scalability in large-scale assessment (Qu et al., 2011). Broadly, automated item alignment methods can be classified into two categories: feature-based models and language model-based approaches. Feature-based methods can be further divided into two categories: linguistic feature-based models and embedding-based models.

Recently, advances in transformer-based language models have introduced new modeling approaches to automated item alignment. These include small language models (SLMs), such as BERT, RoBERTa, and DeBERTa, which are often

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 19–36

fine-tuned on labeled items to directly map item text to the content standards (e.g., Ding et al., 2025; Shen et al., 2021; Tan & Kim, 2024). Another emerging trend involves large language models (LLMs), such as GPT-4, which use prompting or fine-tuning strategies to classify or generate labels without additional training (Li et al., 2024; Liu et al., 2025; Moore et al., 2024).

2. Related Work

Automated item alignment is typically formulated as a classification task, where the goal is to assign items to predefined content standards based on item text. Early studies relied on feature-based models. In supervised or unsupervised classification tasks, test items were mapped to one or more content labels using classifiers such as support vector machines (SVM; Karlovcec et al., 2012; Yilmazel et al., 2007), Latent Dirichlet Allocation (LDA; Anderson et al., 2020), and XGBoost (Tian et al., 2022). For instance, Karlovcec et al. (2012) applied SVM and K-nearest neighbor (KNN) to classify ASSISTments math items into 106 content labels, while Pardos and Dabu (2017) used skip-gram and bag-of-words features for item alignment to 198 content labels. Extracted linguistic features included bag-of-words, TF-IDF, and keyword overlaps, which did not well capture contextual or sequential information.

With the rise of neural network models, convolutional neural networks (CNNs; Kim, 2014) and recurrent neural networks (RNNs; Schuster & Paliwal, 1997) were adopted. BiLSTM, a type of RNN, was particularly effective for sequence modeling. Sun et al. (2018) showed that BiLSTM outperformed classic methods (e.g., SVM) in English question alignment with an F1 score of 0.562 vs. 0.447. More approaches employed embeddings extracted from Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), or contextual embeddings from models like BERT (Devlin et al., 2019). For example, Tian et al. (2022) used Word2Vec embeddings and keyphrase features with XGBoost to align high school math items, outperforming baseline models such as VSM, SVM, NN, and LSTM.

SLMs such as BERT and RoBERTa have been applied in item alignment using fine-tuned methods. Shen et al. (2021) found that fine-tuned BERT outperformed both classic classifiers and BERT model without fine-tuning. Khan et al. (2021) developed the Catalog system to align items with the NGSS standards using BERT and GPT-based semantic similarity measures. Tan and Kim (2024) compared FastText+XGBoost, fine-tuned BERT-base/large, RoBERTa-large, and GPT-3.5 with prompting, reporting that RoBERTa-large consistently performed best. Similarly, Ding et al. (2025) proposed a RoBERTa-based model, which outperformed BiLSTM, BiGRU, and BERT in math item alignment.

LLMs like GPT-3.5 and GPT-4 have also been explored for item alignment via prompting. Wang et al. (2023) used GPT-4 to classify medical test items using zero- and few-shot prompts. Li et al. (2024) explored alignment as binary classification task, prompting LLMs with item text and candidate knowledge descriptions along with a self-reflection step that allow the model to re-evaluate and revise its initial prediction. Their results showed that GPT-4 performed best, achieving over 90% accuracy. Moore et al. (2024) used GPT-4 to directly generate knowledge components, simulating expert annotation and even constructing hierarchical ontologies.

In summary, feature-based models extract linguistic features or use embeddings as features but often lack task adaptation. Fine-tuned SLMs, though less explored, offer an efficient middle ground between classic machine learning models and costly LLMs, with less privacy concern and better scalability for large-scale assessment contexts.

To address gaps in the literature on automated item alignment in large-scale educational assessment, this study investigates how SLMs can be fine-tuned for item content

alignment in large-scale reading and writing assessments. Specifically, this study addresses the following research questions:

1. How do sample size and input data type affect the item alignment accuracy?
2. How do different SLMs perform in aligning test items to skill and domain categories?
3. Where do misclassifications occur?

3. Methods

3.1 Data

This study used 1270 items from the SAT Reading and Writing (RW) section, with 80% for training and 20% for testing. Additionally, 1052 items from the PSAT 8/9 RW section were used as an external test set to evaluate fine-tuned models' generalizability. Each item includeed a prompt, a question, four answer options, the correct answer or key, and a rationale explaining both correct and incorrect answers. Some items also contain graphs or tables, which were converted into text descriptions and LaTeX respectively. Each item measures one of the 10 skills nested under 4 content domains including *Standard English Conventions*, *Information and Ideas*, *Expression of Ideas*, and *Craft and Structure*. Skill labels include *Boundaries*, *Form*, *Structure and Sense*, *Command of Evidence*, *Inferences*, *Central Ideas and Details*, *Transitions*, *Rhetorical Synthesis*, *Words in Context*, *Text Structure and Purpose*, and *Cross-Text Connections*.

3.2 Sample Size and Input Data

To investigate the impact of sample size and input data on item alignment accuracy, the study experimented with different sample sizes and input data in the training dataset. BERT-base was first used for such exploration. Specifically, this study first sampled 500, 750, and 1000 items from the full 1270 dataset. Each subset was further split into training and test datasets using a ratio of 80% vs 20%. Their training datasets contained 400, 600, and 800 items respectively. The models' performance was evaluated on test sets. Nine input data types were experimented as listed below:

1. Prompt only
2. Prompt+table+figure
3. Prompt+table+figure+options
4. Prompt+table+figure+options+key
5. Prompt+table+figure+options+key+rationale
6. Prompt+table+figure+question
7. Prompt+table+figure+question+options
8. Prompt+table+figure+question+options+key
9. Prompt+table+figure+question+options+key +rationale

3.3 Models

To answer the second question about SLMs performance in item alignment, several SLMs were fine-tuned. This study explored both SLM-based modeling approaches and embedding-based classic supervised machine learning models. The 12 fine-tuned SLMs include BERT-base, BERT-large (Devlin et al., 2019), ALBERT-base (Lan et al., 2019), DistilBERT-base (Sanh et al., 2019), All-DistilRoBERTa (Liu et al., 2019; Sanh et al., 2019), ELECTRA-small, ELECTRA-base (Clark et al., 2020), RoBERTa-base, RoBERTa-large (Liu et al., 2019), DeBERTa-base (He et al., 2020), DeBERTa-large (He et al., 2021), and ConvBERT (Jiang et al., 2020).

For comparison, embeddings from multilingual-E5-large-instruct model were extracted using the CLS token and used to train supervised machine learning models including logistic regression, SVM, Naive Bayes, Random Forest, Gradient Boosting, XGBoost, LightGBM, MLP, and KNN.

3.4 Model Fine-Tuning

Prior to setting up the training configuration, this study conducted a series of exploratory experiments to evaluate the effects of different hyperparameter settings. Specifically, this study compared multiple learning rates (1e-5, 2e-5, and 3e-5), warm-up ratio (0 and 0.1), learning rate scheduler (linear and cosine), and checkpoints (epoch-wise and step-wise). Based on model performance with different settings, the following configuration was selected for all models. That is, models were trained with 15 epochs using the AdamW optimizer, a learning rate of 2e-5, a batch

size of 8, and a linear learning rate scheduler with a warmup ratio of 0.1. Each SLM was fine-tuned separately for the domain and skill alignment. Item input texts were tokenized using the tokenizer of each SLM and truncated to a maximum length of 512 tokens. The model performance was evaluated in terms of accuracy, recall, precision, weighted F1 score, and Cohen's kappa coefficient on both the SAT test dataset and the PSAT items.

3.5 Exploration for Misclassification

To understand the underlying causes of model misclassification, this study used a range of embedding-based analytical techniques. First, this study calculated all-pairwise cosine similarity between the selected skill groups with high rates of observed misclassification to quantify their semantic proximity in the embedding space. Second, To visualize the structure of the embeddings, this study applied three common dimensionality reduction techniques, including principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), and isometric mapping (ISOMAP), to project the item embeddings from the best performing models into a two dimensional space for the clustering patterns. Third, KL divergence was calculated between skill-specific embedding distributions. Lower KL scores suggest semantically similarity.

4. Results

4.1 Impact of Sample Size and Input Data

This study examined how input data and sample size affected item alignment accuracy using the BERT-base model. As shown in Table A.1 and A.2 in appendix, input data had a more substantial impact than sample size on both skill and domain alignment performance. Across all sample sizes, models trained with minimal inputs of "prompt_only" consistently yielded the lowest performance, while including more item components such as options, keys, rationales, and question improved model performance. For instance, in the skill alignment task with 400 training samples, weighted F1 score increased from 0.664 with "prompt_only" to 0.919 with all input data. However, the accuracy increase was not monotonic along with adding more input data. For example, when 400 items were used for training, adding the rationale led to decreased weighted F1 from 0.981 to 0.935.

It is worthy of note, adding question resulted in a sharp jump in alignment accuracy. For example, when 400 items were used for training, weighted F1 score for skill alignment increased from 0.664 with "prompt_only" to 0.893 with "prompt_table_figure_qtext". This dramatic increase was due to that many items in the same domain such as "Standard English Conventions" shared nearly identical question templates like "Which choice completes the text so that it conforms to the conventions of Standard English?" These question templates were likely to act as shortcut features, allowing models to memorize superficial patterns rather than learn the semantic relationship between content and skill or domain labels. To mitigate this issue, all questions was removed from the input data.

In contrast, increasing the training sample size from 400 to 800 yielded modest improvement, particularly when compared with the increase achieved through adding input data. For example, for skill alignment with "prompt_only," weighted F1 score improved from 0.664 for a sample size of 400 to 0.787 for a sample size of 600, whereas the same level of performance increase could be surpassed by adding more input data even with small sample sizes. A similar pattern was observed for domain alignment even though weighted F1 score was 0.919 with a sample size of 400 and "prompt_only" but F1 score increased to 0.927 with a sample size of 600 and all input data. These findings suggested that though larger training sample size increased accuracy, the more input data led to larger improvement in alignment accuracy more effectively.

4.2 The Impact of Hyper-Parameters for Fine-Tuning SLMs

To evaluate the effect of fine-tuning settings,

a full factorial experiment was conducted using BERT-base with different combinations of learning rate (1e-5, 2e-5, 5e-5), warm-up ratio (0.0, 0.1), learning rate scheduler (linear, cosine), and checkpoint strategy (epoch-wise, step-wise). The results showed that BERT-base model maintained strong performance across all hyper-parameter combinations. Weighted F1 scores, accuracy, and Cohen's kappa remained above 0.98 in nearly all cases, indicating a high degree of robustness to hyper-parameter choices.

4.3 Model Performance Comparison

Tables A.3 and A.4 in Appendix compared model performance on the SAT test set for skill and domain alignment. Across all metrics, fine-tuned SLMs significantly outperformed classical embedding-based classifiers. For skill alignment, ConvBERT and RoBERTa-large achieved perfect scores on all metrics, and even the worst performing ALBERT-base still performed well with weighted F1 of 0.943. Feature-based classifiers yielded lower performance, with weighted F1 scores ranging from 0.513 to 0.829. Among them, MLP showed the best performance. Domain alignment appeared to be an easier task, with most SLMs achieving nearly perfect results. Several models, including RoBERTa-large, ConvBERT, and DeBERTa-base, achieved perfect scores on all metrics. Feature-based classifiers also performed reasonably well, with weighted F1 scores generally above 0.84, indicating domain alignment task was easier.

The generalizability of fine-tuned SLMs was further tested on the PSAT dataset (Tables A.5 and A.6). While model performance dropped slightly compared to SAT test data, most models still performed well. For skill alignment, ELECTRA-base and RoBERTa-large remained the best performance with weighted F1 scores larger than 0.99, and DeBERTa-base and ALBERT-base performed well too with F1 score larger than 0.95. For domain alignment, DeBERTa-base performed best with all metrics having a value of 0.997. RoBERTa-base, RoBERTa-large also performed well with all metrics of 0.994. These findings suggest that models trained on SAT items can be generalized to PSAT item alignment when the same content framework are followed.

4.4 Exploration of Misclassification

Though the overall accuracy of aligning PSAT items was high using the model trained on SAT items, some skill-specific item alignment displayed high misclassification rate. Table A.7 presents F1 scores for skills on PSAT items. Several models, including BERT-base, BERT-large, ConvBERT, All-DistilRoBERTa, ELECTRA-small, RoBERTa-base, DeBERTa-large, and DistilBERT-base exhibited evident decrease in F1 scores on Skill 4 for *Inferences* and Skill 5 for *Central Ideas and Details*. Items for assessing these two Skills were often misclassified into Skill 8 for *Words in Context*.

To investigate misclassification, this study computed pairwise cosine similarities between iembeddings of items assessing Skills 4, 5, and 8 in SAT and PSAT. Results revealed high semantic similarity between Skill 4 and 8 with mean cosine similarity of 0.827 for SAT and 0.828 for PSAT and between Skill 5 and 8 with mean cosine similarity of 0.825 for SAT and 0.823 for PSAT.

Further, this study visualized the item-level embeddings using dimensionality reduction techniques, including PCA, t-SNE, and ISOMAP. The two-dimension projected embeddings for Skills 4 and 8, as well as Skills 5 and 8, showed considerable overlap across six plots. The four skill clusters occupied overlapping regions in the latent space, with no clear visual boundaries between them, indicating that the items shared highly similar semantic characteristics.

In addition, KL divergence was used to assess how PSAT Skills 4 and 5 align with each SAT skill in the embedding space. The results showed that SAT Skill 8 consistently exhibited low KL divergence (17.986 and 25.491) with the two PSAT skills, indicating the high semantic similarity. These results provide empirical evidence showing the semantic similarity between PSAT Skills 4/5 items and Skill 8 respectively where misclassification occurred.

5. Discussion and Conclusion

This study fine-tuned SLMs for automated item alignment in large-scale reading and writing assessments. Using SAT and PSAT data, items were aligned to both domains and skills, with skills nested within domains. The results demonstrated that fine-tuned SLMs substantially outperformed embedding-based classic machine learning models. Fine-tuned SLMs achieved high performance across all metrics, particularly in domain alignment. Even the weakest model, ALBERT-base, yielded weighted F1 score of 0.943. In contrast, embedding-based models trained on SLM yielded F1 scores ranging from 0.513 to 0.829, highlighting the superiority of end-to-end fine-tuning of SLMs.

More input data consistently outperformed the models trained with fewer input data. Increasing the sample size alone yielded relatively moderate improvements in model performance, especially when the input data were limited. However, the benefit of more input data was not monotonically increasing. With a sample size of 500, adding the rationale to the input data alongside the prompt, tables, figures, question, options, and key led to decreased performance. As sample size increased, this negative effect disappeared, suggesting an interaction between input data and sample size.

ELECTRA-base, RoBERTa-large, and DeBERTa-base demonstrated good generalizability on PSAT item alignment. Nevertheless, items measuring *Inferences* as well as *Central Ideas and Details* were frequently misclassified as *Words in Context*. Cosine Similarity and KL divergence analysis confirmed high overlapping in the embedding space across these skills, while two dimension projections using PCA, t-SNE, and ISOMAP further illustrated indistinct category boundaries.

Despite the promising results of SLMs in item content alignment demonstrated, this study has some limitations. First, items were all single-coded items. In some item content alignment, items may be double, triple, even multiple coded. Future research can explore more complex multi-coded item content alignment. Second, LLMs such as GPT-4 have shown promise in recent studies, they were not included in this study due to cost, transparency, and test security concerns. Future work may examine prompt-based LLMs alongside fine-tuned SLMs to assess their relative strengths in large-scale educational assessment programs.

In summary, this study evaluated multiple SLMs for automated item alignment to content standards. The investigation of the impact of sample size and input data types provided empirical evidence about these design factors in training SLMs for automated item alignment. The analyses related to misclassification errors help future studies to conduct quality control of any low performing cases. Though the current study used SAT and PSAT Reading and Writing items, the methods used for developing models for automated item alignment can be readily applied to state assessment programs when item alignment to content standards is needed.

References

Anderson, D., Rowley, B., Stegenga, S., Irvin, P. S., & Rosenberg, J. M. (2020). Evaluating content-related validity evidence using a text-based machine learning procedure. Educational Measurement: Issues and Practice, 39(4), 53-64.

Bhola, D. S., Impara, J. C., & Buckendahl, C. W. (2003). Aligning tests with states' content standards: Methods and issues. Educational Measurement: Issues and Practice, 22(3), 21–29.

Bier, N., Moore, S., & Van Velsen, M. (2019, March). Instrumenting courseware and leveraging data with the Open Learning Initiative (OLI). In Proceedings of the 9th International Conference on Learning Analytics & Knowledge (LAK19)

Butterfuss, R., & Doran, H. (2025). An application of text embeddings to support alignment of educational content standards. Educational Measurement: Issues and Practice, 44(1), 73–83.

Camilli, G. (2024). An NLP crosswalk between the Common Core State Standards and NAEP item

specifications. arXiv preprint arXiv:2405.17284.

Christopherson, S. C., & Webb, N. L. (2020). Alignment Analysis of Two Forms of the SAT with the Arizona Academic Standards for English Language Arts Grades 11–12, Algebra 1, and Geometry. Wisconsin Center for Education Products and Services.

Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.

Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., & Hu, G. (2020). Revisiting pre-trained models for Chinese natural language processing. In T. Cohn, Y. He, & Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 657–668). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.58

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (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 (pp. 4171–4186).

Ding, Z., Wang, X., Wu, Y., Cao, G., & Chen, L. (2025). Tagging knowledge concepts for math problems based on multi-label text classification. Expert Systems with Applications, 267, 126232.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. arXiv preprint arXiv:2002.06305.

Embretson, S. E., & Reise, S. P. (2000). Item response theory for psychologists. Psychology Press.

He, P., Gao, J., & Chen, W. (2021). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. arXiv preprint arXiv:2111.09543.

He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654.

Herman, J. L., Webb, N. M., & Zuniga, S. (2003). Alignment and college admissions: The match of expectations, assessments, and educator perspectives. Center for the Study of Evaluation, CRESST, UCLA.

Huang, T., Hu, S., Yang, H., Geng, J., Liu, S., Zhang, H., & Yang, Z. (2023). PQSCT: Pseudo-Siamese BERT for concept tagging with both questions and solutions. IEEE Transactions on Learning Technologies, 16(5), 831–846. https://doi.org/10.1109/TLT.2023.3275707

Nemeth, Y., Michaels, H., Wiley, C., & Chen, J. (2016). Delaware system of student assessment and Maine comprehensive assessment system: SAT alignment to the Common Core State Standards. Human Resources Research Organization.

Jiang, Z. H., Yu, W., Zhou, D., Chen, Y., Feng, J., & Yan, S. (2020). Convbert: Improving bert with span-based dynamic convolution. Advances in Neural Information Processing Systems, 33, 12837-12848.

Kane, M. (2006). Content-Related Validity Evidence in Test Development. In S. M. Downing & T. M. Haladyna (Eds.), Handbook of test development (pp. 131–153). Lawrence Erlbaum Associates.

Karlovčec, M., Córdova-Sánchez, M., & Pardos, Z. A. (2012). Knowledge component suggestion for untagged content in an intelligent tutoring system. In S. A. Cerri, W. J. Clancey, G. Papadourakis, & K. Panourgia (Eds.), Intelligent tutoring systems: 11th International Conference, ITS 2012, Chania, Crete, Greece, June 14–18, 2012. Proceedings (Lecture Notes in Computer Science, Vol. 7315, pp. 195–200). Springer. https://doi.org/10.1007/978-3-642-30950-2_25

Khan, S., Rosaler, J., Hamer, J., & Almeida, T. (2021). Catalog: An educational content tagging system. In Proceedings of the 14th International Conference on Educational Data Mining (EDM 2021). International Educational Data Mining Society.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1746–1751). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1181

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

Li, H., Xu, T., Tang, J., & Wen, Q. (2024). Automate knowledge concept tagging on math questions with LLMs. arXiv preprint arXiv:2403.17281.

Lima, P. S. N., Ambrosio, A. P., Felix, I., Brancher, J. D., & de Carvalho, D. T. (2018). Content analysis of student assessment exams. In 2018 IEEE Frontiers in Education Conference (FIE) (pp. 1–9). IEEE. https://doi.org/10.1109/FIE.2018.8659169

Liu, N., Sonkar, S., Basu Mallick, D., Baraniuk, R., & Chen, Z. (2025). Atomic learning objectives and LLMs labeling: A high-resolution approach for physics education. In Proceedings of the 15th International Learning Analytics and Knowledge Conference (LAK '25) (pp. 620–630). Association for Computing Machinery. https://doi.org/10.1145/3706468.3706550

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Martone, A., & Sireci, S. G. (2009). Evaluating alignment between curriculum, assessment, and instruction. Review of Educational Research, 79(4), 1332–1361.

McCormick, C., & Geisinger, K. F. (2017). Alignment Study Full Report. Buros Center for Testing.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Moore, S., Schmucker, R., Mitchell, T., & Stamper, J. (2024). Automated generation and tagging of knowledge components from multiple-choice questions. In Proceedings of the Eleventh ACM Conference on Learning @ Scale (L@S '24) (pp. 122–133). Association for Computing Machinery. https://doi.org/10.1145/3657604.3662030

Mosbach, M., Andriushchenko, M., & Klakow, D. (2020). On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. arXiv preprint arXiv:2006.04884.

Muennighoff, N., Tazi, N., Magne, L., & Reimers, N. (2022). MTEB: Massive text embedding benchmark. arXiv preprint arXiv:2210.07316.

Nemeth, Y., Michaels, H., Wiley, C., & Chen, J. (2016). Delaware System of Student Assessment and Maine Comprehensive Assessment System: SAT alignment to the Common Core State Standards – Final Report. Human Resources Research Organization.

Ozyurt, Y., Feuerriegel, S., & Sachan, M. (2025). Automated knowledge concept annotation and question representation learning for knowledge tracing. arXiv Preprint, arXiv:2410.01727. https://doi.org/10.48550/arXiv.2410.01727

Pardos, Z. A., & Dadu, A. (2017). Imputing KCs with representations of problem content and context. In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17) (pp. 148–155). Association for Computing Machinery. https://doi.org/10.1145/3079628.3079689

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543).

Peters, S., Zhang, N., Jiao, H., Li, M., Zhou, T., Lissitz, R., Fu, Y., & Xu, Q. (2025). Text-based approaches to item difficulty modeling in high-stakes assessments: A systematic review (MARC Research Report). University of Maryland.

Qu, B., Cong, G., Li, C., Sun, A., & Chen, H. (2012). An evaluation of classification models for question topic categorization. Journal of the American Society for Information Science and Technology, 63(5), 889-903.

Ramesh, R., Sasikumar, M., & Iyer, S. (2016). A software tool to measure the alignment of assessment instrument with a set of learning objectives of a course. In 2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT) (pp. 64–68). IEEE. https://doi.org/10.1109/ICALT.2016.10

Reimers, N., & Gurevych, I. (2021). all-distilroberta-v1 [Computer software]. Hugging Face. https://huggingface.co/sentence-transformers/all-distilroberta-v1

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681.

Sebastiani, F. (2002). Machine learning in automated text

categorization. ACM computing surveys (CSUR), 34(1), 1-47.

Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., McGrew, S., & Lee, D. (2021). Classifying math knowledge components via task-adaptive pre-trained BERT. In Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part I 22 (pp. 408-419). Springer International Publishing.

Smith, M. S., & O'Day, J. (1990). Systemic school reform. Journal of Education Policy, 5(5), 233–267. https://doi.org/10.1080/02680939008549074

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of documentation, 28(1), 11-21.

Sun, B., Zhu, Y., Xiao, Y., Xiao, R., & Wei, Y. (2018). Automatic question tagging with deep neural networks. IEEE Transactions on Learning Technologies, 12(1), 29-43.

Tan, C. S., & Kim, J. J. (2024). Automated Math Word Problem Knowledge Component Labeling and Recommendation. In International Conference in Methodologies and intelligent Systems for Techhnology Enhanced Learning (pp. 338-348). Cham: Springer Nature Switzerland.

Tian, Z., Flanagan, B., Dai, Y., & Ogata, H. (2022). Automated matching of exercises with knowledge components. In 30th International Conference on Computers in Education Conference Proceedings (pp. 24-32).

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., & Wei, F. (2024). Multilingual e5 text embeddings: A technical report. arXiv preprint arXiv:2402.05672.

Wang, T., Stelter, K., Floyd, J., O'Neill, T., Hendrix, N., Bazemore, A., Rode, K., & Newton, W. (2023). Blueprinting the future: Automatic item categorization using hierarchical zero-shot and few-shot classifiers. arXiv. https://arxiv.org/abs/2312.03561

Webb, N. L. (1997). Criteria for alignment of expectations and assessments in mathematics and science education. Research Monograph No. 6.

Yilmazel, O., Balasubramanian, N., Harwell, S. C., Bailey, J., Diekema, A. R., & Liddy, E. D. (2007). Text categorization for aligning educational standards. In 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07) (p. 73). IEEE. https://doi.org/10.1109/HICSS.2007.517

Yu, R., Das, S., Gurajada, S., Varshney, K., Raghavan, H., & Lastra-Anadon, C. (2021). A research framework for understanding education-occupation alignment with NLP techniques. In A. Field, S. Prabhumoye, M. Sap, Z. Jin, J. Zhao, & C. Brockett (Eds.), Proceedings of the 1st Workshop on NLP for Positive Impact (pp. 100–106). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.nlp4posimpact-1.11

Zhang, N., Jiao, H., Yadav, C., & Lissitz, R. (2025). Aligning SAT math to state math content standards: A systematic review (Technical report). Maryland Assessment Research Center, University of Maryland.

Zhou, Z., & Ostrow, K. S. (2022). Transformer-based automated content-standards alignment: A pilot study. In G. Meiselwitz (Ed.), HCI International 2022 – Late Breaking Papers: Interaction in New Media, Learning and Games (Vol. 13517, pp. 525–542). Springer. https://doi.org/10.1007/978-3-031-22131-6_39

Appendix

Table A.1

Performance of BERT-base Models across Sample Sizes and Input Data for Skill Alignment

Sample Sizes	Input Conditions	Accuracy	Precision	Recall	Weighted F1	Cohen's Kappa
400	prompt_only	0.700	0.690	0.700	0.664	0.662
	prompt_table_figure	0.810	0.813	0.810	0.801	0.786
	prompt_table_figure_options	0.900	0.904	0.900	0.897	0.886
	prompt_table_figure_options_key	0.880	0.886	0.880	0.876	0.864
	prompt_table_figure_options_key_rationale	0.920	0.926	0.920	0.919	0.909
	prompt_table_figure_qtext	0.890	0.915	0.890	0.893	0.876
	prompt_table_figure_qtext_options	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options_key	0.980	0.984	0.980	0.981	0.977
	prompt_table_figure_qtext_options_key_rationale	0.940	0.970	0.940	0.935	0.932
600	prompt_only	0.787	0.796	0.787	0.787	0.760
	prompt_table_figure	0.767	0.795	0.767	0.754	0.738
	prompt_table_figure_options	0.880	0.876	0.880	0.871	0.865
	prompt_table_figure_options_key	0.900	0.911	0.900	0.898	0.887
	prompt_table_figure_options_key_rationale	0.933	0.948	0.933	0.932	0.925
	prompt_table_figure_qtext	0.947	0.948	0.947	0.947	0.940
	prompt_table_figure_qtext_options	0.993	0.994	0.993	0.993	0.992
	prompt_table_figure_qtext_options_key	0.980	0.980	0.980	0.980	0.977
	prompt_table_figure_qtext_options_key_rationale	0.980	0.982	0.980	0.980	0.977
800	prompt_only	0.800	0.817	0.800	0.798	0.777
	prompt_table_figure	0.815	0.812	0.815	0.811	0.793
	prompt_table_figure_options	0.865	0.887	0.865	0.871	0.849
	prompt_table_figure_options_key	0.890	0.915	0.890	0.896	0.877
	prompt_table_figure_options_key_rationale	0.850	0.883	0.850	0.855	0.832
	prompt_table_figure_qtext	0.950	0.950	0.950	0.950	0.944
	prompt_table_figure_qtext_options	0.990	0.990	0.990	0.990	0.989
	prompt_table_figure_qtext_options_key	0.995	0.995	0.995	0.995	0.994
	prompt_table_figure_qtext_options_key_rationale	0.995	0.995	0.995	0.995	0.994

Table A.2

Performance of BERT-base Models across Sample Sizes and Input Data for Domain Alignment

Sample Sizes	Input Conditions	Accuracy	Precision	Recall	Weighted F1	Cohen's Kappa
400	prompt_only	0.920	0.929	0.920	0.919	0.891
	prompt_table_figure	0.930	0.931	0.930	0.930	0.905
	prompt_table_figure_options	0.960	0.963	0.960	0.960	0.945

	prompt_table_figure_options_key	0.970	0.973	0.970	0.970	0.959
	prompt_table_figure_options_key_rationale	0.990	0.990	0.990	0.990	0.986
	prompt_table_figure_qtext	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options	0.970	0.973	0.970	0.970	0.959
	prompt_table_figure_qtext_options_key	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options_key_rationale	0.980	0.981	0.980	0.980	0.973
600	prompt_only	0.900	0.900	0.900	0.900	0.866
	prompt_table_figure	0.900	0.902	0.900	0.899	0.866
	prompt_table_figure_options	0.953	0.958	0.953	0.954	0.937
	prompt_table_figure_options_key	0.953	0.960	0.953	0.954	0.937
	prompt_table_figure_options_key_rationale	0.927	0.934	0.927	0.927	0.902
	prompt_table_figure_qtext	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options_key	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options_key_rationale	0.987	0.987	0.987	0.987	0.982
800	prompt_only	0.885	0.888	0.885	0.885	0.846
	prompt_table_figure	0.900	0.901	0.900	0.900	0.866
	prompt_table_figure_options	0.965	0.966	0.965	0.965	0.953
	prompt_table_figure_options_key	0.960	0.962	0.960	0.960	0.947
	prompt_table_figure_options_key_rationale	0.940	0.947	0.940	0.941	0.920
	prompt_table_figure_qtext	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options_key	1.000	1.000	1.000	1.000	1.000
	prompt_table_figure_qtext_options_key_rationale	0.990	0.990	0.990	0.990	0.987

Table A.3

Model Performance on SAT Skill Alignment

Model	Precision	Recall	Accuracy	Weighted F1	Cohen's Kappa
BERT-base	0.996	0.996	0.996	0.996	0.996
BERT-large	0.989	0.988	0.988	0.988	0.987
ALBERT-base	0.949	0.945	0.945	0.943	0.938
ConvBERT	1.000	1.000	1.000	1.000	1.000
All-DistilRoBERTa	0.985	0.984	0.984	0.984	0.982
ELECTRA-base	0.992	0.992	0.992	0.992	0.991
ELECTRA-small	0.974	0.969	0.969	0.966	0.965
RoBERTa-base	0.996	0.996	0.996	0.996	0.996
RoBERTa-large	1.000	1.000	1.000	1.000	1.000
DeBERTa-base	0.985	0.984	0.984	0.984	0.982
DeBERTa-large	0.996	0.996	0.996	0.996	0.996
DistilBERT-base	0.992	0.992	0.992	0.992	0.991
Logistic Regression	0.538	0.646	0.646	0.563	0.593
SVM	0.642	0.701	0.701	0.643	0.658

Naive Bayes	0.764	0.744	0.744	0.749	0.713
Random Forest	0.591	0.610	0.571	0.513	0.554
Gradient Boosting	0.575	0.583	0.594	0.573	0.526
XGBoost	0.618	0.610	0.610	0.597	0.560
LightGBM	0.652	0.665	0.665	0.643	0.621
MLP	0.816	0.823	0.835	0.829	0.800
KNN	0.524	0.535	0.535	0.513	0.476

Table A.4

Model Performance on SAT Domain Alignment

Model	Precision	Recall	Accuracy	Weighted F1	Cohen's Kappa
BERT-base	0.996	0.996	0.996	0.996	0.995
BERT-large	0.996	0.996	0.996	0.996	0.995
ALBERT-base	0.967	0.965	0.965	0.965	0.952
ConvBERT	1.000	1.000	1.000	1.000	1.000
All-DistilRoBERTa	0.996	0.996	0.965	0.965	0.995
ELECTRA-base	0.996	0.996	0.996	0.996	0.995
ELECTRA-small	0.980	0.980	0.980	0.980	0.973
RoBERTa-base	1.000	1.000	1.000	1.000	1.000
RoBERTa-large	1.000	1.000	1.000	1.000	1.000
DeBERTa-base	1.000	1.000	1.000	1.000	1.000
DeBERTa-large	0.996	0.996	0.996	0.996	0.995
DistilBERT-base	0.992	0.992	0.992	0.992	0.989
Logistic Regression	0.879	0.878	0.878	0.878	0.834
SVM	0.901	0.894	0.894	0.894	0.857
Naive Bayes	0.839	0.827	0.827	0.827	0.767
Random Forest	0.812	0.807	0.783	0.781	0.735
Gradient Boosting	0.852	0.850	0.846	0.846	0.796
XGBoost	0.829	0.823	0.823	0.824	0.760
LightGBM	0.848	0.846	0.846	0.847	0.792
MLP	0.923	0.921	0.921	0.921	0.893
KNN	0.727	0.724	0.724	0.719	0.627

Table A.5

Model Performance on PSAT Skill Alignment

Model	Precision	Recall	Accuracy	Weighted F1	Cohen's Kappa
BERT-base	0.935	0.894	0.894	0.878	0.879
BERT-large	0.906	0.827	0.827	0.797	0.802
ALBERT-base	0.969	0.961	0.961	0.961	0.956
ConvBERT	0.902	0.887	0.887	0.870	0.871
All-DistilRoBERTa	0.931	0.907	0.907	0.887	0.895
ELECTRA-base	0.993	0.993	0.993	0.993	0.993
ELECTRA-small	0.744	0.760	0.760	0.722	0.728

RoBERTa-base	0.959	0.942	0.942	0.929	0.935
RoBERTa-large	0.994	0.994	0.994	0.994	0.994
DeBERTa-base	0.978	0.976	0.976	0.976	0.973
DeBERTa-large	0.927	0.894	0.894	0.868	0.879
DistilBERT-base	0.940	0.920	0.920	0.910	0.910
Logistic Regression	0.708	0.723	0.723	0.653	0.682
SVM	0.861	0.804	0.804	0.763	0.776
Naive Bayes	0.862	0.853	0.853	0.855	0.834
Random Forest	0.938	0.933	0.920	0.919	0.924
Gradient Boosting	0.881	0.879	0.883	0.882	0.864
XGBoost	0.917	0.914	0.914	0.914	0.903
LightGBM	0.938	0.937	0.937	0.936	0.929
MLP	0.963	0.963	0.961	0.961	0.958
KNN	0.695	0.695	0.695	0.687	0.655

Table A.6

Model Performance on PSAT Domain Alignment

Model	Precision	Recall	Accuracy	Weighted F1	Cohen's Kappa
BERT-base	0.947	0.934	0.934	0.934	0.912
BERT-large	0.986	0.985	0.985	0.985	0.980
ALBERT-base	0.892	0.820	0.820	0.803	0.762
ConvBERT	0.971	0.967	0.967	0.967	0.956
All-DistilRoBERTa	0.986	0.986	0.986	0.986	0.981
ELECTRA-base	0.928	0.904	0.904	0.902	0.872
ELECTRA-small	0.949	0.937	0.937	0.937	0.916
RoBERTa-base	0.994	0.994	0.994	0.994	0.992
RoBERTa-large	0.994	0.994	0.994	0.994	0.992
DeBERTa-base	0.997	0.997	0.997	0.997	0.996
DeBERTa-large	0.988	0.988	0.988	0.988	0.983
DistilBERT-base	0.940	0.926	0.926	0.925	0.901
Logistic Regression	0.899	0.898	0.898	0.899	0.864
SVM	0.934	0.933	0.933	0.933	0.911
Naive Bayes	0.860	0.857	0.857	0.857	0.810
Random Forest	0.959	0.959	0.953	0.953	0.945
Gradient Boosting	0.959	0.959	0.958	0.958	0.945
XGBoost	0.968	0.968	0.968	0.968	0.957
LightGBM	0.969	0.969	0.969	0.969	0.958
MLP	0.964	0.964	0.963	0.963	0.952
KNN	0.799	0.798	0.798	0.796	0.730

Table A.7

Skill Level Performance of Fine-Tuned Small Language Models for PSAT

Model	Skill 1	Skill 2	Skill 3	Skill 4	Skill 5	Skill 6	Skill 7	Skill 8	Skill 9	Skill 10
BERT-base	0.996	0.992	0.997	0.692	0.250	0.991	1.000	0.737	0.924	0.986

BERT-large	0.992	0.992	0.981	0.200	0.075	1.000	0.996	0.630	0.672	1.000
ALBERT-base	0.988	0.976	0.968	0.824	0.797	0.995	1.000	0.993	0.981	1.000
ConvBERT	0.992	0.996	0.972	0.150	0.678	1.000	1.000	0.741	0.900	1.000
All-DistilRoBERTa	0.992	0.992	0.963	0.108	0.683	0.926	1.000	0.937	0.917	0.986
ELECTRA-base	0.988	0.992	0.997	1.000	0.974	1.000	1.000	0.993	0.987	1.000
ELECTRA-small	0.988	0.988	0.672	0.000	0.000	1.000	1.000	0.619	0.653	0.839
RoBERTa-base	0.992	0.992	0.955	0.333	0.774	1.000	1.000	0.997	0.993	1.000
RoBERTa-large	0.996	0.996	0.991	0.986	0.974	1.000	1.000	0.997	1.000	1.000
DeBERTa-base	0.996	0.996	0.984	0.867	0.900	0.995	1.000	0.984	0.980	1.000
DeBERTa-large	0.992	0.984	1.000	0.056	0.798	0.995	1.000	0.800	0.695	1.000
DistilBERT-base	0.996	0.996	0.991	0.824	0.424	0.995	0.996	0.904	0.746	1.000

Note. Skill 1 = Boundaries; Skill 2 = Form, Structure, and Sense; Skill 3 = Command of Evidence; Skill 4 = Inferences; Skill 5 = Central Ideas and Details; Skill 6 = Transitions; Skill 7 = Rhetorical Synthesis; Skill 8 = Words in Context; Skill 9 = Text Structure and Purpose; Skill 10 = Cross-Text Connections.

Figure A.1

PCA Projection of Embeddings for Skill 4 (Inferences) vs. Skill 8 (Words in Context) for SAT and PSAT Items

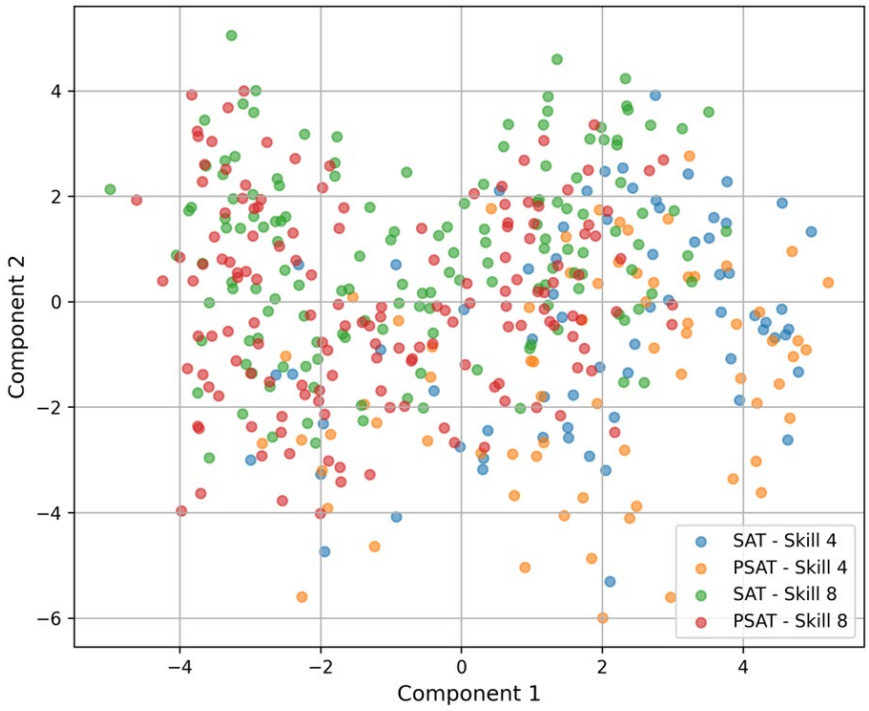

Figure A.2

t-SNE Projection of Embeddings for Skill 4 (Inferences) vs. Skill 8 (Words in Context) for SAT and PSAT Items

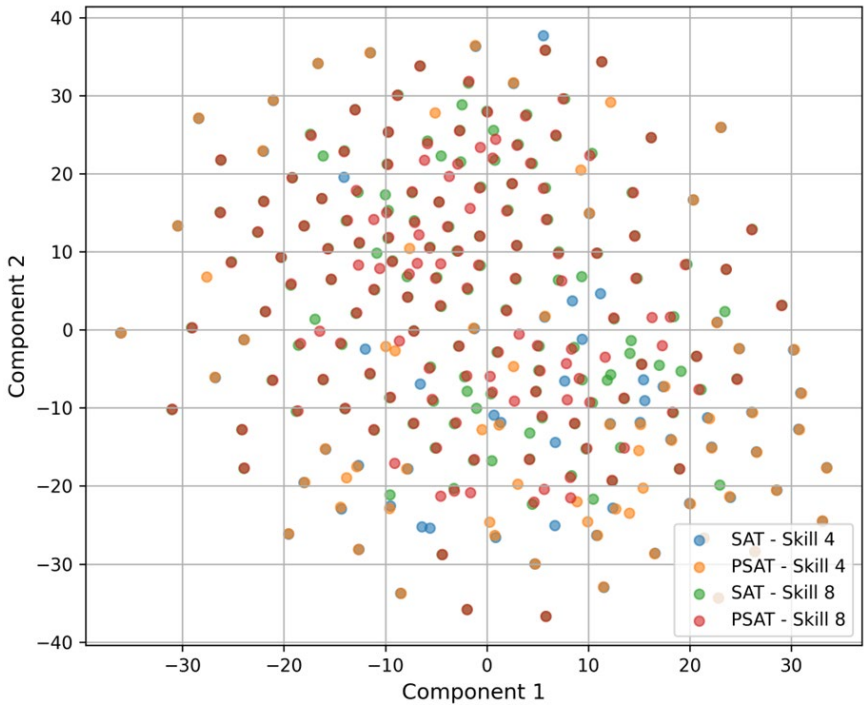

Figure A.3

ISOMAP Projection of Embeddings for Skill 4 (Inferences) vs. Skill 8 (Words in Context) for SAT and PSAT Items

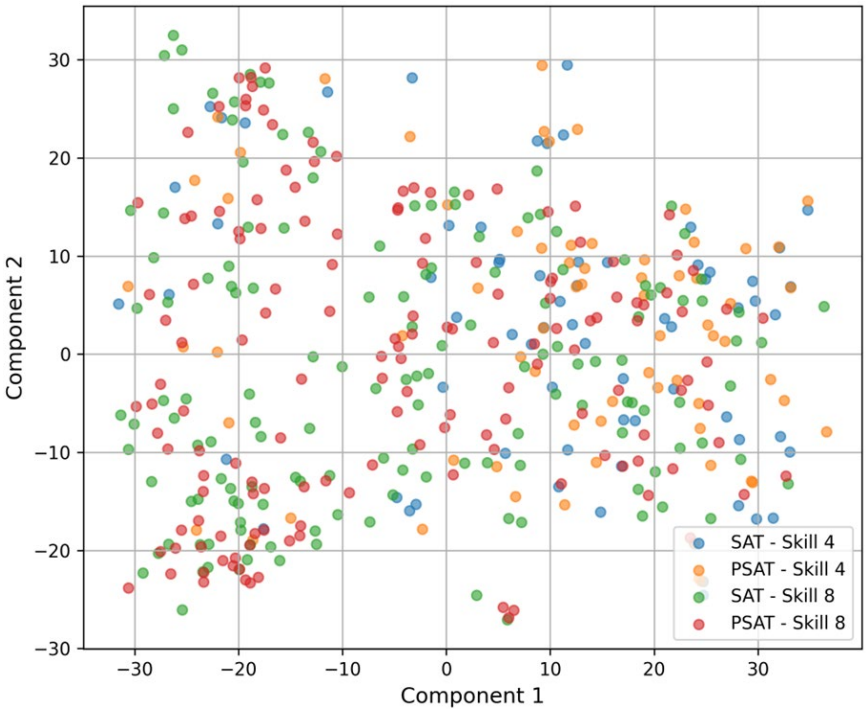

Figure A.4

PCA Projection of Embeddings for Skill 5 (Central Ideas and Details) vs. Skill 8(Words in Context) for SAT and PSAT Items

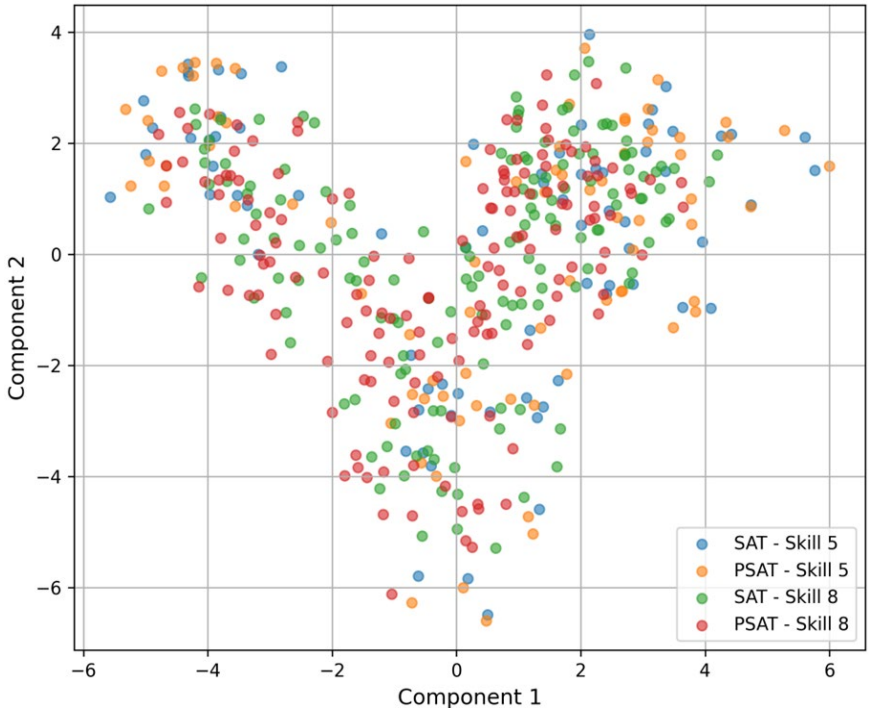

Figure A.5

t-SNE Projection of Embeddings for Skill 5 (Central Ideas and Details) vs. Skill 8 (Words in Context) for SAT and PSAT Items

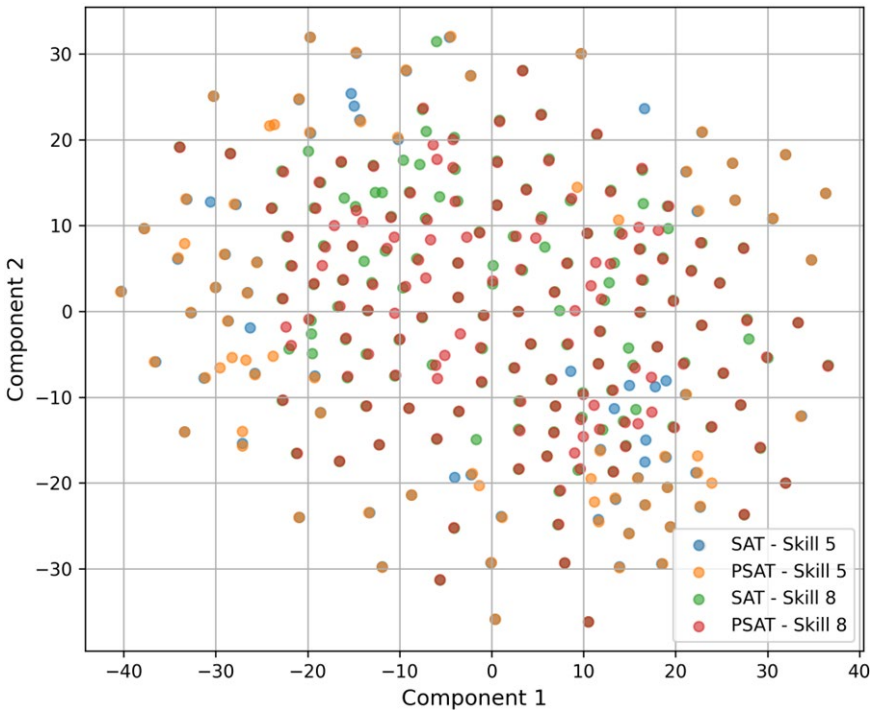

Figure A.6

ISOMAP Projection of Embeddings for Skill 5 (Central Ideas and Details) vs. Skill 8 (Words in Context) for SAT and PSAT Items

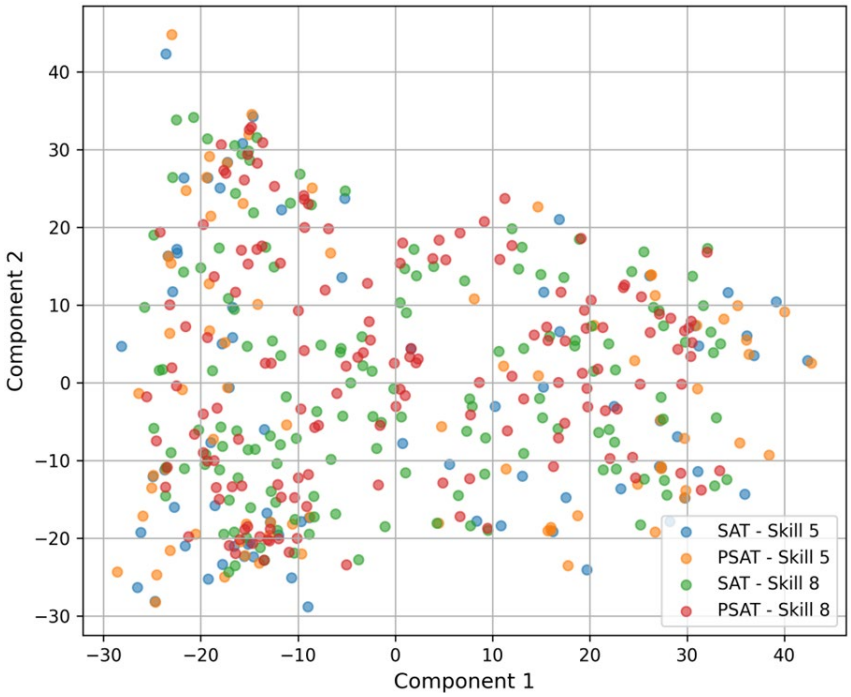

Table A.8

KL Divergence betweeen PSAT Skill 4 and Each SAT Skill

From	To	KL divergence
PSAT skill 4	SAT skill 1	32.927
PSAT skill 4	SAT skill 2	38.059
PSAT skill 4	SAT skill 4	42.588
PSAT skill 4	SAT skill 4	44.503
PSAT skill 4	SAT skill 5	40.996
PSAT skill 4	SAT skill 6	13.610
PSAT skill 4	SAT skill 7	26.869
PSAT skill 4	SAT skill 8	17.986
PSAT skill 4	SAT skill 9	44.342
PSAT skill 4	SAT skill 10	74.312

Table A.9

KL Divergence betweeen PSAT Skill 5 and Each SAT Skill

From	To	KL divergence
PSAT skill 5	SAT skill 1	44.096
PSAT skill 5	SAT skill 2	48.358
PSAT skill 5	SAT skill 3	48.800
PSAT skill 5	SAT skill 4	65.873
PSAT skill 5	SAT skill 5	41.134
PSAT skill 5	SAT skill 6	44.554
PSAT skill 5	SAT skill 7	40.371
PSAT skill 5	SAT skill 8	25.491
PSAT skill 5	SAT skill 9	43.649
PSAT skill 5	SAT skill 10	83.533

Review of Text-Based Approaches to Item Difficulty Modeling in Large-Scale Assessments

Sydney Peters, Nan Zhang, Hong Jiao, Ming Li, Tianyi Zhou
University of Maryland
sjpeters@umd.edu, hjiao@umd.edu

Abstract

Item difficulty plays a crucial role in evaluating item quality, test form assembly, and interpretation of scores in large-scale assessments. Traditional approaches to estimate item difficulty rely on item response data collected in field testing, which can be time-consuming and costly. To overcome these challenges, text-based approaches leveraging machine learning and natural language processing, have emerged as promising alternatives. This paper reviews and synthesizes 37 articles on automated item difficulty prediction in large-scale assessments. Each study is synthesized in terms of the dataset, difficulty parameter, subject domain, item type, number of items, training and test data split, input, features, model, evaluation criteria, and model performance outcomes. Overall, text-based models achieved moderate to high predictive performance, highlighting the potential of text-based item difficulty modeling to enhance the current practices of item quality evaluation.

1 Introduction

Large-scale assessments are often used to make high-stakes decisions such as grade promotion, professional certification, or college admission, so they must adhere to professional standards for test development to ensure validity, reliability, and fairness (AERA, APA, & NCME, 2014). The most common approach for estimating item difficulty has conventionally been conducted through field-testing, where newly created items are embedded in an operational test form. These items are used to collect item response data to estimate item parameters (e.g., difficulty) using classical test theory (CTT) or item response theory (IRT) and they are not used for scoring (Benedetto, 2023). Despite its ability to yield accurate item difficulty estimates, this approach has been criticized for being time-consuming and costly (AlKhuzaey et al., 2024; Hsu et al., 2018). Another approach for estimating item difficulty has been through expert ratings, though this is seldom used in developing large-scale assessments due to its subjective nature.

To address these limitations, text-based approaches for item difficulty prediction have offered a fast, objective, and scalable alternative. The timeline of these approaches followed a few noticeable trends. In the early stages, the literature was dominated by feature-based approaches that relied on manually defined variables that are hypothesized to influence item difficulty (e.g., Loukina et al., 2016; Perkins et al., 1995). Later, studies started to include word embeddings, which are numeric vectors representing the semantic relationships among words (e.g., Hsu et al., 2018). With the development of deep learning, embeddings were also extracted from deep neural networks, considering how words and phrases interact within the context of the text (e.g., Xue et al., 2020). Most recently, since 2020, transformer-based models, including small language models (SLMs) and large language models (LLMs), have been utilized, capturing nuanced semantic and contextual relationship (e.g., Li et al., 2025; Tack et al., 2024). These models have the potential to improve model predictive performance, but it comes at the cost of interpretability.

The goal of the present review is to highlight the recent developments in the use of machine learning and language-model based approaches for item difficulty prediction, with a focus on large-scale assessments. Several research questions guide our investigation: (1) What text-based methods, especially advanced language model-based approaches, were applied to predict item difficulty? (2) What domains and item types were most frequently investigated? (3) Which text-based features were most frequently investigated in classic machine learning models? (4) Which

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 37–47

evaluation criteria were used to assess model performance? (5) What was the distribution of evaluation outcomes? What does this reveal about the typical range and variability in item difficulty prediction modeling performance?

2 Related Work

The exploration of text-based approaches to model item difficulty has been ongoing for decades and a few studies have synthesized the research findings in a systematic way. Ferrara et al. (2022) conducted a domain-specific review that summarized 13 item difficulty modeling studies, focusing on high-stakes reading comprehension exams. This review found that statistical models such as ordinary least squares regression were utilized in every study and only two studies employed natural language processing (NLP) techniques. These findings highlight the emerging but still limited text-based methods for item difficulty estimation.

More recent reviews included articles that employed advanced models, which rapidly emerged from the mid-2010s (e.g., AlKhuzaey et al., 2024; Benedetto et al., 2023). Benedetto et al. (2023) conducted a narrative review of the literature and focused on approaches for question difficulty estimation from text from 38 studies published between 2015-2021. They provided a structured taxonomy to organize the approaches and analyzed the most effective methods in different scenarios. Results showed that, in general, simple models leveraging linguistic features performed just as well as end-to-end neural networks for language assessments; but for other subject domains (e.g., math, science) end-to-end neural networks, especially transformers led to increased performance. Their findings also highlighted a shift from readability and word-complexity features-based classic machine learning models to modern deep learning-based, NLP approaches.

AlKhuzaey et al. (2024) conducted a systematic review of 55 item difficulty prediction articles that placed no constraint on time frame, resulting in coverage from the years 1995 to 2022. Compared to previous reviews, they extended the scope to include an in-depth analysis of the most frequently investigated content domains, difficulty parameters, model features, models, input, item types, evaluation metrics, and the number of publications produced over the years. The results highlighted that linguistic play a critical role in estimating item difficulty, syntactic features are frequently captured using NLP tools to count textual elements, and with the development of neural language models, semantic features were increasingly explored.

Similarly, Luecht (2025) summarized years of item difficulty modeling research through 2022. The author explains that item difficulty modeling has evolved along two pathways: the strong theory pathway and the statistical control pathway. The strong theory pathway was most prevalent in the early years of item difficulty modeling research, and it is based on the idea that item design choices should be grounded in strong cognitive and learning theories. With the rise of machine learning and NLP-based text analytics, there has been a gradual shift to the statistical control pathway, that aims to identify variables that empirically explain item difficulty. Under this framework, the primary focus is improving model prediction performance, rather than aligning with cognitive theory.

Though AlKuzaey et al. (2024) and Bendetto et al. (2023) provided an in-depth summary of automated item difficulty prediction methods, they share similar limitations. All included articles were published no later than 2022 and they did not focus on large-scale assessments. Additionally, AlKhuzaey et al. (2024) included articles that used expert ratings as ground truth difficulty, but this is not a valid approach for item difficulty estimation in large-scale assessments due to subjectivity and inconsistency.

Given that language model-based approaches have vastly developed in the past three years (2023-2025), an updated synthesis of the literature is warranted. Another unique contribution of our review is the reporting of model performance outcomes, including the distribution of values obtained across evaluation metrics. This can act as a useful reference for future research by providing reference points for evaluating model performance.

3 Methods

We conducted a comprehensive literature search for articles published through May 2025 across multiple databased, including Google, Google Scholar, IEEE Xplore, ArXiv, Scopus, Springer, and ERIC. Additional searches were performed on the websites of the National Council on Measurement in Education (NCME) and a relevant competition platform (i.e., the NBME Item Difficulty Prediction Competition) to locate papers

submitted by participants. A Boolean search strategy was employed using keyword combinations in full text: *(item OR question) AND difficulty AND (AI prediction OR prediction using machine learning OR automatic prediction OR modeling).*

After an initial screening based on titles, 93 articles were identified. Next, 17 articles were excluded after reviewing the abstract and keywords for relevance, resulting in 76 articles. The full text of these articles was screened and 52 articles were excluded based on one or more of the following reasons: (1) the assessment was not large-scale, (2) the study focused on text complexity or readability rather than item difficulty, (3) the study did not focus on automated prediction based on item text (4) the article was a review, or (5) the item difficulty parameter was not obtained from item responses from human test-takers. A total of 24 articles remained for in-depth analysis.

Later, a forward hand-search was conducted to ensure that all related articles have been comprehensively included. For each included eligible article, we found all subsequent articles that cited it and conducted another round of screening. In this procedure, 19 additional articles were found, and after screening following the same exclusion criteria listed above, 13 articles were included in the review. In total, 37 articles were coded and analyzed for this review, consisting of conference papers (n = 20), journal articles (n = 7), research reports (n = 3), pre-prints (n = 5), and master's or doctoral theses (n = 2).

Since there could be more than one dataset or difficulty parameter analyzed in one paper, we treat these as separate studies. Consequently, 46 studies resulted from the 37 articles. To differentiate the number of articles from the number of studies, we used n for the number of articles and k for the number of studies, hereafter.

For each study we record the article information including title, authors, and publication year; dataset name, difficulty parameter, subject domain, item type, number of items, train and test dataset split, engineered features, models, evaluation criteria, and model performance. Descriptive analyses were performed, and results were reported using count-based aggregation and percentages. Model performance outcomes for each evaluation criterion with sufficient data across studies were summarized using descriptive statistics including minimum, maximum, median, mean, and standard deviation.

4 Results

4.1 Publication Year

Automated item difficulty prediction has come in two waves: one in the mid 1990s, and another beginning in the early 2010s. The resurgence is likely related to the peak of automated question generation research and the rise of computerized adaptive testing around 2014 to 2018 (AlKhuzaey et al., 2024; Kurdi et al., 2021), since item difficulty modeling is essential to evaluate the quality of newly created items. Ever since then research on this topic has been on the rise, with a large spike in 2024 due to the Building Educational Applications (BEA) shared task on automated item difficulty prediction and response time that launched in June 2024.

4.2 Item Difficulty Parameter

When the item difficulty parameter is a continuous parameter, item difficulty prediction is framed as a regression problem. In contrast, when it is defined using categorical levels (e.g., easy, medium, hard), it becomes a classification task (e.g., Hsu et al., 2018). In the context of large-scale exams, it was found that most item difficulty studies predicted a continuous value, which is consistent with the common practice of representing item difficulty in terms of either p-values or IRT b-parameters. Specifically, the most frequently reported methods were IRT b-parameter (k=14, 30.43%), transformed p-value (k=11, 23.91%), and traditional p-value (k=9, 19.57%). Other approaches including categorical difficulty levels (k=5, 10.87%), error rate (k=4, 8.70%), and Delta (k=3, 6.52%), were less common.

4.3 Subject Domain

Test subject domains included language proficiency (k = 23, 50.00%), medicine (k = 15, 32.61%), math (k = 4, 8.70%), science (k = 2, 4.35%), analytical reasoning (k = 1, 2.17%), and social studies (k = 1, 2.17%). Language proficiency and medicine dominate the literature likely due to that the high volume of large-scale exams in these domains made the data publicly available.

4.4 Item Type

Counting each item type once per study, a total of 60 item types were identified across the reviewed materials because several articles examined multiple item types within the same study. Multiple choice (MC) items accounted for the largest share, appearing 38 times (63.33%), followed by fill-in-the-blank reported eight times (13.33%), constructed-response items reported four times (6.67%), and matching items reported twice (3.33%). Each of the following item types were only reported once: complete-the-forms, notes, table, flowchart, or summary, complete-the-table, label the diagram, plan, or map, true/false, classifying, and sorting (3.33% each).

4.5 Number of Items

There was a wide range (348 to 106,210) in the number of items that were used across studies, showing great variability in dataset size. Most studies (k=17) used datasets between 500 and 2,000 items, largely because 11 studies used the data from the BEA shared task with 667 items. Only two studies used a very large dataset with more than 30,000 items (i.e., RACE++ (106,210) used in Benedetto, 2023; IFLYTEK (30,817) used in Huang et al., 2017)[1].

4.6 Training and Test Dataset Split

To develop different models for item difficulty prediction, a dataset is often divided into training, validation, and test datasets. The training set is used to learn patterns and relationships in the data, validation is used to fine-tune the model, and test is used to evaluate model performance on unseen data. Not all studies reported the percentage of data used for validation, so for consistency, training and validation percentages were combined for studies with three splits. We also note that several studies experimented with multiple train and test dataset splits (Benedetto, 2023; Bulut et al., 2024; Huang et al., 2017).

A wide variety of train/test data splits were observed. Reported as percentages they include: 40/60, 50/50, 60/40, 70/30, 75/25, 80/20, 83/17, 84/16, 85/15, 90/10, 93/7, and 95/5. The most common dataset split was 70% for training and 30% for testing, reported 14 times (28.00%), followed by 80/20 reported six times (12.00%), and 50/50, 90/10, and 95/5 reported three times each (6.00%, each). The remaining train and test data split combinations only appeared in two studies or less, and nine studies (18.00%) did not report this information.

4.7 Input

The input used to train the model refers exclusively to the original, unprocessed components of the item (i.e., item stem (lead-in and/or questions), correct answer, distractors, figures or reading passages when applicable). Again, some studies experimented with multiple combinations, and each was counted once per study, for a total count of 62. The most common combination of item components used as input was item stem, correct answer, and options, reported 19 times (30.65%) followed by item stem only reported nine times (14.52%), and item stem and correct answer reported six times (9.68%).

Some articles from the language proficiency tests included reading passages in the input; item stem, reading passage, correct answer and options was reported nine times (14.52%), and item stem and reading passage was reported seven times (11.29%). In general, utilizing all item components appears to be the most frequently used input text source for item difficulty modeling in the reviewed studies.

4.8 Features

A total count of 131 feature groups were found in 46 studies, which are generally categorized as hand-crafted features or embeddings. This can be further classified into five broad categories: hand-crafted linguistic features, features related to item metadata, LLM generated features, static embeddings, and contextualized embeddings.

The first category of hand-crafted features are linguistic features (79 counts, 60.31%), and they consist of lexical features (e.g., number of words, length of words), syntactic features (e.g., sentence count, use of conjunctions), morphological features (e.g., word stems, lemmas), semantic features (e.g., semantic similarity between item stem and options), readability indices (e.g., Flesch Reading Ease, Gunning FOG Index), and content specific features (e.g., number of text-based numerical

[1] RACE++ is a large-scale reading comprehension dataset; IFLYTEK refers to a language dataset from iFlytek, a Chinese technology company.

values for math). The second set of hand-crafted features include item metadata features (21 counts, 16.03%) including cognitive complexity, content standards, expert ratings, and item characteristics (e.g., number of choices for MC items). The third type of hand-crafted features are reasoning and thinking level features generated from LLMs (5 counts, 3.82%). Some examples of this type include first-token probability, choice-order sensitivity, and justification length.

As for embedding features, there are two categories: static embeddings and contextualized embeddings. Static embeddings (8 counts, 6.11%) include count-based model embeddings (e.g., Glove), and predictive model embeddings (e.g., word2vec). Contextual embeddings (18 counts, 13.74%) include deep learning-based embeddings (e.g., ELMo), word-level SLM embeddings (e.g., BERT-base, DistilBERT, MPNet), sentence-level SLM embeddings from sentence-BERT and Longformer, and embeddings extracted from LLMs.

4.9 Models

Among all reviewed studies, a total of 61 models have been explored 160 times, as it was common for studies to compare multiple models. The models were classified into three categories: classical machine learning models (94 counts, 58.75%), neural network based deep learning models (20 counts, 12.50%), and transformer-based language models (46 counts, 28.75%). Among the transformer models, 39 counts (84.78%) were SLMs and 7 (15.22%) were LLMs. For a full list of models included in the review see Appendix A.

Classical machine learning models typically rely on engineered features and are built on either statistical assumptions or algorithmic decision rules. They can be further classified as follows: linear and penalized regression models, decision tree-based models, probabilistic models, ensemble learning methods, kernel and distance-based models, and simple neural network-based models.

Neural network-based deep learning models use multiple layers that mimic the functioning of human neurons to learn complex, non-linear representations from data. In this review, we define this category as including only neural network models with more than one hidden layer and that are not based on attention mechanisms. These models consisted of basic neural network architectures, convolutional neural networks, bi-directional long short-term memory, and embeddings from language models (ELMo).

Transformer-based language models represent a specialized subset of deep learning in which the transformer architecture, characterized by self-attention mechanisms, is employed. The self-attention mechanism contextualizes each word in the text by considering its relationship with all other words, regardless of position or distance. This category contains both SLMs and LLMs, where we defined SLM as language models containing less than 1 billion parameters. SLMs consisted of BERT and its variants, long-sequence transformers, T5, and GPT-2. LLMs consisted of the models in the families of GPT, Llama, Mistral-7B, Gemma-7B, Qwen-2, Yi-34b, and Phi3, though Claude and Gemini families could be utilized as well.

We note several trends about the use of models through the years. Classical machine learning techniques have retained momentum due to their transparency, interpretability, efficiency, and robustness with small sample sizes. Neural network based deep learning models have been intermittently used beginning in 1995 and gaining moderate traction in 2019 to 2020. During this time, the use of neural-network-based deep learning models was approximately equal to the use of classical machine learning models. However, there has been a decline in the use of neural network based deep learning models that coincides with the rise of transformer-based models around 2020. Since then, classical machine learning models are still used, while transformer-based models have been used for both predicting item difficulty and generating embeddings as features.

4.10 Evaluation Criteria

Model performance was assessed using 23 unique evaluation criteria and their application depended on whether item difficulty prediction was a regression or classification task. In this review 43 studies were regression tasks and 3 were classification tasks. With regards to the regression tasks, the most common evaluation criteria were root mean square error (RMSE) (k=28, 31.82%), Pearson product moment correlation (k=17, 19.32%), and R^2 (k=13, 14.77%), mean absolute error (k=8, 9.09%), and mean square error (k=5, 5.68%). For classification tasks, exact accuracy was used for each study (k=3, 37.50%), and

adjacent accuracy was used when there were more than three difficulty levels (k=2, 25.00%). F1-score, recall, and precision were used once (12.50% each).

4.11 Model Performance

Appendix B presents a table that summarizes the best-performing value for each evaluation criteria used in the reviewed studies. Evaluation criteria that were used twice were summarized with the minimum and maximum values. It is important to note that although the table summarizes the outcomes from most commonly used evaluation criteria, values are not directly comparable across studies due to different subject domains and item difficulty parameters without a common scale. Instead, it should be used to provide a sense of what constitutes a "typical result" based on the range and distribution of values obtained in the literature.

For the most commonly used evaluation metric in regression tasks, RMSE, the summary was made for p-value, transformed p-value, and Rasch model b-parameter. The RMSE for studies using p-value ranged from 0.165 to 0.268 (N=6, M=0.216, SD=0.035), while RMSE for studies using transformed p-values ranged from 0.253 to 0.308 (N=10, M=0.291, SD=0.018). RMSE for studies using the Rasch model b parameter ranged from 0.354 to 1.295 (N=8, M = 0.740, SD = 0.297). The RMSE based on other difficulty parameters (i.e., 3PL, Delta, categorical levels), only contained one value for each, therefore a meaningful summary could not be produced.

The pattern persisted for other regression evaluation metrics. The Pearson correlation ranged from 0.040 to 0.870 (N=17, M=0.545, SD=0.225). R^2 values ranged from 0.208 to 0.788 (N=13, M=0.478, SD=0.200).

Similarly, the range for classification evaluation metrics also varied greatly across studies. Exact accuracy ranges from 0.325 to 0.806 (N=3, M=0.567, SD=0.241). However, the moderate to very high adjacent accuracy values (N = 2, 0.65 and 0.982) indicate that even when the model's prediction is not exactly correct, it is close, often just one category away from the true difficulty level.

5 Discussion

The aim of this review was to highlight and summarize trends in text-based item difficulty prediction research in the large-scale assessment setting, with a focus on advanced machine learning and language model-based approaches. A total of 46 studies from 37 articles were synthesized and results showed high potential for automated prediction of item difficulty parameters.

Our review makes several contributions to large-scale educational assessments. We provide large-scale educational assessment programs with foundational information that can be used to guide the implementation of automated approaches for item difficulty in the test development process. We provide practical insights into the optimal input and prompting strategies such as including all item components in the input and using a larger portion of the data for training leads to increased model performance. Our review can also be used as guidance for model and feature selection, outlining critical considerations for methodological choices. Overall, automated item difficulty modeling can be used to reduce the time and cost of traditional field testing to evaluate item quality.

Additionally, this review presents major contributions to the field of machine learning. Unlike previous reviews that have only synthesized the literature through 2022, the present review captures the significant growth of research in the past three years, as well as how methodological approaches have evolved since the 1990s. Another unique contribution of our review is the numerical distribution of model performance outcomes across all studies. The distribution of outcomes acts as a reference point that future researchers can use to set realistic expectations and to contextualize their model performance results.

Nonetheless, our review has a few limitations including the potential bias due to the overrepresentation of papers from the BEA shared task, lack of diversity in certain aspects of the datasets (e.g., item type, content domain), unexplained variability in model performance, and limited reporting of observed range of the IRT b-parameter. The latter complicates interpretation of scale-dependent evaluation metrics that were summarized in the model performance section. Future studies should prioritize dataset diversity, transparent reporting of methodology, and approaches that balance interpretability with the capabilities of state-of-the-art language models.

References

References marked with an asterisk (*) indicate studies included in the meta-analysis.

AlKhuzaey, S., Grasso, F., Payne, T. R., & Tamma, V. (2024). Text-based question difficulty prediction: A systematic review of automatic approaches. International Journal of Artificial Intelligence in Education, 34(3), 862-914.

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). Standards for educational and psychological testing. American Educational Research Association.

*Aryadoust, V. (2013, April). Predicting item difficulty in a language test with an Adaptive Neuro Fuzzy Inference System. In 2013 ieee workshop on hybrid intelligent models and applications (hima) (pp. 43-50). IEEE.

*Aryadoust, V., & Goh, C. C. (2014). Predicting listening item difficulty with language complexity measures: A comparative data mining study. CaMLA Work. Pap, 2, 1-16.

*Beinborn, L., Zesch, T., & Gurevych, I. (2015). Candidate evaluation strategies for improved difficulty prediction of language tests. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications (pp. 1–11). Denver, CO: Association for Computational Linguistics.

*Benedetto, L. (2023, June). A quantitative study of NLP approaches to question difficulty estimation. In International Conference on Artificial Intelligence in Education (pp. 428-434). Cham: Springer Nature Switzerland.

*Boldt, R. F., & Freedle, R. (1996). Using a neural net to predict item difficulty. *ETS Research Report Series*, *1996*(2), i-19.

*Boldt, R. F. (1998). GRE analytical reasoning item statistics prediction study. *ETS Research Report Series*, *1998*(2), i-23.

*Bulut, O., Gorgun, G., & Tan, B. (2024). Item Difficulty and Response Time Prediction with Large Language Models: An Empirical Analysis of USMLE Items.

*Dueñas, G., Jimenez, S., & Ferro, G. M. (2024, June). Upn-icc at bea 2024 shared task: Leveraging llms for multiple-choice questions difficulty prediction. In Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024) (pp. 542-550).

*El Masri, Y. H., Ferrara, S., Foltz, P. W., & Baird, J. A. (2016). Predicting item difficulty of science national curriculum tests: the case of key stage 2 assessments. *The Curriculum Journal*, 28(1), 59–82.

*Feng, W., Tran, P., Sireci, S., & Lan, A. (2025). Reasoning and Sampling-Augmented MCQ Difficulty Prediction via LLMs. arXiv preprint arXiv:2503.08551.

Ferrara, S., Steedle, J. T., & Frantz, R. S. (2022). Response demands of reading comprehension test items: A review of item difficulty modeling studies. *Applied Measurement in Education*, 35(3), 237-253.

*Fulari, R., & Rusert, J. (2024, June). Utilizing Machine Learning to Predict Question Difficulty and Response Time for Enhanced Test Construction. In Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024) (pp. 528-533).

*Gombert, S., Menzel, L., Di Mitri, D., & Drachsler, H. (2024, June). 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) (pp. 483-492).

*Groot, N. (2023). *Using Task Features to Predict Item Difficulty and Item Discrimination in 3F Dutch Reading Comprehension Exams* (Master's thesis, University of Twente).

*Ha, L. A., Yaneva, V., Baldwin, P., & Mee, J. (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 (pp. 11–20). Florence, Italy: Association for Computational Linguistics.

*He, J., Peng, L., Sun, B., Yu, L., & Zhang, Y. (2021). Automatically predict question difficulty for reading comprehension exercises. In 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 1398-1402).

*Hsu, F. Y., Lee, H. M., Chang, T. H., & Sung, Y. T. (2018). Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques. Information Processing & Management, 54(6), 969–984.

*Huang, Z., Liu, Q., Chen, E., Zhao, H., Gao, M., Wei, S., ... & Hu, G. (2017, February). Question Difficulty Prediction for READING Problems in Standard Tests. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 31, No. 1).

*Kapoor, R., Truong, S. T., Haber, N., Ruiz-Primo, M. A., & Domingue, B. W. (2025). Prediction of Item Difficulty for Reading Comprehension Items by Creation of Annotated Item Repository. arXiv preprint arXiv:2502.20663.

Kurdi, G., Leo, J., Matentzoglu, N., Parsia, B., Sattler, U., Forge, S., ... Dowling, W. (2021). A comparative study of methods for a priori prediction of MCQ difficulty. Semantic Web, 12(3), 449–465

*Li, M., Jiao, H., Zhou, T., Zhang, N., Peters, S., Lissitz, R. (2025). Item Difficulty Modeling Using Fine-Tuned Small and Large Language Models. Educational and Psychological Measurement. (accepted).

Luecht, R. M. (2025). Assessment engineering in test design: Methods and applications. Taylor & Francis.

*Loukina, A., Yoon, S. Y., Sakano, J., Wei, Y., & Sheehan, K. (2016, December). Textual complexity as a predictor of difficulty of listening items in language proficiency tests. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 3245-3253).

*McCarthy, A. D., Yancey, K. P., LaFlair, G. T., Egbert, J., Liao, M., & Settles, B. (2021). Jump-Starting Item Parameters for Adaptive Language Tests. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Pp. 883-899.

*Perkins, K., Gupta, L., & Tammana, R. (1995). Predicting item difficulty in a reading comprehension test with an artificial neural network. Language Testing, 12(1), 34-53.

*Qunbar, S. A. (2019). Automated item difficulty modeling with test item representations (ERIC No. ED601723). [Doctoral dissertation, The University of North Carolina at Greensboro].

*Razavi, P., & Powers, S. J. (2025). Estimating Item Difficulty Using Large Language Models and Tree-Based Machine Learning Algorithms. arXiv preprint arXiv:2504.08804.

*Rodrigo, A., Moreno-Álvarez, S., & Peñas, A. (2024, June). Uned team at bea 2024 shared task: Testing different input formats for predicting item difficulty and response time in medical exams. In Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024) (pp. 567-570).

*Rogoz, A. C., & Ionescu, R. T. (2024). UnibucLLM: Harnessing LLMs for Automated Prediction of Item Difficulty and Response Time for Multiple-Choice Questions. *arXiv preprint arXiv:2404.13343*.

*Sano, M. (2015). Automated capturing of psycho-linguistic features in reading assessment text. Paper presented at the annual meeting of the National Council on Measurement in Education (NCME), Chicago, IL.

SIGEDU. (2024). BEA 2024 Shared Task: Automated Prediction of Item Difficulty and Item Response Time. https://sig-edu.org/sharedtask/2024

*Štěpánek, L., Dlouhá, J., & Martinková, P. (2023). Item difficulty prediction using item text features: Comparison of predictive performance across machine-learning algorithms. Mathematics, 11(19), 1-30.

*Tack, A., Buseyne, S., Chen, C., D'hondt, R., De Vrindt, M., Gharahighehi, A., Metwaly, S., Nakano, F. K., & Noreillie, A.-S. (2024). ITEC at BEA 2024 shared task: Predicting difficulty and response time of medical exam questions with statistical, machine learning, and language models. In Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024) (pp. 512–521).

*Trace, J., Brown, J. D., Janssen, G., & Kozhevnikova, L. (2017). Determining cloze item difficulty from item and passage characteristics across different learner backgrounds. Language Testing, 34(2), 151–174.

*Veeramani, H., Thapa, S., Shankar, N. B., & Alwan, A. (2024, June). Large Language Model-based Pipeline for Item Difficulty and Response Time Estimation for Educational Assessments. In Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024) (pp. 561-566).

*Xue, K., Yaneva, V., Runyon, C., & Baldwin, P. (2020). Predicting the difficulty and response time of multiple choice questions using transfer learning. In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications.

*Xue, M., Han, S., Boykin, A., & Rijmen, F. (2025, April). *Leveraging Large Language Models in Predicting Item Difficulty*. Paper presented at the annual meeting of the National Council on Measurement in Education, Denver, CO.

*Yaneva, V., Baldwin, P., & Mee, J. (2019, August). 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* (pp. 11-20).

*Yaneva, V., Ha, L. A., Baldwin, P., & Mee, J. (2020). Predicting item survival for multiple-choice questions in a high-stakes medical exam. *In Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 6812–6818). European Language Resources Association.

*Yi, X., Sun, J., & Wu, X. (2024). Novel feature-based difficulty prediction method for mathematics items

using XGBoost-based SHAP model. Mathematics, 12(10), 1455.

*Yousefpoori-Naeim, M., Zargari, S., & Hatami, Z. (2024, June). Using machine learning to predict item difficulty and response time in medical tests. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)* (pp. 551-560).

*Zotos, L., van Rijn, H., & Nissim, M. (2024). Are You Doubtful? Oh, It Might Be Difficult Then! Exploring the Use of Model Uncertainty for Question Difficulty Estimation. arXiv preprint arXiv:2412.1183.

Appendix A. List of Models Included in the Review.

Classical Machine Learning Models:

1. *Linear and Penalized Regression Models:* Ordinary Least Square Regression, Principal Components Regression, Partial Least Squares Regression, Elastic Net Regression, Lasso Regression, Ridge Regression/Ridge (L2) Penalized Regression, Linear Logistic Test Model (LLTM)
2. *Decision Tree-Based Models:* Classification and Regression Trees (CART), Classification Trees, Decision Tree Regression, Extra Trees, Random Forest, Regression Trees
3. Probabilistic Models: Naive Bayes Classifier, Gaussian Processes, Probabilistic language model
4. *Ensemble Learning Models*: AdaBoost, Cat-Boost, Gradient Boosting, Gradient Boosting Decision Trees, Light Gradient Boosting Machine, XGBoost, XGBoost-based SHAP Model
5. *Kernel and Distance-Based Models*: k-Nearest Neighbors, Support Vector Machines
6. *Simple Neural Network Based Models*: Adaptive Neuro-Fuzzy Inference System (ANFIS), One Neuron Network (with no hidden layer), Three-Layer Backpropagation Neural Network (with only one hidden layer)

Neural Network Based Deep Learning Models:

1. *Basic Neural Network Architectures*: Artificial Neural Network (ANN), Multilayer-Perceptron (MLP), Dense Neural Network
2. *Convolutional Neural Networks (CNNs) and Variants*: Convolutional Neural Network (CNN), Attention-based CNN (ACNN), Hierarchical Attention-Based CNN (HBCNN), Multi-Scale Attention CNN (MACNN), Temporal CNN (TCNN), Temporal Attention CNN (TACNN).
3. *Bidirectional Long Short-Term Memory (Bi-LSTM)*

Transformer-Based Language Models
Small Language Models:

1. *BERT and its Variants*: BERT, BERT-ClinicalQA, Clinical-BERT, BioClinicalBERT, Bio_ClinicalBERT_emrqa, Bio_ClinicalBERT_FTMT, Clinical-BigBird, BioMedBERT, PubMedBERT, DistilBERT, ConvBERT, DeBERTa, RoBERTa, Electra, BioMedElectra
2. *Long-Sequence Transformers:* Longformer, Clinical-Longformer, Longformer-Base-4096, BigBird
3. *GPT-2*
4. *T5*

Large Language Models:

1. *GPT Family*: GPT-4, GPT-4o
2. Llama-7B
3. *Mistral*-7B
4. *Gemma*-7B
5. *Phi 3*

Appendix B. Model Performance Summary.

Evaluation Criterion	Count	Min	Max	Median	Mean	SD
Regression Tasks						
RMSE						
Based on p-value	6	.165	.268	.214	.216	.035
Based on transformed p-value	10	.253	.308	.297	.291	.018
Based on Rasch model	8	.354	1.295	.693	.740	.297
MSE	5	.013	.521	.064	.203	.227
MAE	7	.185	.58	.240	.307	.159
Correlation						
Pearson	17	.04	.87	.550	.545	.225
Spearman	4	.25	.790	.496	.508	.221
R-Squared	13	.208	.788	.525	.478	.200
Match	2	.757	.780	-	-	-
Classification Tasks						
Accuracy						
Exact	3	.325	.806	.569	.567	.241
Adjacent	2	.65	.982	-	-	-

Note. For studies that reported multiple models or evaluation criteria, only the best-performing value for each evaluation criterion was included. Only evaluation criteria that provided enough information ($k \geq 2$) for meaningful analysis were included. We also note that we report the same number of decimals that were presented in the articles.

Item Difficulty Modeling Using Fine-Tuned Small and Large Language Models

Ming Li, Hong Jiao, Tianyi Zhou, Nan Zhang, Sydney Peters, Robert W Lissitz
University of Maryland
minglii@umd.edu, hjiao@umd.edu

Abstract

This study investigates methods for item difficulty modeling in large-scale assessments using both small and large language models. We introduce novel data augmentation strategies, including on-the-fly augmentation and distribution balancing, that surpass benchmark performances, demonstrating their effectiveness in mitigating data imbalance and improving model performance. Our results showed that fine-tuned small language models such as BERT and RoBERTa yielded lower root mean squared error than the first-place winning model in the BEA 2024 Shared Task competition, whereas domain-specific models like BioClinicalBERT and PubMedBERT did not provide significant improvements due to distributional gaps. Majority voting among small language models enhanced prediction accuracy, reinforcing the benefits of ensemble learning. Large language models (LLMs), such as GPT-4, exhibited strong generalization capabilities but struggled with item difficulty prediction, likely due to limited training data and the absence of explicit difficulty-related context. Chain-of-thought prompting and rationale generation approaches were explored but did not yield substantial improvements, suggesting that additional training data or more sophisticated reasoning techniques may be necessary. Embedding-based methods, particularly using NV-Embed-v2, showed promise but did not outperform our best augmentation strategies, indicating that capturing nuanced difficulty-related features remains a challenge.

1 Introduction

Standardized tests rely on a detailed analysis of item attributes to ensure psychometric quality of items and test forms. A key attribute is the difficulty level of each item, which is related to the likelihood that an examinee will answer an item correctly. By producing items across a wide difficulty spectrum, it is expected the same measurement precision can be achieved at different ability levels. Moreover, while items that are more challenging typically result in longer response times, the duration of responses can also shed light on examinees' engagement and cognitive strategies, thereby enhancing the validity of the test outcomes. In addition, having a comprehensive understanding of item characteristics is critical for implementing advanced testing methods such as automated item generation, automated item selection in test form assembly, computerized adaptive testing, and individualized assessments (Baylari and Montazer, 2009; Wauters et al., 2012; Kubiszyn and Borich, 2024)

Typically, estimation of item difficulty and the response time required to answer items are derived from item response data gathered during field testing. However, field testing demands a large sample of examinees, which in turn drives up test administration costs (Bejar, 1983; Impara and Plake, 1998). As a result, researchers have explored alternative methods to predict item characteristics without resorting to actual test administration. One strategy involves soliciting difficulty estimates from domain experts and professionals involved in test development, yet this method has not consistently yielded reliable or satisfactory results (Wauters et al., 2012; Attali et al., 2014).

Another research avenue focuses on predicting item attributes based solely on the textual content of the items, including source passages, item stems, and response options (Hsu et al., 2018; Yaneva et al., 2019). This approach leverages text-mining techniques to extract both superficial features (e.g., word counts) and more complex features (e.g., semantic similarities between sentences), which are then used in sophisticated statistical models for prediction. In our study, we employed cutting-edge language models (LMs) for the development of predictive models aimed at estimating these item characteristics. This paper provides a comprehen-

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 48–55

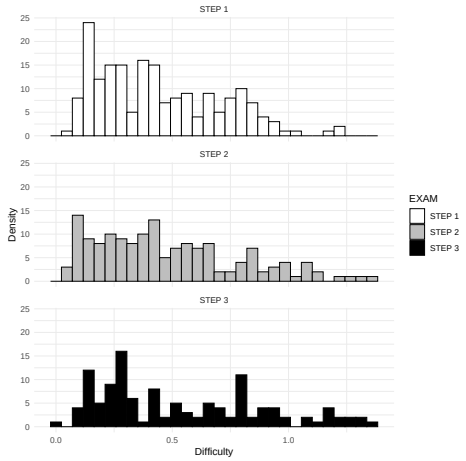

Figure 1: The Item Difficulty Distributions of USMLE Steps 1, 2, and 3 Training Datasets.

sive account of the methodologies implemented and the results obtained from our best-performing models for predicting item difficulty demonstrated using an empirical dataset.

2 Methods

2.1 Datasets

Building on this line of research, this study used data from the National Board of Medical Examiners (NBME) initiated BEA 2024 Shared Task [1] to automate the prediction of item difficulty and response time. The released dataset included 667 items that were previously used and have since been retired from the United States Medical Licensing Examination® (USMLE®)—a series of high-stakes exams [2] that inform medical licensure decisions in the United States. These items, drawn from USMLE Steps 1, 2 Clinical Knowledge (CK), and 3, span a diverse range of topics relevant to medical practice. During the BEA 2024 Shared Task, participating research teams were challenged to leverage NLP techniques using 466 items to develop models to predict item difficulty.

Subsequently, the models developed from the initial phase were applied to a second dataset containing 201 items. This testing set shared the same structural characteristics as the first, except that the values for item difficulty and response time were initially concealed. These values were disclosed only after the BEA 2024 submission deadline, thereby facilitating a fair evaluation of the model's performance in predicting outcomes.

[1] https://sig-edu.org/sharedtask/2024

[2] https://www.usmle.org/step-exams

Figure 1 presents the item difficulty distributions for Steps 1, 2, and 3 USMLE in the training data. The larger values indicate more difficult items. The item difficulty for each Step exam is not evenly distributed. The data imbalance issue is severely critical for this task as the majority of the data lies in the low-difficult range, and only a small number of items is difficult items. This data sparsity in some item difficulty ranges may cause non-representation issues when the item difficulty modeling is developed.

2.2 Models and Methods for Item Difficulty Prediction

This study explored a variety of different methods ranging (i) from small language models (SLM) to large language models (LLMs); (ii) from embedding-based methods to auto-regressive methods; and (iii) from finetune-based methods to inference-only methods. In addition, LLMs with different fine-tuning and prompting techniques were explored. These explorations thoroughly covered the most widely accepted methods off the shelf, which can serve as detailed guidance for future endeavors to other datasets.

2.2.1 SLMs: BERT and its Variants

This study started experimentation with directly fine-tuning small language models for difficulty prediction. We treat this task as a regression task that directly predicts the difficulty value for each item. The models incorporated are mostly encoder-only language models but also some models with encoder-decoder structures, including BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), DistillBERT (Sanh et al., 2020),deBERTa (He et al., 2021),ELECTRA (Clark et al., 2020), ConvBERT (Jiang et al., 2020), T5 (Raffel et al., 2023), BioClinicalBERT (Alsentzer et al., 2019), and PubMedBERT (Gu et al., 2021). These models intend to set the baseline for comparison. Further, data augmentation was implemented to enhance the prediction accuracy.

2.2.2 Ensemble of SLMs with Majority Voting

Usually, ensemble models are expected to perform better than single-base models. Thus, we explored a commonly used majority voting method to generate more robust results from single SLMs. For the regression task, we used majority voting, which is the average predicted value from different models that participate in the voting process. Compared

with single-model predictions, majority voting is expected to be more robust since the training process is always affected by randomness, and the voting may alleviate the effects of randomness.

2.2.3 SLMs with Data Augmentation

The NLPAug (Ma, 2019) package is utilized for implementing data augmentation. Two types of data augmentation strategies were explored in this study. The two types of data augmentation strategies include: *(i) Augmentation on the fly*, and *(ii) Augmentation with distribution balancing*.

Augmentation-on-the-Fly

In this strategy, we randomly augment original training samples every time it is sent to the model for training. Under this circumstance, all the samples seen by the model are a random augmented version of the original sample, which means the model will not see any identical samples during the epochs of training. This strategy is mostly widely used in the machine learning community as it prevents the model from overfitting to the samples.

Augmentation with Distribution Balancing

This strategy is more complicated as it is specially designed for this task. As shown in Figure 1, the data imbalance issue is severely critical for this task. The majority of the data lies in the low-difficult range, and only a small set of data has high difficulties, e.g., above 0.8. Thus, this imbalance issue causes most of the methods to fail for the prediction, even for our Augmentation-on-the-Fly strategy, as it does not change the frequencies of each sample trained by the model.

Thus, to deal with this issue, we were motivated to balance the sample sizes across the whole distribution, i.e., generate more data for the difficulty levels with lower density and fix them during training. As a regression task, it is naturally difficult to make the data more balanced as they are not as discrete as classification tasks. So, in order to solve it, we first separate all the data samples into 20 bins by fixed intervals and then merge the adjacent bins such that there are at least 2 samples in each bin. This preliminary process converts the consecutive values into discrete bins. Then, we randomly sample 1 instance from each bin to form the validation set. The remaining instances form the training set. This separation ensures that the validation set is balanced enough for fair evaluation. In the remaining bins of the training set, we then randomly augment the existing training samples into a predefined count, i.e., 8 in our experiment. Under this circumstance, the sample counts across all the bins, i.e., the whole distribution, become largely balanced. Then, during training, we fix these samples and do not do augmentation during training. This strategy largely alleviates the distribution imbalance issue and prevents the model from overfitting to high-frequent samples.

Ensemble of Two Data Augmentation Strategies

Both strategies have their own merits. Thus, we further implement an ensemble strategy. For each given instance, each of the above two models generates its own prediction, and then these two predicted values are averaged to simulate the ensemble of the two strategies.

2.2.4 SLMs with LLM Rationales

The training dataset contains only the questions, options, and answers; however, the goal of this study is to predict the difficulties of these items. Thus, there exists a critical gap between the input (item text) and the target (item difficulty). If the input does not contain any information regarding the difficulty, obviously, it will be difficult for SLMs to predict the item's difficulties.

Given the strong reasoning capabilities of LLMs and the potential insights that the chain-of-thought (CoT) prompting technique may provide in reasoning, we hypothesize that incorporating additional rationales that specifically analyze the difficulty of the given items will benefit SLMs in capturing the representative key features in item difficulty modeling. Thus, motivated by the success of CoT (Wei et al., 2023), we employed GPT-4 (OpenAI et al., 2024) to generate a detailed analysis of the item difficulties of different instances, which we refer to as rationales. Then, we concatenate these rationales with the original item text for training the SLMs. The SLMs experimented with are BERT, T5, and Longformer (Beltagy et al., 2020).

2.2.5 BERT with Step-Wise Data Augmentation

Another critical issue of the existing training dataset is its imbalanced nature across exams in the three steps. As shown in Figure 1, the number of step 1 exam items is larger than that for the step 2 and step 3 exam items. To solve this issue, a step-wise data augmentation strategy was implemented using the Python package: NLPAug(Ma, 2019) to augment data in steps 2 and 3 exams for training the BERT model as it was the best-performing base model. Thus, the proposed step-wise data augmen-

tation strategy yielded more augmented data points for step 2 and step 3 exams, while fewer augmented data points for step 1 exam. Further, this step-wise data augmentation method was applied to augment data for the BERT model augmented with the LLM rationales already.

2.2.6 LLMs Finetuning and In-context Learning

All the above methods are based on small language models. In addition, we explored how LLMs performed on this task. Finetuning on LLMs is typically the most commonly used technique when we need LLMs to handle a new task. However, most of the modern LLMs follow the decoder-only structure, which predicts each token in an auto-regressive manner. The modern decoder-only LLMs are more capable of text generation rather than regression tasks, especially when the task is not previously learned during the pretraining phase.

The first category of method we explored was the finetune-based method. Since we only have hundreds of training samples, which is typically not enough for LLMs, we select Phi3 (Abdin et al., 2024) as our base LLM and utilize the full finetuning and LoRA finetuning strategies (Hu et al., 2021). In addition, In-context Learning (ICL) (Brown et al., 2020) is another widely used method for LLM prediction. ICL is less affected by the sparsity of the training data. Thus, we also explored using ICL. One of the biggest advantages of ICL is that it does not require training, thus, it can be used in any LLM, even for closed-source LLMs like GPT-4.

2.2.7 LLMs Embeddings

Previous explorations on LLMs employed the auto-regressive manner of decoder-only structures, which might not be able to well capture the distribution from training data. Typically, embeddings are more effective for regression tasks. Thus, we further explored using LLM embeddings for the prediction, specifically, we utilized the most current state-of-the-art embedding model NV-Embed-v2 (Lee et al., 2024) as the encoder for item difficulty modeling. Further, we trained several additional layers for the difficulty prediction. We also explored combining the benefits of the auto-regressive manner and the benefits of the embedding method together by first generating rationales that specifically analyze the difficulty of the given items, then the SOTA embedding LLMs are utilized to capture the overall distribution of the training data.

2.3 Evaluation of Model Performances

This study used the root mean squared error (RMSE) to evaluate the model performance. This is the evaluation criterion used in the competition. To use the results from the competition as a reference to evaluate the performance of the models and the methods we proposed in this study, we computed RMSE for each model and method explored.

3 Results

3.1 SLMs: BERT, Its Variants, and the Ensemble Models

The performance of the fine-tuned SLMs and the ensemble models is summarized in Table 1. As noted, BERT and Roberta have the top performances, with BERT yielding the smallest RMSE. Contrary to our expectation, utilizing BERT trained with medical-related data (BioClinicalBERT and PubMedBert) does not show an evident improvement in model performance in predicting item difficulty, which might be caused by the class imbalance in the potential distribution gaps. These two models might have a better general understanding of medical-related knowledge, but this knowledge still has a gap in understanding and reasoning item difficulty, which is a basic concept in the psychometric analysis of test items.

The ensembled BERT with a Majority Voting strategy (with RMSE of 0.2981) also exceeds the first place on the leaderboard, showing the effectiveness of this strategy. However, BioClinicalBERT and PubMedBert do not benefit from the ensemble strategy. It is reasonable that the performances of these models are originally not good, and voting by multiple not high-performing models may not necessarily further increase the performance.

3.2 SLMs with Data Augmentation

Our best-performing model is BERT with an ensemble of two types of data augmentation strategies. Both data augmentation strategies (Augmentation-on-the-fly and Augmentation with distribution balancing result in excellent performances that exceeded the first place (RMSE: 0.299) on the leaderboard. Augmentation-on-the-fly yielded RMSE of 0.2975 while augmentation with distribution balancing yields 0.2985 of RMSE, which also exceeds the first place on the leaderboard. Further, the en-

Table 1: Performances for Fine-Tuned SLMs: BERT and its Variants and SLMs with Majority Voting

Model	RMSE
BERT	0.2990
RoBERTa	0.2997
DistilBERT	0.3022
DeBERTa	0.3060
ELECTRA	0.3026
ConvBERT	0.3015
T5	0.3023
BioClinicalBERT	0.3043
PubMedBert	0.3067
BERT (Majority Voting)	0.2981
BioClinicalBERT (Majority Voting)	0.3052
PubMedBert (Majority Voting)	0.3086

Table 2: Performance of the BERT Models with Different Data Augmentation Strategies and the Top Performing Models in the Leaderboard.

Rank	Studied Methods/Team Name	RMSE
Ours	Ensemble of Two Strategies	0.2926
Ours	Augmentation-on-the-Fly	0.2975
Ours	Augmentation with Balancing	0.2985
1	electra	0.299
2	UPN-ICC (run1)	0.303
3	Roberta	0.304
4	RandomForest	0.305
5	ENSEMBLE	0.305
6	Predictions	0.305
7	FEAT	0.305
8	ROBERTA	0.306

semble of these two strategies further leads to an extraordinary performance of 0.2926 in terms of RMSE, which also exceeds the first place on the leaderboard by a really large margin. The performances are further compared in Table 2.

3.3 SLMs with LLM rationales

We used GPT-4 to generate detailed rationales for the difficulties of the items. We concatenated the generated rationales with the original item text for training BERT, T5, and Longformer. The model performances are summarized in Table 3. The models did not perform as effectively as expected. This might be related to the sparsity of the training data in each step exam. When the sample size of the training data is relatively small for each step exam, even the generated rationales may not be able to capture the key item characteristics that distinguish them in terms of item difficulty, though the sample size for each step exam has been increased.

Table 3: Performances for SLMs with LLM Rationales

Model	RMSE
BERT + GPT4 rationales	0.3029
T5 + GPT4 rationales	0.3047
Longformer + GPT4 rationales	0.3050

Table 4: Performances for SLMs with Step-wise Data Augmentation

Model	RMSE
BERT + Step	0.3009
BERT + GPT4 rationales + Step	0.3000

3.4 BERT with Step-Wise Data Augmentation

With the step-wise data augmentation strategy, more synthetic data points were generated to increase more item samples for step 2 and step 3 exams, while slightly more items for the step 1 exam. Step-wise data augmentation was applied to both the BERT model and the BERT model with rationals as data augmentation. The performances of these two models are presented in Table 4. The step-wise data augmentation did not improve the performances of the BERT model, and the BERT model with rationales as augmented data was not better than the first-place model on the leaderboard, both with a slightly larger RMSE of 0.3. This finding indicated that the class imbalance issue in item difficulty distribution is more severe than the class imbalance across the exams in different steps.

3.5 LLMs Finetuning and In-context Learning

We fine-tuned Phi3 as our base LLM and utilized the full finetuning and LoRA finetuning methods. Although the LoRA finetuning method is typically useful for low-resource situations, the tremendous distribution gap between the LLM itself and the learning target causes the LLM to hardly learn anything. On the other hand, when utilizing full finetuning, the LLM is able to partially learn the distribution of the learning target and thus predict item difficulty in the testing dataset with a reasonable value. However, the sparsity of the training samples

Table 5: Performances for LLMs In-context Learning.

Model	RMSE
Phi3 with full finetuning	0.3816
Phi3 with Lora finetuning	0.7632
GPT4	0.3556
GPT4 (ICL)	0.3553

Table 6: Performances for LLMs Embeddings

Model	RMSE
NV-Embed-v2	0.3065
NV-Embed-v2 + GPT4 rationales	0.3023

largely affects its performance, yielding an RMSE of 0.3816, the worst among the models explored in this study except Phi3 with Lora finetuning.

The performances of GPT-4 for item difficulty prediction, with or without using ICL are presented in Table 5. The performances for GPT-4 and GPT-4 with ICL were both worse than the first place on the leaderboard with RMSE larger than 0.355. These results not only show that ICL is not effective on this task but also indicate the difficulty of this task, even the powerful GPT-4 can not yield promising performance.

3.6 LLM Embeddings

As embeddings are effective for regression tasks, we utilized the embedding model NV-Embed-v2 as the encoder and trained several additional layers for the difficulty prediction. Further, NV-Embed-v2 was enhanced by the rationales generated by GPT-4. The model performances are summarized in Table 6. The performances of these two approaches did not beat the first place in the leaderboard with RMSE larger than 0.302. Again, this indicates the difficulty of this task due to the small sample size of training data and the class imbalance issue when item difficulty is represented on a continuous scale.

4 Discussion and Conclusion

In this study, we explored different language models as well as different data augmentation methods for item difficulty modeling for large-scale standardized assessments, leveraging both SLMs and LLMs. Our results demonstrated that the application of data augmentation techniques, particularly our proposed method combining both on-the-fly data augmentation and distribution balancing data augmentation, achieved a slightly lower RMSE of 0.2926. This performance surpasses the first-place winning model in the BEA 2024 Shared Task competition leaderboard (RMSE = 0.299), indicating that our ensemble approach slightly outperforms all other reported models on the leaderboard for this dataset. This finding highlights the effectiveness of data augmentation in improving model performance and mitigating the challenges posed by data imbalance sparsity in some regions of the item difficulty scale.

Our comparative analysis of different modeling approaches revealed several key insights. Firstly, while fine-tuning SLMs such as BERT and RoBERTa yielded smaller RMSE, the introduction of domain-specific models such as BioClinicalBERT and PubMedBERT did not significantly improve model performance, likely due to distributional gaps between medical literature and test item difficulty prediction. Moreover, majority voting among multiple models provided additional robustness, further confirming the benefits of ensemble learning techniques in regression tasks.

The integration of LLMs introduced additional challenges. While models such as GPT-4 exhibited strong generalization capabilities in other NLP tasks, their performance in item difficulty prediction was limited. This outcome suggests that the scarcity of training data and the absence of explicit difficulty-related context in the input might hinder the effectiveness of LLMs in this task. Our attempts to bridge this gap using chain-of-thought prompting and rationale generation did not yield substantial improvements, likely due to insufficient training data to fully capture the key item characteristics along the item difficulty scale and ultimately exploit the advantages of LLM-based reasoning.

Note that even though the difference between the RMSE of our best-performing model, an ensemble of BERT models with two data augmentation strategies, and that of the first-place model in the competition was only 0.0064, the impact of such a difference might be meaningful for high-stakes testing programs. In large-scale standardized assessments for high-stakes decisions like the USMLE, small numerical improvements in predictive metrics such as RMSE may translate into practically meaningful impacts. More accurate item difficulty predictions may lead to improved item selection, test assembly, and better-informed decisions about examinees. Future studies may explore the impact of such slight improvement in item difficulty prediction on improvements at the overall test level.

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.

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical bert embeddings. *Preprint*, arXiv:1904.03323.

Yigal Attali, Luis Saldivia, Carol Jackson, Fred Schuppan, and Wilbur Wanamaker. 2014. Estimating item difficulty with comparative judgments. *ETS Research Report Series*, 2014(2):1–8.

Ahmad Baylari and Gh A Montazer. 2009. Design a personalized e-learning system based on item response theory and artificial neural network approach. *Expert Systems with Applications*, 36(4):8013–8021.

Isaac I Bejar. 1983. Subject matter experts' assessment of item statistics. *Applied Psychological Measurement*, 7(3):303–310.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

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

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.

James C Impara and Barbara S Plake. 1998. Teachers' ability to estimate item difficulty: A test of the assumptions in the angoff standard setting method. *Journal of Educational Measurement*, 35(1):69–81.

Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848.

Tom Kubiszyn and Gary D Borich. 2024. *Educational testing and measurement*. John Wiley & Sons.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

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.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

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.

Kelly Wauters, Piet Desmet, and Wim Van Den Noortgate. 2012. Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, 58(4):1183–1193.

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.

Victoria Yaneva, Peter Baldwin, Janet Mee, and 1 others. 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.

Operational Alignment of Confidence-Based Flagging Methods in Automated Scoring

Corey Palermo, Troy Chen & Arianto Wibowo

Abstract

In hybrid scoring systems, confidence thresholds determine which responses receive human review. This study evaluates a relative (within-batch) thresholding method against an absolute benchmark across ten items. Results show near-perfect agreement and modest distributional differences, supporting the relative method's validity as a scalable, operationally viable approach for flagging low-confidence responses.

1 Introduction

In large-scale summative assessment programs, hybrid scoring systems that combine automated engines with human raters are commonly used to balance efficiency and accuracy. To preserve human scoring resources while maintaining scoring validity, these systems often rely on a measure of model confidence to identify which responses should be routed to human reviewers. For example, in Measurement Incorporated's hybrid automated-human scoring system, each student response is first evaluated by a scoring engine that assigns both a rubric-based score and a continuous confidence value. This confidence value reflects how well the response aligns with patterns learned from previously human-scored examples. When confidence is high, the model's score is accepted; when confidence is low, the response is routed to an expert human rater for review and final score assignment. The engine's use of floating-point scores rather than discrete categories acknowledges that writing quality exists on a continuum. As such, low-confidence predictions often arise when a response falls between score points or exhibits features that are atypical relative to the model's training data.

In practice, we use a relative (within-batch) thresholding approach to determine which responses are flagged for human scoring. Because operational scoring occurs continuously over several weeks, responses are processed in discrete batches as tests are completed. The model evaluates scoring certainty within each batch and flags approximately 10% of responses reflecting the lowest confidence. This strategy enables consistent workload distribution for human raters, supports timely data delivery, and ensures manageable flow across the scoring window. By contrast, an absolute thresholding approach—which would require evaluating confidence relative to the full population of responses—poses logistical challenges, in particular delayed score reporting.

Although the relative approach offers clear operational advantages, it is not known how well it approximates the theoretical ideal of an absolute confidence threshold. The present study investigates the extent to which the relative (within-batch) thresholding approach provides a robust and valid method for identifying low-confidence responses.

The study addresses three research questions:

RQ1: To what extent do the relative and absolute methods identify the same responses as low-confidence?

RQ2: Do the responses flagged by the relative method exhibit similarly low confidence values compared to those flagged by the absolute method, in terms of their overall distribution?

RQ3: Do the responses flagged by the relative method differ in median confidence values from those flagged by the absolute method?

Through a series of statistical comparisons aligned with each research question, we examine the extent to which the relative method replicates the behavior and outcomes of an absolute thresholding approach. These analyses evaluate the overlap in flagged responses, the similarity in their confidence value distributions, and potential

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 56–60

differences in central tendency, providing a multi-faceted assessment of the relative method's robustness.

2 Methods

To address the study's research questions, we conducted a series of statistical analyses across ten items, each designed to evaluate a distinct aspect of the alignment between the relative (within-batch) thresholding method and an absolute confidence threshold.

To evaluate the extent to which the relative and absolute methods identify the same responses as low-confidence (RQ1), we conducted McNemar's tests to assess whether the proportion of discordant classifications—responses flagged by one method but not the other—differed significantly. To further quantify the degree of agreement, we calculated F1 scores and Cohen's kappa for each item.

To assess whether the relative method captures responses with similarly low confidence values as those flagged by the absolute method (RQ2), we conducted Kolmogorov–Smirnov (K-S) tests. These non-parametric tests compared the full distributions of raw confidence values for responses flagged by each method, providing a measure of distributional similarity without assuming a specific shape or variance structure.

Finally, to examine whether the responses flagged by the relative method differ in central tendency from those flagged by the absolute method (RQ3), we performed Mann–Whitney U tests. These tests specifically assessed whether there were significant differences in the median confidence values between the two groups, offering a complementary perspective to the K-S analyses focused on overall distribution.

Together, these methods provide a multi-dimensional evaluation of the relative thresholding approach's robustness and its alignment with the conceptual goals of confidence-based response flagging.

3 Results

3.1 RQ1

McNemar's tests were used to assess whether the relative and absolute methods differ in how they classify responses as low-confidence. For each item, a 2×2 contingency table was constructed, and the test evaluated whether the proportion of discordant cases—responses flagged by one method but not the other—was significantly asymmetric. As shown in Table 1, none of the McNemar tests reached statistical significance (all p-values > .95), indicating no evidence of systematic disagreement between the methods.

To complement these significance tests, F1 scores and Cohen's kappa values were computed

Item ID	Grade	N	McNemar χ^2	p-value	F1 Score (%)	Kappa
X01	8	74050	0.003	0.956	97.8	0.976
X02	4	71140	0.001	1.000	91.0	0.901
X03	7	73928	0.003	0.953	98.1	0.979
X04	8	73806	0.002	0.967	96.0	0.956
X05	5	73422	0.003	0.957	97.6	0.974
X06	3	70868	0.001	0.974	93.2	0.925
X07	6	73764	0.002	0.966	96.3	0.959
X08	6	73900	0.003	0.960	97.4	0.971
X09	5	73322	0.001	0.974	93.2	0.924
X10	7	73902	0.001	0.972	94.5	0.939

Table 1: Agreement between relative and absolute methods across items.

to quantify the degree of agreement. F1 scores ranged from 91.02% to 98.08%, reflecting a high degree of precision and recall across items. Cohen's kappa values, which adjust for chance agreement, ranged from 0.901 to 0.979, consistently exceeding the commonly cited threshold ($\kappa > 0.90$) for near-perfect agreement (Landis & Koch, 1977). Item X02 showed the lowest observed agreement (F1 = 91.02%, κ = 0.901), while item X03 showed the highest (F1 = 98.08%, κ = 0.979). These findings indicate that despite using different thresholding strategies, the relative and absolute methods align closely in practice, identifying largely overlapping subsets of responses for human scoring.

3.2 RQ2

To examine whether the confidence value distributions of flagged responses differed between the relative and absolute methods, Kolmogorov–Smirnov (K-S) tests were conducted for each of the ten items. Table 2 displays the K-S test statistics, sample sizes, and p-values for each item.

Statistically significant differences in the distributions were observed for eight of the ten items ($p < .05$), with K-S statistics ranging from 0.0236 to 0.0898. For the remaining two items (X01 and X03), the tests were not statistically

significant, indicating no detectable difference in confidence distributions between the two methods for those items.

Although statistical significance was common, the magnitude of the observed differences—as indicated by the K-S statistics—was consistently small, with all values falling below 0.10. In the context of the Kolmogorov–Smirnov test, the KS statistic represents the maximum vertical distance between the empirical cumulative distribution functions of the two samples. A value below 0.10 suggests that the most extreme divergence between the relative and absolute confidence distributions is less than 10% at any point along the confidence continuum. These values are often interpreted as indicating a negligible to small effect size, implying that while the distributions are not identical, the differences are modest and unlikely to meaningfully alter the classification of responses as low-confidence.

Item ID	Grade	N Flagged	K-S Statistic	p-value
X01	8	7552	0.022	0.052
X02	4	6796	0.090	< .001
X03	7	7605	0.019	0.121
X04	8	7360	0.040	< .001
X05	5	7430	0.024	0.032
X06	3	7106	0.068	< .001
X07	6	7309	0.037	< .001
X08	6	7484	0.026	0.012
X09	5	7084	0.068	< .001
X10	7	7449	0.055	< .001

Table 2: Kolmogorov–Smirnov test results comparing confidence distributions

To illustrate these patterns, Figure 1 displays histograms of confidence values for three representative items—one with no significant difference (X03), one with moderate divergence (X10), and one with the largest observed difference (X02). As shown, the distributions overlap substantially, with only minor shifts in the region of greatest density. These visualizations reinforce the conclusion that the relative method tends to flag responses from the same general region of the confidence distribution as the absolute method. While some divergence is detectable, the observed differences are limited and unlikely to compromise scoring validity.

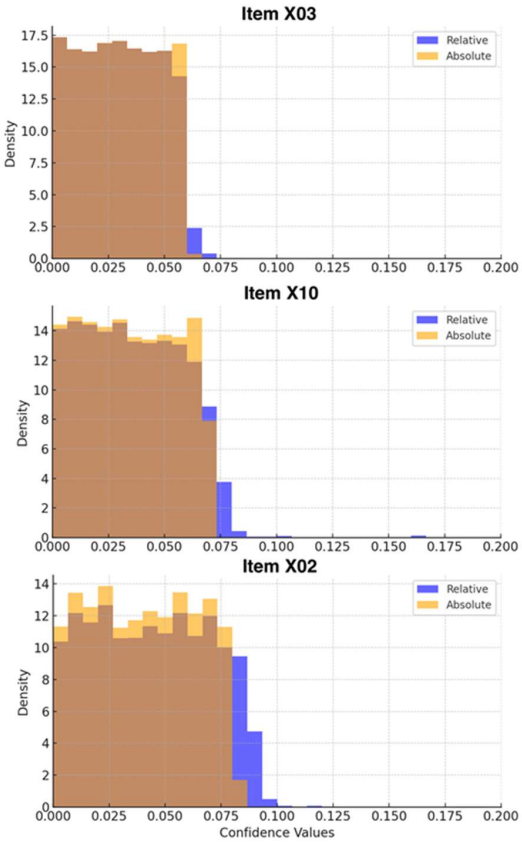

Figure 1: Histograms of confidence values: relative vs. absolute flagging.

3.3 RQ3

Table 3 presents the sample sizes, U statistics, *p*-values, and corresponding estimates of the Common Language Effect Size (Vargha & Delaney, 2000; McGraw & Wong, 1992), denoted as $\hat{P}_1$.

$\hat{P}_1$ represents the probability that a randomly selected response from one group (e.g., flagged by the absolute method) has a higher confidence value than a randomly selected response from the other group (e.g., flagged by the relative method). Under the null hypothesis of equal distributions, $\hat{P}_1 = 0.5$, indicating no systematic difference in central tendency. Values modestly above or below 0.5 suggest directional but generally small effects. Statistically significant differences in median confidence values were observed for six of the ten items, with p-values ranging from < .001 to .012. For the remaining four items (X01, X03, X05, and X08), results were not statistically significant, indicating no detectable difference in central tendency between the two groups of flagged responses.

The estimated $\hat{P}_1$ values across all items ranged from 0.501 to 0.543. These values suggest that even when differences were statistically

Item ID	Grade	N Flagged	U Statistic	p-value	$\hat{P}_1$
X01	8	7552	28700007	0.493	0.503
X02	4	6796	25070858	< .001	0.543
X03	7	7605	28975153	0.833	0.501
X04	8	7360	28008532	< .001	0.517
X05	5	7430	27915699	0.231	0.506
X06	3	7106	26815639	< .001	0.531
X07	6	7309	27639771	< .001	0.517
X08	6	7484	28466365	0.081	0.508
X09	5	7084	26357199	< .001	0.525
X10	7	7449	28402035	0.012	0.512

Table 2: Mann–Whitney U test results comparing median confidence values.

significant, the relative method only slightly increased the probability of flagging responses with higher confidence values compared to the absolute method. Importantly, all $\hat{P}_1$ values were greater than 0.5, indicating a consistent directional trend across items. This pattern aligns with the design of the relative method, which evaluates responses within batches; as a result, it may include some responses that exceed a fixed global threshold but still represent the lower-confidence tail within that batch.

Taken together, these findings reinforce the conclusion that the two methods target similar segments of the confidence distribution. While the relative method yields small, systematic shifts in central tendency compared to the absolute approach, these shifts are modest and consistent with its operational design. They do not undermine its ability to identify responses with genuinely low scoring confidence.

4 Discussion

This study evaluated the robustness of a relative (within-batch) thresholding method for identifying low-confidence responses by comparing it to an absolute thresholding approach across ten assessment items. The results indicate that although the two methods use different reference frames for determining confidence, they yield closely aligned outcomes in practice. McNemar's tests showed no statistically significant differences in flagging decisions across any item, indicating that the two methods do not systematically disagree in their classifications. Agreement metrics further reinforced this pattern, with F1 scores above 91% and Cohen's kappa values consistently exceeding 0.90—benchmarks associated with near-perfect agreement. Kolmogorov–Smirnov tests revealed statistically significant differences in the distributions of confidence values for most items, but the observed effect sizes were small, suggesting only modest divergence in how the two methods segment the confidence continuum. Mann–Whitney U tests found no significant difference in median confidence values for four items and only modest, consistently directional shifts for the others. In each case where a difference was detected, the relative method flagged responses with slightly higher confidence values than the absolute method—an expected outcome given its within-batch operational logic. These findings suggest that the relative method approximates the behavior of an absolute thresholding strategy not only in terms of response-level agreement but also in the distributional and central tendencies of flagged confidence values. The minor and systematic nature of these shifts underscores the method's practical validity, even in the absence of global thresholds.

One strength of the study is its multi-method evaluation strategy. By employing three distinct statistical tests—each aligned to a specific research question—the analysis provides a comprehensive and nuanced view of how the relative method compares to the absolute approach. This triangulation enhances the internal validity of the findings by ensuring that observed patterns are not artifacts of a single analytic lens. Prior research in assessment and machine scoring has emphasized the importance of using multiple indicators of agreement and reliability when evaluating human-machine alignment (e.g., Williamson, Xi, & Breyer, 2012). Extending this principle to thresholding methods strengthens the interpretive clarity of the current study.

A related strength is the use of operational data across ten unique items, which increases the generalizability of findings within the context of a real-world scoring system. Rather than relying on a narrow test set or simulated data, this study reflects real-world scoring dynamics, where batch effects, prompt variability, and distributional shifts routinely occur. Literature in automated writing evaluation has frequently called for validation

studies using authentic operational data (e.g., Bejar, 2011), and this study responds directly to that need.

Despite these strengths, a notable limitation is that the study treats the absolute threshold as a benchmark without fully interrogating its own validity or optimality. While the absolute method offers a theoretically attractive ideal—especially under conditions of complete data availability—it is not immune to its own biases, such as those introduced by non-uniform response distributions or scoring model calibration. A more complete validation strategy might compare both methods not only to each other but also to an external criterion, such as expert judgment of borderline cases.

Another area for future exploration involves the operational consequences of the observed differences. While McNemar's tests found no systematic disagreement in flagging decisions, statistical significance was more common in the comparisons of flagged-response distributions and central tendencies. The practical impact of these differences remains unclear. For example, do differences in flagged responses influence rater workload, response turnaround time, or score stability at the aggregate level? Future studies could simulate or analyze batch-level scoring flow under different flagging schemes to evaluate the impact of relative versus absolute methods on scoring efficiency and quality control. In this way, we might move beyond verifying that the relative method is good enough and begin to explore whether it is, in some cases, better suited to the realities of large-scale assessment.

In sum, the relative thresholding method performs robustly when compared to an absolute alternative, even though it makes decisions based only on within-batch information. It offers a stable, scalable solution that aligns well with theoretical expectations and empirical behavior of scoring confidence.

References

Bejar, I.I. (2011). A validity-based approach to quality control and assurance of automated scoring. Assessment in Education: Principles, Policy & Practice, 18(3), 319–341.

Landis, J.R., & Koch, G.G. (1977). The measurement of observer agreement for categorical data. Biometrics, 33(1), 159–174.

McGraw, K.O., & Wong, S.P. (1992). A common language effect size statistic. Psychological Bulletin, 111(2), 361–365.

Vargha, A., & Delaney, H.D. (2000). A critique and improvement of the CL common language effect size statistics of McGraw and Wong. Journal of Educational and Behavioral Statistics, 25(2), 101–132.

Williamson, D.M., Xi, X., & Breyer, F.J. (2012). A framework for evaluation and use of automated scoring. Educational Measurement: Issues and Practice, 31(1), 2–13.

Pre-Pilot Optimization of Conversation-Based Assessment Items Using Synthetic Response Data

Tyler Burleigh
Khan Academy
tylerb@khanacademy.org

Jing Chen
Khan Academy
jing@khanacademy.org

Kristen DiCerbo
Khan Academy
kristen@khanacademy.org

Abstract

Correct answers to math problems don't reveal if students understand concepts or just memorized procedures. Conversation-Based Assessment (CBA) addresses this through AI dialogue, but reliable scoring requires costly pilots and specialized expertise. Our Criteria Development Platform (CDP) enables pre-pilot optimization using synthetic data, reducing development from months to days. Testing 17 math items through 68 iterations, all achieved our reliability threshold (MCC $\geq$ 0.80) after refinement – up from 59% initially. Without refinement, 7 items would have remained below this threshold. By making reliability validation accessible, CDP empowers educators to develop assessments meeting automated scoring standards.

1 Background

When students solve math problems correctly, teachers face a critical challenge: they cannot tell if students understand the concepts or just memorized the steps. A student who correctly solves $1.5(2 - 4h) = 6h$ might understand why division maintains equality, or might simply execute a memorized procedure. When students do not solve a problem correctly, the only information available is that they entered an incorrect answer. It is unknowable whether they had a partial or incomplete understanding of the problem. Traditional tests cannot provide evidence about students' thought process when answering questions, creating a gap that affects teaching decisions and student support.

Conversation-Based Assessment (CBA) enables the assessment of conceptual understanding through adaptive dialogue (Yildirim-Erbasli and Bulut, 2023). In CBA, students explain their reasoning, similar to constructed response items (Williamson et al., 2012). Unlike static written responses, CBA adapts based on student answers – asking follow-ups when needed and providing appropriate feedback (Jackson et al., 2018). This interaction provides evidence indicating whether students grasp underlying concepts.

CBA technology has progressed from scripted to generative systems. Early approaches required authoring specific response-reply pairs (Zapata-Rivera et al., 2015), essentially building complete dialogue trees that anticipated every possible student response. Later systems like Quizbot used semantic matching to map student responses to pre-written feedback (Ruan et al., 2019), but educators still had to design and construct all potential conversation paths beforehand.

Large Language Models (LLMs) introduced in 2022 (OpenAI, 2024) marked a paradigm shift. Instead of pre-building dialogue trees, these newer systems allow students to respond openly in their own words, with the AI using NLP methods to understand and categorize responses dynamically (Bergerhoff et al., 2024). This eliminates the burden of anticipating and scripting every possible conversation branch, making CBA development accessible to educators without the resources for complex dialogue engineering.

Yet this freedom from pre-coding dialogue paths creates a different challenge. When systems can accept any student response rather than matching against predetermined patterns, they must interpret novel expressions of understanding in real-time. While these models are capable of such evaluation, without explicit guidance about what constitutes conceptual mastery, their scoring decisions may lack the consistency needed for reliable assessment.

Scoring criteria provide this needed guidance, giving structure to open-ended evaluation. By explicitly defining what constitutes conceptual mastery for each item, these criteria enable CBA systems to evaluate diverse student responses consistently and generate appropriate follow-ups. Good criteria help AI Scorers match human grader reliability (Henkel et al., 2024), especially when subject

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 61–68

matter experts write item-specific criteria rather than generic prompts (Frohn et al., 2025).

Creating reliable scoring criteria requires meeting established assessment standards, with AI and human graders reaching similar conclusions. Educational assessment typically requires strong agreement (e.g., $\kappa \geq 0.70$) (Williamson et al., 2012; Wood et al., 2021). These thresholds challenge traditional empirical validation because they require extensive time and resources to reach (Williamson et al., 2012). Developers draft criteria, pilot them with real students, and compare AI scores to human ratings. Discrepancies trigger revision and re-piloting. Most items require multiple cycles, taking months and requiring fresh student data each time.

Even when time and resources are available, the validation process requires specialized technical knowledge that content authors often lack, such as: dataset development (developing and labeling balanced, diverse synthetic datasets), metric computation (choosing and calculating coefficients), iteration management (managing multiple criteria refinements and their associated datasets), and interpreting results (setting targets and identifying which changes were meaningful). Without this expertise, efforts may yield unreliable results.

These twin challenges – lengthy validation cycles and specialized expertise requirements – create a bottleneck in CBA development. Without tools to test criteria before student pilots, developers must choose between deploying potentially unreliable assessments or investing months in iterative pilot studies. At Khan Academy, these challenges drove us to develop an alternative to time-consuming student pilots for validating scoring criteria. To solve this problem, we developed a platform that lets creators test criteria using synthetic data and provides step-by-step guidance.

1.1 Explain Your Thinking (EYT): A Modern Conversation-Based Assessment System

Before describing our solution, we first describe the EYT system itself. Understanding how EYT uses criteria to both score responses and generate follow-up questions reinforces why criteria quality is so critical to CBA success.

Explain Your Thinking (EYT) is our implementation of modern CBA. Students first solve problems, then explain their reasoning in AI-guided conversations. For example, when a student solves $1.5(2 - 4h) = 6h$ by dividing both sides by 1.5,

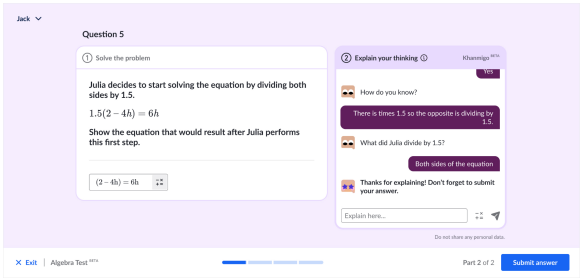

Figure 1: Screenshot of the Explain Your Thinking conversation-based assessment item type. The student first answers a math problem (left), and then has a conversation about the problem (right) which is designed to assess their conceptual understanding.

we know whether or not they can execute the procedure. But EYT goes deeper: can they explain why division maintains equality? Do they understand that division and multiplication are inverse operations? The system uses scoring criteria to evaluate these conceptual understandings and generate appropriate follow-up questions.

Each assessment activity starts with a math problem, the student's answer, and criteria defining complete understanding. The platform operates through three integrated functions that enable assessment. First, it recognizes varied expressions of concepts, allowing students to explain their thinking in their own words. Second, it generates probing questions that explore understanding without revealing answers. Third, it maintains assessment validity by avoiding teaching during the evaluation process.

Importantly, EYT's effectiveness depends on a criteria-driven cascade. At each turn, an AI Scorer evaluates the conversation history to determine which criteria the student has satisfied. These evaluations then flow to a Response Generator, which receives a list of unsatisfied criteria and generates targeted follow-up questions to probe those specific gaps. When criteria are vague or missing, this cascade breaks down: the AI Scorer misclassifies responses, passing incorrect information downstream, and the Response Generator asks about the wrong concepts, leading to unproductive conversations.

Students experience a natural conversation flow. They explain their approach and receive targeted follow-ups that probe gaps without teaching. The conversation continues until students demonstrate understanding or reach a four-turn limit.

2 Criteria Development Platform (CDP)

Given that poor criteria can derail EYT's assessment cascade and compromise validity, we needed a way to ensure criteria quality before deployment. Our Criteria Development Platform (CDP) addresses this need by enabling content creators to test and refine AI scoring criteria using synthetic student responses, eliminating the months-long pilot cycles traditionally required for validation.

CDP operates through an iterative workflow where creators write scoring criteria, generate synthetic responses that test edge cases, evaluate AI Scorer performance against these responses, and refine their criteria based on the results.

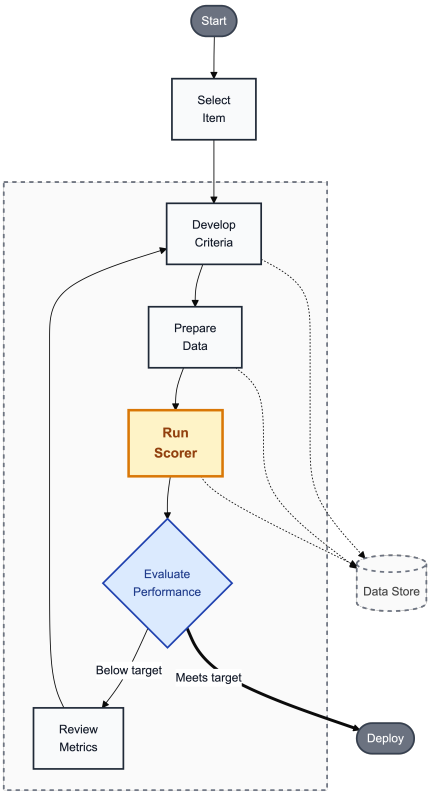

Figure 2: The Criteria Development Platform's iterative workflow. Content creators write scoring criteria, generate synthetic responses, test performance, and refine their criteria based on results.

CDP addresses both core challenges of criteria development. First, it reduces validation time from months to days by eliminating the need for multiple rounds of student pilots. Creators can test dozens of iterations in hours or days rather than weeks or months. Second, it provides guided support that makes reliable assessment creation accessible without specialized expertise. The platform automatically tracks versions, computes performance metrics, and provides targets and actionable feedback to guide criteria development.

To evaluate CDP's effectiveness, we analyzed 68 development cycles from six content creators developing 17 math assessment items. Our analysis addresses three key research questions:

1. **Engagement**: Do content creators effectively engage in iterative refinement when using CDP?
2. **Improvement**: When creators iterate, do their scoring criteria demonstrate measurable performance gains?
3. **Achievement**: What proportion of items ultimately meet established reliability standards?

The following sections detail CDP's design and demonstrate its effectiveness through empirical analysis of these development cycles.

2.1 How CDP works

Creators follow four steps (Figure 2) to create scoring criteria. They select an item and then iterate through: writing criteria, generating data, and testing performance until meeting standards. Throughout this cycle, the tool preserves all data and metrics, allowing creators to track improvements and learn from each iteration. This four-step process addresses the challenges of time and expertise: synthetic data allows rapid iteration, while metric generation and feedback provide scaffolding for creators.

2.1.1 Step 1: Selecting the item

Content creators start by selecting an item for which to develop criteria.

2.1.2 Step 2: Writing scoring criteria

Next, creators write up to seven criteria that define a complete response. For instance, an item about solving equations might include criteria like "identifies the inverse operation needed" and "explains why division undoes multiplication." The platform tracks all versions of criteria, allowing creators to try different approaches and revert to previous versions as needed.

2.1.3 Step 3: Generating test data

Creators need synthetic data to evaluate their criteria without real students. The platform guides creators in developing balanced datasets of 150 simulated responses per test, with 50 responses each from correct, partially correct, and incorrect

categories. Each response includes the student's initial answer, their conversational explanation, and a human-assigned ground truth label (correct, partially correct, or incorrect) indicating the response's category. Content creators must carefully assign these ground truth labels when developing the synthetic dataset, as they serve as the authoritative reference for evaluating the AI Scorer's performance. This balanced distribution across the three categories ensures comprehensive testing of the criteria. The sample size of 150 was determined through simulation-based power analysis, achieving >80% Bayesian posterior probability that MCC $\geq$ 0.8 when the true MCC is at least 0.84. This provides strong statistical evidence for identifying scorers that meet the performance threshold.

Creators generate these responses through a combination of manual writing and AI assistance. To ensure quality and authenticity, we instructed creators to manually write at least 10-15 example responses for each correctness category, capturing realistic student thinking patterns. (Note that for scoring purposes, the AI Scorer uses a binary classification approach and treats partially correct responses as incorrect. However, including partially correct responses in the dataset serves a critical purpose: they enhance diversity by capturing edge cases and boundary conditions where students demonstrate some but not all required understanding. This helps creators test whether their criteria can distinguish between complete and incomplete responses, identifying potential ambiguities before deployment.)

When using the AI generator, the platform prompts a language model such as GPT-4.1 with these manually-created examples, information about the item, and the criteria. The model generates unique, plausible student responses matching the specified category. It receives instructions to vary both reasoning patterns and writing style. This ensures responses remain meaningfully different from the provided examples. Creators must verify all AI-generated responses and correct ground truth labels if necessary before adding them to their dataset.

This approach combines human expertise with AI's ability to generate variations at scale. Human creators identify realistic student thinking patterns. AI generates diverse examples based on these patterns. Human oversight ensures classification accuracy throughout. We also encouraged creators to include edge cases in test data, such as correct reasoning with unusual terminology. Testing against these challenging cases helps creators identify and fix ambiguities in their criteria before real students encounter them.

The system helps ensure response diversity through similarity checking. When generating synthetic responses, the platform uses semantic embeddings to compare each new response against all other responses within the same correctness category, flagging pairs that exceed 85% similarity. For mathematical problems, the similarity checker includes additional detection for responses that differ only in numerical values, as these may have high semantic similarity despite representing fundamentally different solutions. This prevents dataset contamination from near-duplicate responses that would artificially inflate performance metrics.

2.1.4 Step 4: Testing and refining

With criteria and test data ready, creators run the AI Scorer and see how well it performs with the current criteria. The platform calculates performance metrics by comparing the AI Scorer's predictions against the human-assigned ground truth labels from the synthetic dataset. Metrics like accuracy and false positive rates reveal how well the AI Scorer aligns with human judgment, helping creators identify problems with their criteria definitions. Additionally, the tool provides the AI Scorer's reasoning for each evaluation, showing where criteria might be unclear or ambiguous (Figure 3). This transparency addresses the expertise challenge by making the AI Scorer's evaluation process interpretable to non-experts.

3 Research

To evaluate CDP's effectiveness in enabling pre-pilot optimization, we conducted an empirical analysis of platform usage data. This analysis directly addresses the three research questions posed in our Aims: examining creator engagement patterns, measuring performance improvements, and determining achievement rates.

3.1 Methods

3.1.1 Dataset

Six content creators (curriculum specialists and assessment designers) independently developed 17 mathematics items over three months. Each development cycle followed the same workflow: writing scoring criteria, generating synthetic responses, and evaluating AI Scorer performance. Items covered grades 6-12 mathematics, addressing algebra,

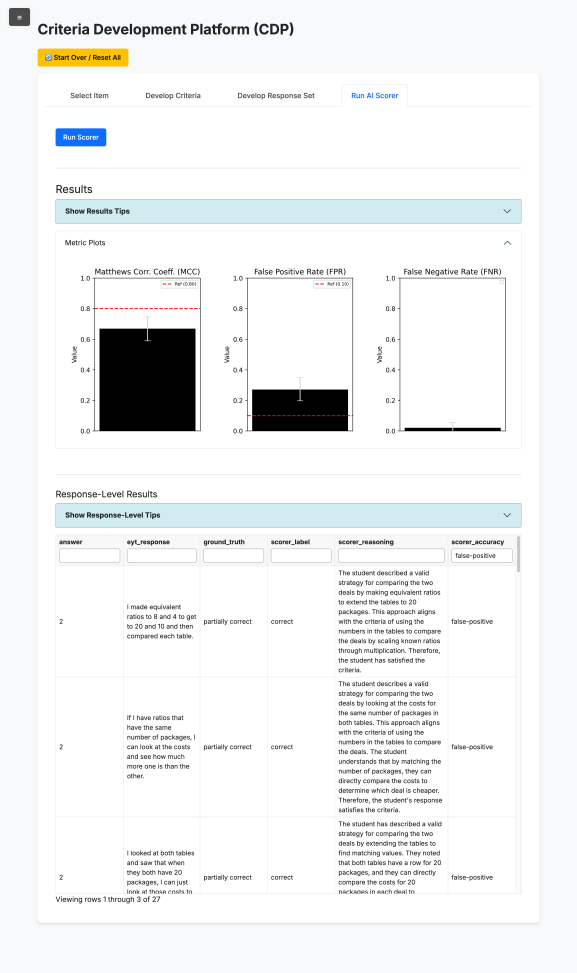

Figure 3: The Criteria Development Platform's AI Scorer interface displaying performance metrics (MCC, FPR, FNR) and response-level results with the AI's scoring reasoning for each evaluation.

geometry, and ratio topics aligned with Common Core State Standards.They generated 68 development cycles. When content creators tested the same criteria version multiple times during development, we included only the final run for each version in our analysis. This resulted in 61 distinct criteria versions with 10,200 synthetic response evaluations.

We analyzed two distinct item groups. Eight items (47%) underwent iterative refinement through multiple criteria revisions. Nine items (53%) achieved strong performance without criteria changes, maintaining consistent criteria across runs. This division lets us examine both the refinement process and cases of immediate success.

3.1.2 Performance Metrics

We evaluated AI Scorer performance using four metrics that measure agreement between AI-generated scores and ground truth labels (expert human scoring):

Matthew's Correlation Coefficient (MCC) serves as our primary metric for evaluating scoring reliability, considering all classification outcomes. Values range from -1 to +1, with ≥ 0.80 threshold. MCC balances imbalanced datasets.

Three additional metrics provide comprehensive evaluation alongside our primary MCC metric:

Cohen's kappa (κ_C) quantifies agreement beyond what random chance would produce, with ≥ 0.81 indicating substantial reliability (per Landis and Koch (1977)).

False Positive Rate (FPR) tracks a critical failure mode: marking incorrect responses as correct, which terminate conversations prematurely. Our threshold of ≤ 0.10 ensures the AI does not often terminate conversations prematurely.

Accuracy: proportion correct.

3.1.3 Statistical Analysis

We compared first versus last criteria versions across refined items (n=8) using bootstrap methods with 10,000 iterations. Bootstrap provides robust p-values without requiring distributional assumptions that may not hold for our metrics. For non-refined items (n=9), we report performance metrics for the single iteration only. We used one-tailed tests for all metrics: expecting increases for MCC, κ_C, and accuracy, and expecting a decrease for FPR.

3.1.4 Results

RQ1: Creator Engagement in Iterative Refinement Creators used iteration effectively. Items underwent a median of 3 versions (mean 5.2 versions per item), with 58.8% of items being revised at least once (10 of 17 items). Among all 17 items, 8 (47%) had meaningful criteria version changes that we analyze as "refined items," while 9 (53%) maintained consistent criteria across runs ("non-refined items"). This iteration pattern suggests creators found a productive balance between refinement and effort: enough iteration to improve performance without excessive revision cycles.

The platform enabled rapid development. Items were developed over a median of 1 day (range: 1-4 days), addressing the time bottleneck.

Criteria became more detailed through iteration, with a median increase of 5 words from first to last version (60 to 65 words, representing an 8.3% increase). The number of individual criteria also increased modestly from an average of 1.9 to 2.2.

RQ2: Performance Improvements Through Iteration For the 8 items that underwent criteria refinement (47% of the dataset), comparing first versus last criteria versions revealed statistically significant improvements across key performance metrics (Table 1). Our primary metric, MCC, improved from 0.659 to 0.863 ($p < 0.001$), representing a +0.203 improvement in scoring reliability. This improvement means refined items moved from moderate to strong reliability. κ_C also improved significantly, from 0.620 to 0.860 ($p < 0.001$), a gain of +0.240. According to Landis and Koch (1977), this represents improvement from substantial agreement (0.61-0.80) to almost perfect agreement (≥ 0.81).

False positive rates decreased from 0.148 to 0.087 (-0.060, $p = 0.163$). While not statistically significant in aggregate, individual items showed varied patterns. Some items achieved large FPR reductions (one item improved by 0.420). Others experienced FPR increases while creators prioritized our primary MCC metric (another item's FPR increased from 0.020 to 0.140 while its MCC improved by 0.323). Overall accuracy improved significantly from 0.818 to 0.938 ($p < 0.001$), representing a +0.119 improvement. Notably, 100% of refined items showed improvements in both MCC and κ_C, demonstrating that the iterative refinement process consistently led to better scoring reliability.

For example, item A-CED.A.3 asks students to interpret inequality solutions in real-world contexts. Through iteration, creators refined the criteria for greater precision. The refined criteria specified that students must **explicitly** state why a whole number is needed for the real-world scenario and **explicitly** explain why rounding down is necessary to satisfy the inequality. These refinements, which instructed the AI Scorer not to accept implied reasoning, improved the AI Scorer's MCC from 0.554 to 0.971 (Figure 4; see Appendix A for complete criteria text).

RQ3: Achievement of Reliability Standards With iterative refinement, 100% of items achieved our primary reliability standard (MCC $\geq$ 0.80), compared to only 58.8% based on first-attempt performance. This 100% success rate shows how CDP's guided refinement process makes reliable assessment development accessible to creators regardless of their psychometric expertise. This improvement suggests that CDP rescued 7 items that would have required abandonment or costly pilot-based revision. When considering both MCC and our secondary FPR threshold ($\leq$ 0.10), 76% of items (13 of 17) met both standards after CDP refinement.

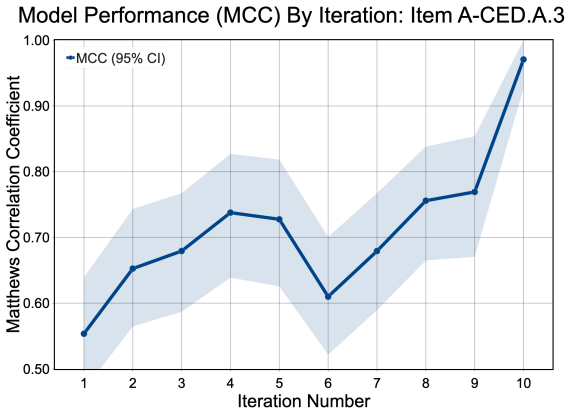

Figure 4: Item A-CED.A.3 scoring reliability across 10 iterations. Matthews Correlation Coefficient improved from 0.554 to 0.971 (95% confidence intervals shown), demonstrating how iterative refinement strengthened the scoring criteria.

The items demonstrated two development patterns. Nine items (53%) achieved strong performance immediately, meeting the MCC threshold of 0.80 without criteria changes. These items maintained consistent criteria across all runs. The remaining 8 items (47%) underwent iterative refinement. Of these refined items, only 1 (12.5%) initially met the MCC threshold but was still refined (possibly to improve other metrics or address creator concerns). Through CDP's iterative process, all 8 refined items achieved MCC $\geq$ 0.80, with a final mean MCC of 0.863.

All 8 refined items achieved the MCC threshold, but FPR outcomes varied. Five of 8 items (62.5%) met the FPR $\leq$ 0.10 threshold after refinement. This reflects the challenge of optimizing multiple metrics simultaneously. Creators sometimes prioritize specific metrics based on their assessment goals.

These results validate CDP's solution to the twin challenges of time and expertise: all items achieved reliability standards (expertise) within days rather than months (time).

4 Limitations

Three limitations shape the interpretation of these results.

First, synthetic responses cannot capture all the ways real students think. Students use unexpected

Table 1: First vs. Last Criteria Version Performance Comparison for Refined Items (n=8)

Metric	First Run [95% CI]	Last Run [95% CI]	Change	p-value	Items Improved
MCC	0.659 [0.589-0.734]	0.863 [0.833-0.900]	+0.203	<0.001***	8 (100%)
κ_C	0.620 [0.533-0.711]	0.860 [0.830-0.898]	+0.240	<0.001***	8 (100%)
FPR	0.148 [0.033-0.288]	0.087 [0.045-0.130]	-0.060	0.163	3 (37.5%)
Accuracy	0.818 [0.769-0.867]	0.938 [0.924-0.954]	+0.119	<0.001***	8 (100%)

Note: *** $p < 0.001$; Bootstrap tests with 10,000 iterations, one-tailed

terminology, creative analogies, and unique error patterns that synthetic generation misses. Future work must validate with real student data.

Second, we validated CDP with mathematics items and GPT-4o. While the approach should generalize to other domains using the same EYT format, criteria optimized for GPT-4o's scoring tendencies might not transfer directly to other LLMs.

Third, CDP optimizes scoring reliability but doesn't evaluate conversation quality. Criteria both evaluate and trigger follow-ups. We measured scoring, not dialogue quality. Future work should examine whether improvements in scoring reliability correlate with better conversation flow and more effective probing of student understanding.

These limitations point to clear next steps: validating with real student data, testing beyond math and GPT-4o, and measuring conversation quality.

5 Conclusions

In order to create effective conversation-based assessments, we need effective criteria for scoring them. These criteria are traditionally difficult and time-consuming to develop. The Criteria Development Platform addresses this challenge through pre-pilot optimization with synthetic data. Our analysis of 68 development cycles across 17 mathematics items demonstrates CDP's impact: success rates improved from 59% to 100%, rescuing 7 items from abandonment or costly pilot revision. The eight items that underwent refinement showed substantial gains, with MCC improving from 0.659 to 0.863. CDP solves both traditional CBA development challenges: reducing timelines from months to days (median 1 day) while enabling non-technical experts to achieve reliable results through guided refinement.

These results have broader implications for educational technology. Pre-pilot optimization with synthetic data provides an effective approach when authentic data is expensive or unavailable. The platform's transparency shows creators exactly why scoring succeeds or fails, transforming development from intuition to evidence. By making reliable assessment development accessible to educators without specialized expertise, tools like CDP enable more practitioners to create LLM-based assessments that measure deep understanding.

6 Appendix A: Example Criteria Changes

This appendix documents the criteria refinement process for Item A-CED.A.3, which improved from MCC = 0.554 to 0.971.

Students must explain two things: why decimal solutions need whole number rounding, and why rounding down (not up) satisfies the constraint.

6.1 First Criteria Version (MCC = 0.554)

The initial criteria were:

- **Criterion 1:** Student recognizes that the answer has to be a whole number of rides in order to make sense in the real world.
- **Criterion 2:** Student acknowledges that rounding the decimal answer down to the lower whole number is necessary because rounding up to the higher whole number makes the inequality that defines the number of credits no longer true.

6.2 Final Criteria Version (MCC = 0.971)

Testing revealed the AI accepted implied reasoning when explicit statements were needed. Revised:

- **Criterion 1:** Student must explicitly state reasoning for rounding to a whole number that includes making sense in the real-world (for example, "it does not make real-world sense for a quantity of rides to be a fraction or decimal"). It is not correct for a student to imply reasoning or to only say that they rounded down.

- **Criterion 2:** Student acknowledges that rounding the decimal answer down to the lower whole number is necessary to satisfy the inequality. Student must explicitly refer to the inequality or explain why they round down in the context of the problem (example: the most number of rides without going over in credits). It is not correct for a student to imply reasoning.

References

Jan Bergerhoff, Johannes Bendler, Stefan Stefanov, Enrico Cavinato, Leonard Esser, Tommy Tran, and Aki Härmä. 2024. Automatic conversational assessment using large language model technology. In *Proceedings of the 2024 16th International Conference on Education Technology and Computers*, pages 39–45, Porto Vlaams-Brabant Portugal. ACM.

Scott Frohn, Tyler Burleigh, and Jing Chen. 2025. Automated Scoring of Short Answer Questions with Large Language Models: Impacts of Model, Item, and Rubric Design. In *Artificial Intelligence in Education*, volume VI of *Lecture Notes in Artificial Intelligence*, pages 44–51, Palermo, Italy. Springer.

Owen Henkel, Libby Hills, Adam Boxer, Bill Roberts, and Zach Levonian. 2024. Can Large Language Models Make the Grade? An Empirical Study Evaluating LLMs Ability To Mark Short Answer Questions in K-12 Education. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, pages 300–304, Atlanta GA USA. ACM.

G. Tanner Jackson, Katherine E. Castellano, Debra Brockway, and Blair Lehman. 2018. Improving the Measurement of Cognitive Skills Through Automated Conversations. *Journal of Research on Technology in Education*, 50(3):226–240.

J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159.

OpenAI. 2024. Introducing ChatGPT.

Sherry Ruan, Liwei Jiang, Justin Xu, Bryce Joe-Kun Tham, Zhengneng Qiu, Yeshuang Zhu, Elizabeth L. Murnane, Emma Brunskill, and James A. Landay. 2019. QuizBot: A Dialogue-based Adaptive Learning System for Factual Knowledge. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, Glasgow Scotland Uk. ACM.

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.

Scott Wood, Erin Yao, Lisa Haisfield, and Susan Lottridge. 2021. Establishing Standards of Best Practice in Automated Scoring. Technical report, ACT, Inc. Publication Title: ACT, Inc. ERIC Number: ED616491.

Seyma N. Yildirim-Erbasli and Okan Bulut. 2023. Conversation-based assessment: A novel approach to boosting test-taking effort in digital formative assessment. *Computers and Education: Artificial Intelligence*, 4:100135.

Diego Zapata-Rivera, Tanner Jackson, and Irvin R. Katz. 2015. Authoring Conversation-based Assessment Scenarios. In Robert A. Sottilare, Arthur C. Graesser, Xiangen Hu, and Keith Brawner, editors, *Design Recommendations for Intelligent Tutoring Systems, Volume 3: Authoring Tools & Expert Modeling Techniques*, pages 169–178. U.S. Army Research Laboratory, Orlando.

When Humans Can't Agree, Neither Can Machines: The Promise and Pitfalls of LLMs for Formative Literacy Assessment

Owen Henkel
University of Oxford
owen.henkel@education.ox.ac.uk

Kirk Vanacore
Cornell University
kpv27@cornell.edu

Bill Roberts
Legible Labs
bill@legiblelabs.com

Abstract

Story retell assessments provide valuable insights into reading comprehension but face implementation barriers due to time-intensive administration and scoring. This study examines whether Large Language Models (LLMs) can reliably replicate human judgment in grading story retells. Using a novel dataset we conduct three complementary studies examining LLM performance across different rubric systems, agreement patterns, and reasoning alignment. We find that LLMs (a) achieve near-human reliability with appropriate rubric design, (b) perform well on easy-to-grade cases but poorly on ambiguous ones, (c) produce explanations for their grades that are plausible for straightforward cases but unreliable for complex ones, and (d) different LLMs display consistent "grading personalities" (systematically scoring harder or easier across all student responses). These findings support hybrid assessment architectures where AI handles routine scoring, enabling more frequent formative assessment while directing teacher expertise toward students requiring nuanced support.

1 Introduction

Story retell tasks offer unique advantages for assessing reading comprehension, requiring students to actively reconstruct understanding rather than simply recognize correct answers. Despite their pedagogical value, implementation faces significant barriers: administering, transcribing, and scoring individual responses is time-intensive, particularly for teachers managing large classes in resource-constrained environments.

Recent advances in Large Language Models (LLMs) present opportunities to address these barriers while maintaining assessment quality. This study examines whether LLMs can reliably replicate human judgment in grading story retells and, critically, whether they can identify cases requiring human attention. Such capabilities could enable hybrid assessment systems that automate routine scoring while preserving teacher expertise for complex decisions.

We address three interconnected questions: (1) To what extent can LLMs replicate human judgments about story retell quality across different rubric systems? (2) What patterns emerge in model-human agreement, particularly for cases humans find ambiguous? (3) How well do LLM explanations correspond with human reasoning?

Using 95 student story retells from Ghana, we examine these questions through three complementary studies. Study 1 establishes baseline performance across rubric types. Study 2 investigates agreement patterns and model "grading personalities." Study 3 explores the relationship between explanation quality and scoring accuracy.

Our findings have direct implications for educators considering AI-supported assessment tools, providing evidence for how such technologies might enhance rather than replace human judgment in literacy assessment, particularly in contexts where frequent formative assessment is essential but difficult to implement.

2 Prior Work

2.1 Story Retell as Reading Comprehension Assessment

Story retelling provides a unique window into reading comprehension by requiring students to actively reconstruct narratives rather than simply recognize correct answers. The cognitive demands of retelling—drawing upon memory for factual details, generating inferences to fill gaps, and reconstructing events in sequence—mirror authentic comprehension processes (Reed and Vaughn, 2012; Wilson et al., 1985). This active recall requirement distinguishes retelling from recognition-based assessments, potentially providing richer insights into student understanding.

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 69–78

Research on retell effectiveness has yielded mixed but generally positive findings. Reed and Vaughn (2012)'s review of 54 studies found moderate correlations between retell scores and standardized comprehension measures across grade levels. However, some studies report inconsistent relationships with reading abilities (Hagtvet, 2003; Marcotte and Hintze, 2009), suggesting retelling may capture distinct aspects of comprehension not fully reflected in traditional assessments.

2.2 Scoring Approaches and Challenges

Multiple scoring methods exist, each with inherent trade-offs. Idea-unit analysis divides passages into weighted narrative elements, enabling granular assessment but requiring text-specific development that limits cross-story comparison (Maria, 1990). Component-based scoring evaluates narrative elements (characters, setting, plot) more generalizably but faces reliability challenges, with researchers observing inconsistent inter-rater agreement (Klesius and Homan, 1985).

Holistic scoring assigns overall quality ratings, balancing efficiency with detail but introducing subjectivity that can compromise reliability without careful rubric design and rater calibration. Word-count measures offer objectivity and ease of automation but may reward verbosity over comprehension quality, as critics note students could manipulate metrics without demonstrating understanding (Altwerger et al., 2007; Goodman, 2006).

2.3 Formative Assessment in Literacy Contexts

Black and Wiliam (1998)'s seminal meta-analysis established formative assessment as one of education's most powerful interventions, demonstrating effect sizes between 0.4 and 0.7 standard deviations. Building on Sadler (1989)'s framework—understanding quality standards, comparing work against standards, and possessing gap-closing strategies—formative assessment becomes "formative" when evidence actively adapts instruction to meet student needs.

In literacy contexts, formative assessment plays a critical role in comprehension development. The comprehensive nature of reading assessment demands substantial time investment that conflicts with classroom constraints, particularly given increasing student-teacher ratios.

Story retelling emerges as a particularly powerful formative tool, demonstrating moderate correlations with other comprehension measures while showing stronger relations to authentic literacy tasks than traditional assessments. The interactive nature provides diagnostic capabilities revealing thinking strategies inaccessible through traditional measures.

2.4 Large Language Models and Educational Assessment

Recent advances in Large Language Models present distinctive capabilities for assessment support. Unlike rigid scoring systems, LLMs demonstrate capacity to generalize to new tasks with minimal examples, completing assessments through prompt modification rather than retraining (Ouyang et al., 2022). However, Schneider et al. (2023) caution that readiness for independent grading remains uncertain given the complexity of human narrative interpretation. This study examines whether these capabilities can be systematically applied to story retell assessment within educational contexts.

3 Dataset and Methods

3.1 Dataset Description

The ROARS dataset comprises responses from 130 Ghanaian adolescent students who read one of two 400-word fictional stories and completed comprehension tasks including story retell. Of these, 95 students completed the retell task, with remaining responses left blank. All retells were transcribed verbatim and word counts recorded. This dataset provides a diverse context for examining AI-assisted assessment capabilities in a Global South educational setting.

3.2 Human Rating Process and Rubric Development

Three human raters evaluated the 95 story retells using three distinct rubric systems adapted from literature. All raters held Master of Education degrees and had classroom teaching experience, providing professional expertise in literacy assessment. Ground truth scores were determined by averaging ratings and rounding to the nearest whole number. This averaging process itself highlights inherent assessment ambiguity—unanimous agreement occurred in only 66% of cases.

The adapted rubrics are presented in Appendix **??**.

Examples of different rater's scoring by rubric			
Retell 1			
lucy was a girl who like learning around she round through the country stole that night allways when everyone including the sheeps and lambs were as sleep. lucy helped to save the shepherd when the shepherd got a broke in his leg.			
	2-class	3-class	5-class
Rater 1	0	1	1
Rater 2	0	1	2
Rater 3	0	1	2
Ground Truth	0	1	2
Retell 2			
Lucy was different from all the other sheep right from the start. One day something terrible happened. The shepherd fell over and broke his leg...			
	2-class	3-class	5-class
Rater 1	1	2	4
Rater 2	1	2	3
Rater 3	1	1	4
Ground Truth	1	2	4

Table 1: Examples of different rater's scoring by rubric

3.3 LLM Assessment Methodology

For the automated grading component, we used GPT-4 (GPT-4o-2024-05-13) with carefully designed prompts that replicated the human rating context. The prompts instructed the model to act as a literacy teacher evaluating reading comprehension through story retell assessment. Each prompt included: the complete rubric with detailed scoring criteria, the original story text for context, instructions to provide only the numeric score output, and role-based framing to establish appropriate assessment perspective.

4 Study 1: LLM Replication of Human Judgments

4.1 Inter-Rater Agreement Analysis

Before examining LLM performance, we first established baseline human inter-rater agreement to understand the inherent reliability of the assessment task. Analysis of average ratings revealed systematic differences between raters. Rater 3 consistently awarded higher scores than Raters 1 and 2 across rubrics, suggesting more lenient grading standards. For the two-class rubric, average scores were 0.17, 0.21, and 0.37 respectively (scale 0-1). Similar patterns emerged for three-class (0.37, 0.36, 0.59; scale 0-2) and five-class rubrics (1.41, 1.14, 1.22; scale 0-4).

Inter-rater reliability varied substantially across rubric types. For the binary rubric, Fleiss' kappa was 0.56, indicating moderate agreement. Agreement improved markedly for the three-class rubric (Kendall's W = 0.81) and further for the five-class rubric (Kendall's W = 0.85). As validation, pairwise Cohen's kappa averages showed the same progression: 0.60 (two-class), 0.74 (three-class), and 0.78 (five-class).

This pattern suggests that more granular rubrics enable higher inter-rater reliability, possibly because they provide clearer distinctions between performance levels. The binary forced choice between "bad" and "good" may inadequately capture the complexity of student responses, leading to inconsistent judgments when responses fall near the decision boundary.

Prediction	Prec.	Rec.	F1	Supp.
Bad Retell (0)	0.86	1.00	0.93	74
Good Retell (1)	1.00	0.25	0.40	16
Average	0.89	0.87	0.83	90

Table 2: Two-class rubric performance (LWK/QWK = 0.35)

4.2 LLM Performance

We evaluated GPT-4's ability to score student retells using the same rubrics as human raters. The model received simple prompts explaining the task, relevant rubric, original story, and student response, then provided numeric scores. This straightforward approach established baseline capabilities without sophisticated prompt engineering.

4.2.1 Two-Class Rubric Results

The model's performance on the binary rubric revealed significant challenges, as shown in Table 2.

The model showed bias toward the "Bad Retell" category, correctly identifying all poor responses but capturing only 25% of good retellings. This conservative grading produced perfect precision for "Good Retell" (no false positives) but poor recall. The LWK of 0.35 indicates low agreement with human consensus, substantially below the human inter-rater agreement of 0.56.

4.2.2 Three-Class Rubric Results

Performance improved dramatically with the three-class rubric, as shown in Table 3.

The model excelled at identifying bad retellings (94% recall, 95% precision) and showed improved recognition of mediocre responses (81% recall). However, it remained conservative with "Good

Prediction	Prec.	Rec.	F1	Supp.
Bad Retell (0)	0.95	0.94	0.94	64
Mediocre (1)	0.57	0.81	0.67	16
Good Retell (2)	1.00	0.25	0.40	8
Average	0.89	0.85	0.84	88

Table 3: Three-class rubric performance (QWK = 0.78)

Prediction	Prec.	Rec.	F1	Supp.
Bad (0)	0.68	0.91	0.78	23
Poor (1)	0.77	0.68	0.72	34
Mediocre (2)	0.65	0.58	0.61	19
Acceptable (3)	0.50	0.38	0.43	7
Good (4)	1.00	1.00	1.00	1
Average	0.69	0.69	0.68	84

Table 4: Five-class rubric performance (QWK = 0.82)

Retell" classifications, capturing only 25% despite perfect precision. The QWK of 0.78 approaches the human inter-rater agreement of 0.81, suggesting near-human reliability.

4.2.3 Five-Class Rubric Results

The five-class rubric yielded the highest agreement levels, as shown in Table 4.

While individual category performance varied, the QWK of 0.82 nearly matches human agreement (0.85). The model showed strongest performance at the extremes—identifying clearly bad (91% recall) and the single good retelling (100% recall)—with more uncertainty in middle categories. This pattern mirrors human rating behavior, where edge cases between adjacent categories prove most challenging.

4.3 Comparative Analysis

The progression of model-human agreement across rubrics closely parallels the pattern in human inter-rater reliability:

Rubric	Model-Human	Human-Human
2-Class	0.35	0.56
3-Class	0.78	0.81
5-Class	0.82	0.85

Table 5: Agreement comparison across rubric types

This parallel suggests that model performance is fundamentally constrained by the same factors affecting human reliability. The poor performance on binary classification appears to stem from the rubric's inadequacy rather than model limitations. When provided with sufficiently granular evaluation criteria, LLMs approach human-level reliability.

4.4 Implications

These findings demonstrate that LLMs can achieve near-human reliability in story retell assessment when provided with appropriate rubric structures. The critical factor appears to be rubric design rather than model capability. Binary classifications prove problematic for both humans and machines, while detailed rubrics enable consistent evaluation.

The model's conservative grading tendency—high precision but lower recall for positive categories—may actually benefit educational applications. False positives (incorrectly identifying poor comprehension as good) pose greater instructional risks than false negatives, as they could lead teachers to overlook students needing support. The model's bias toward identifying weaknesses aligns with formative assessment goals of catching students who need help.

The strong performance on five-class rubrics (QWK = 0.82) suggests AI assessment has reached practical viability for supporting classroom instruction. However, this performance depends critically on well-designed evaluation criteria that provide sufficient granularity to capture meaningful distinctions in student performance.

5 Study 2: Model-Human Agreement Patterns

5.1 Research Questions and Approach

Building on Study 1's finding that models approach human reliability with detailed rubrics, Study 2 investigates deeper patterns in model-human agreement. Specifically, we examine how rating consistency relates to assessment uncertainty and whether models exhibit systematic grading tendencies similar to human raters.

We expanded our analysis to include Claude Sonnet 4 and Gemini 2.0 Flash alongside GPT-4, testing each at three temperature settings (0, 0.5, 1.0) to examine consistency. This provided nine model configurations plus three human raters for comparison. Given Study 1's poor results with binary classification, we focused exclusively on the three-class rubric.

We analyze these data in two ways: first by examining the differences in Cohen's Kappa when the human raters agreed and disagreed, and whether

or not the human raters score the comprehension activity as 1. Next we evaluate the consistency of raters directional bias (i.e., whether they gave higher or lower grades than average) by estimating a multilevel regression model with random intercepts of the students who was being graded and raters (humans and models). Then we compared these random intercepts to see whether their were patterns of rater bias within and between models.

Figure 1 presents Cohen's kappa values for pairwise comparisons between human raters and AI models. Across all ratings (human and model), agreement was moderate to high, with kappas ranging from .50 to 1.0. The highest agreement occurred within models, indicating that temperature differences between 0 and 1 do not introduce meaningful variability in coding.

5.2 Key Findings

Cohen's kappa values for pairwise comparisons between human raters and AI models revealed moderate to high agreement, with kappas ranging from .50 to 1.0. The highest agreement occurred within models, indicating that temperature differences between 0 and 1 do not introduce meaningful variability in coding. By contrast, the lowest agreements were observed between human coders and the models.

When all human coders agreed, the kappa values between models and human coders increased to between .66 and .80. Alternatively, when at least one coder disagreed, the models showed low inter-rater reliability with the human raters, with kappas ranging from 0.15 to 0.45. The same pattern emerged when comparing cases in which no human coder gave a score of 1 (i.e., raters were confident the student either did or did not comprehend the text) versus cases in which at least one human rater assigned a score of 1 (i.e., at least one human was uncertain about whether the student comprehended the text). These findings suggest that the models align more closely with human judgments when humans themselves are more certain of the outcome.

It is noteworthy that two of the human raters (Rater 1 and Rater 2) tended to agree even in uncertain cases. Their kappas were 0.65 when at least one human rater disagreed and 0.70 when at least one human rater gave a score of 1. However, the models still tended to diverge more from these raters under uncertain circumstances. This suggests that even when humans reach consensus in difficult cases, the models may continue to struggle to

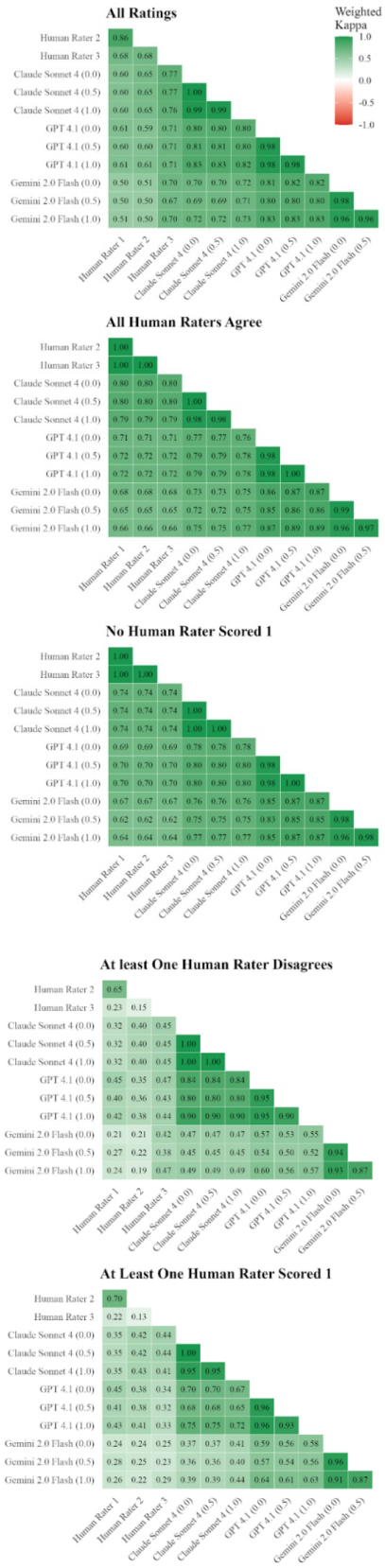

Figure 1: Pairwise weighted kappa heatmaps comparing agreement among human raters and AI models across multiple rating conditions.

align with them, potentially because the models are sensitive to the uncertainty reflected in these cases.

Furthermore, even when the humans were uncertain, the models maintained high internal consistency, as indicated by strong within-model reliability. For example, when at least one human rater disagreed or assigned a retell a score of 1, the Claude Sonnet 4 models consistently exhibited very high agreement within model ($\kappa = .95$–1.0).

Figure 2: Random intercept estimates for human and AI raters across groups.

Figure 2 presents the random intercepts from a multilevel regression model predicting ratings by rater. The model included random intercepts for each retell to account for performance-related differences across students. Thus, the random intercepts for each rater can be interpreted as systematic deviations from the overall mean rating (i.e., an intercept of 0 indicates that the rater's judgments consistently align with the grand mean).

Random intercepts from a multilevel regression model predicting ratings by rater revealed systematic grading tendencies. Models exhibited consistent levels and directions of bias relative to the mean. Specifically, Gemini tended to assign higher scores than all other raters, Claude tended to assign lower scores, and GPT produced ratings closest to the mean.

Human raters showed more variation: two raters (Human Rater 1 and Human Rater 2) consistently assigned lower scores, while one rater (Human Rater 3) tended to assign higher scores. This pattern aligns with the inter-rater reliability findings, which showed that Human Raters 1 and 2 had higher reliability with each other compared to Human Rater 3.

5.3 Implications for Hybrid Assessment

These findings suggest that AI models may be particularly effective for evaluating clear-cut cases, where human raters also show high certainty and agreement. In contrast, more difficult-to-evaluate student responses—those requiring additional context, nuanced interpretation, or expert teacher judgment—could be flagged for human review. Leveraging AI confidence scores to identify such cases can support hybrid assessment approaches that combine the efficiency of automated scoring with the depth and expertise of human evaluation. This layered approach ensures that straightforward judgments are handled quickly and consistently, while complex or ambiguous cases receive the careful consideration of trained educators.

6 Study 3: Analysis of Grading Rationales [Exploratory]

6.1 Research Questions

While Studies 1 and 2 established that models can achieve human-level scoring reliability, a critical question remains: Do models reach correct answers through human-like reasoning? Understanding whether models identify the same strengths and weaknesses that teachers recognize has profound implications for using AI-generated feedback in formative assessment. This exploratory study examines whether model explanations align with human reasoning and whether such alignment predicts scoring accuracy.

6.2 Methods

One human rater (former classroom teacher, M.Ed.) provided written justifications for all 94 story retell scores using the three-class rubric. We then modified prompts for Claude Sonnet 4, GPT-4.1, and Gemini 2.0 Flash to request explanations alongside scores.

We assessed explanation similarity based on conceptual alignment rather than exact wording, examining: identification of similar strengths or weaknesses, reference to comparable story elements or gaps, assessment of narrative flow and comprehension quality, overall evaluation tone (weak/medium/strong). For example, human noting "lacks essential narrative elements" and model stating "missing key story components" were considered conceptually similar. This approach fo-

cused on substantive agreement rather than linguistic matching.

6.3 Results

6.3.1 Quantitative Context: When Models Succeed and Struggle

Before examining reasoning quality, we established when models achieve accurate scoring. Analysis revealed no instances of maximal disagreement among humans (0 vs 2 scores), suggesting disagreements occur at category boundaries rather than reflecting fundamental assessment differences.

For the approximately two-thirds of cases (62 out of 94) where all three human raters assigned identical scores, we considered these as potentially "easy to grade" responses—those with clear indicators of quality that multiple raters could consistently identify. For these unanimous cases, the human ground truth score was simply the agreed-upon score. For the remaining one-third of cases (32 out of 94) where human raters showed some level of disagreement, we considered these as potentially ambiguous or difficult-to-assess responses. For these non-unanimous cases, the human ground truth was determined by majority vote.

We then compared these human ground truth scores against the model consensus scores (determined by majority vote across Claude Sonnet 4, GPT-4.1, and Gemini 2.0 Flash) to assess how model performance varies based on the inherent difficulty of the assessment task.

Agreement Level	Cases	Direct	QWK
Unanimous	62 (66%)	82.3%	0.808
Non-Unanimous	32 (34%)	50.0%	0.516

Table 6: Model consensus performance by human agreement

The results reveal a striking pattern: when human raters unanimously agree on a score, the model consensus achieves strong agreement (QWK = 0.808) with the human judgment. However, when human raters disagree—suggesting inherent ambiguity in the student response—model performance drops substantially to moderate agreement (QWK = 0.516), with direct agreement falling to chance levels (50%). This pattern suggests that models excel at identifying clear-cut cases but struggle with the same ambiguous responses that challenge human raters.

6.3.2 Explanation-Score Alignment Analysis

Given that models achieve high accuracy on clear cases but struggle with ambiguous ones, we examined whether the reasoning behind their scores aligns with human thinking. Do models identify the same strengths and weaknesses that teachers recognize, even when they arrive at the correct score?

Table 7 presents the relationship between explanation similarity and scoring accuracy across all three models:

Model	Similar		Different	
	Match	No Match	Match	No Match
Claude	87.0%	13.0%	66.7%	33.3%
GPT-4.1	81.3%	18.7%	65.2%	34.8%
Gemini	69.5%	30.5%	77.1%	22.9%

Table 7: Relationship Between Explanation Similarity and Score Agreement

Claude and GPT-4 demonstrate strong alignment: when their explanations resemble human reasoning, scores match 81-87% of the time. This high precision suggests that explanation similarity could serve as a confidence indicator for automated scoring. However, it's important to note that similar explanations occurred in only about 50% of cases for Claude and GPT-4, while Gemini achieved 62.8% explanation similarity.

The moderate occurrence of similar explanations (50-63% across models) reveals an important insight: many accurate scores emerge through different reasoning paths than humans employ. This suggests that models may identify alternative but potentially valid indicators of comprehension quality that differ from traditional human assessment approaches.

Gemini presents an interesting anomaly—achieving the highest rate of similar explanations but showing the weakest correlation between explanation similarity and score agreement (69.5%). This pattern suggests that surface-level explanation similarity may not always indicate deep alignment in assessment reasoning, and that the quality of reasoning alignment may be more important than the quantity.

6.3.3 Explanation Similarity as a Trust Signal

To evaluate whether explanation similarity could serve as a practical indicator of scoring reliability in operational systems, we calculated performance

metrics treating explanation similarity as a predictor of score agreement:

Model	Precision	Recall	F1	FPR
Claude Sonnet 4	87.0%	55.6%	0.68	13.0%
GPT-4.1	81.3%	56.5%	0.67	18.7%
Gemini 2.0	69.5%	60.3%	0.65	30.5%

Table 8: Performance Metrics for Using Explanation Similarity to Predict Score Agreement

The high precision for Claude and GPT-4 (>80%) suggests that when these models "speak the same language" as human raters, their scores are generally trustworthy. The low false positive rates (13-19% for Claude and GPT-4) indicate they rarely provide human-like explanations for incorrect scores—a desirable property for building educator trust.

However, the moderate recall values (55-60%) reveal that many correct scores emerge through different reasoning paths. This asymmetry has practical implications: similar explanations strongly predict reliable scores, but divergent explanations don't necessarily indicate unreliability. Models may identify alternative but valid indicators of comprehension quality that human raters don't typically consider.

6.4 Implications and Limitations

These findings suggest limited utility for direct student feedback from model explanations. While models can identify obvious strengths and weaknesses in clear-cut cases, their explanations for ambiguous responses—where students most need guidance—prove unreliable. The moderate overall explanation similarity (50-63%) indicates models identify alternative but potentially valid comprehension indicators that humans don't typically consider. This could enrich assessment if properly understood but requires careful interpretation.

The finding that models reach correct scores through different reasoning paths reinforces that AI assessment should complement rather than replace human evaluation. Models may notice patterns humans miss, but their reasoning remains opaque and potentially misleading, particularly for challenging cases.

This exploratory analysis has significant limitations. Single human rater evaluation limits generalizability. Subjective determination of explanation "similarity" introduces potential bias. The specific task and rubric may not represent broader assessment contexts. Despite limitations, consistent patterns across scoring and explanation analysis suggest current language models can handle routine assessment but shouldn't be trusted with generating feedback for ambiguous responses. The relationship between model reasoning and human judgment merits systematic study with multiple raters across diverse contexts.

7 Discussion and Conclusion

7.1 Key Findings and Implications

This research demonstrates that Large Language Models can achieve near-human reliability in story retell assessment, but with critical nuances that guide implementation. The convergence of model performance (QWK = 0.82) with human inter-rater reliability (0.85) represents practical viability, yet this aggregate metric masks important performance stratification.

The most significant finding is the dramatic performance difference based on case ambiguity. When human raters unanimously agree—approximately 66% of cases—models achieve 82% direct agreement. For the 34% generating human disagreement, model performance drops to chance levels (50%). This natural segmentation suggests a clear division of labor: AI handles routine cases while humans address ambiguous responses requiring professional judgment.

The discovery of consistent "grading personalities" across model families has important implementation implications. Claude's systematic strictness, Gemini's leniency, and GPT's moderation persist across temperature settings, indicating these are fundamental model characteristics. Schools must be aware of these tendencies to avoid inadvertently advantaging or disadvantaging students through model selection.

Rubric design emerges as foundational for both human and AI reliability. The progression from poor binary classification to strong five-class performance underscores that technology amplifies rather than compensates for assessment design quality. The conditional reliability of model explanations—strong for clear cases but unreliable for ambiguous ones—limits their utility for direct student feedback.

These findings collectively support hybrid assessment architectures that leverage respective strengths. For teachers managing 25+ students, automating the 66% of clear-cut cases could en-

able weekly rather than monthly retell assessments, dramatically increasing formative data availability while redirecting teacher expertise toward students most needing support.

7.2 Limitations and Future Directions

This exploratory study examined specific models, rubrics, and student populations. Generalization requires systematic investigation across diverse educational contexts. The single human rater providing explanations limits reasoning analysis conclusions. Future research should examine longitudinal impacts on learning outcomes and develop robust methods for uncertainty detection beyond simple confidence scores.

7.3 Conclusion

Large Language Models can reliably support story retell assessment when implemented thoughtfully within hybrid human-AI systems. The technology has reached sufficient maturity for practical application, but success depends on understanding both capabilities and limitations. By handling routine assessment tasks, AI can free teachers to focus on complex pedagogical decisions that truly require professional judgment. The goal isn't to automate education but to enhance human connections at its heart, providing teachers with better tools for understanding and supporting student learning.

Limitations

This study has several important limitations. First, our dataset consists of only 95 student responses from a specific educational context in Ghana, which may not generalize to other populations or educational settings. Second, while we tested three prominent LLMs, the rapid pace of model development means our findings may not apply to newer or different architectures. Third, our analysis of grading rationales relied on a single human rater's judgments, limiting the generalizability of our conclusions about explanation quality. Fourth, we examined only story retell tasks, and performance may differ for other literacy assessment types. Finally, our study is cross-sectional and cannot address the long-term impacts of AI-assisted assessment on student learning outcomes or teacher practices. Future work should address these limitations through larger, more diverse datasets, longitudinal studies, and multiple raters for explanation analysis.

Ethics Statement

This research was conducted with appropriate ethical oversight and student data was anonymized prior to analysis. We acknowledge several ethical considerations: First, automated assessment systems risk perpetuating or amplifying biases present in training data or human rating patterns. Second, over-reliance on AI assessment could diminish valuable teacher-student interactions that occur during traditional assessment. Third, the use of student data from Ghana raises questions about technological colonialism and the appropriateness of applying Western assessment frameworks in diverse cultural contexts. We advocate for AI assessment tools as supplements rather than replacements for human judgment, emphasizing transparency about system limitations and maintaining teacher agency in all assessment decisions. Any deployment should involve stakeholder consultation, particularly in Global South contexts, to ensure cultural appropriateness and educational benefit.

Acknowledgements

We thank the students and teachers who participated in this study, as well as the human raters who provided expert assessments.

References

Bess Altwerger, Nancy Jordan, and Nancy Rankie Shelton. 2007. *Rereading Fluency: Process, Practice, and Policy*. Heinemann, Portsmouth, NH.

Paul Black and Dylan Wiliam. 1998. Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice*, 5(1):7–74.

Kenneth S Goodman. 2006. *The Truth About DIBELS: What It Is – What It Does*. Heinemann, Portsmouth, NH.

Bente E Hagtvet. 2003. Listening comprehension and reading comprehension in poor decoders: Evidence for the importance of syntactic and semantic skills as well as phonological skills. *Reading and Writing: An Interdisciplinary Journal*, 16(6):505–539.

Janell P Klesius and Susan P Homan. 1985. A validity and reliability update on the informal reading inventory with suggestions for improvement. *Journal of Learning Disabilities*, 18(2):71–76.

Alicia M Marcotte and John M Hintze. 2009. Incremental and predictive utility of formative assessment methods of reading comprehension. *Journal of School Psychology*, 47(5):315–335.

Katherine Maria. 1990. *Reading Comprehension Instruction: Issues and Strategies*. York Press, Parkton, MD.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul 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.

Deborah K Reed and Sharon Vaughn. 2012. Retell as an indicator of reading comprehension. *Scientific Studies of Reading*, 16(3):187–217.

D Royce Sadler. 1989. Formative assessment and the design of instructional systems. *Instructional Science*, 18(2):119–144.

J Schneider, B Schenk, C Niklaus, and M Vlachos. 2023. Towards LLM-based autograding for short textual answers. arXiv preprint arXiv:2309.11508.

R M Wilson, Linda B Gambrell, and W R Pfeiffer. 1985. The effects of retelling upon reading comprehension and recall of text information. *The Journal of Educational Research*, 78(4):216–220.

Beyond the Hint: Using Self-Critique to Constrain LLM Feedback in Conversation-Based Assessment

Tyler Burleigh
Khan Academy
tylerb@khanacademy.org

Jenny Han
Khan Academy
jennyhan@khanacademy.org

Kristen DiCerbo
Khan Academy
kristen@khanacademy.org

Abstract

Large Language Models in Conversation-Based Assessment tend to provide inappropriate hints that compromise validity. We demonstrate that self-critique – a simple prompt engineering technique – effectively constrains this behavior. Through two studies using synthetic conversations and real-world high school math pilot data, self-critique reduced inappropriate hints by 90.7% and 24-75% respectively. Human experts validated ground truth labels while LLM judges enabled scale. This immediately deployable solution addresses the critical tension in intermediate-stakes assessment: maintaining student engagement while ensuring fair comparisons. Our findings show prompt engineering can meaningfully safeguard assessment integrity without model fine-tuning.

1 Background

1.1 Introduction

Conversation-Based Assessment (CBA) represents an innovative approach to educational evaluation. In CBA, students engage in dialogue with a chatbot while being assessed, which can improve test score validity (Yildirim-Erbasli and Bulut, 2023). Unlike traditional formats, CBA enables natural language responses that expand construct coverage (Bejar, 2017) while providing two unique assessment advantages: immediate, tailored feedback to enhance engagement, and follow-up questions that probe deeper understanding when initial responses are incomplete.

While CBA has shown promise in low-stakes formative assessments, intermediate-stakes assessments present a unique challenge (Perie et al., 2009). These assessments require both student engagement to ensure validity (Eklöf, 2010; Finn, 2015) and standardized conditions to enable fair comparisons between students. This creates tension between providing motivating feedback and maintaining assessment standardization.

The integration of Large Language Models (LLMs) into CBA systems presents both an opportunity and a challenge. LLMs excel at providing supportive, encouraging responses that could enhance student engagement – a critical factor for assessment validity. They achieve this through training that maximizes human preferences (Ziegler et al., 2020). However, this same preference-maximizing behavior leads LLMs to naturally provide overly helpful responses. These responses may include inappropriate hints, solutions, or answers (Jones and Bergen, 2024). For assessments where protecting validity and comparability is critical, LLM behavior must be carefully constrained to harness engagement benefits while preventing inappropriate assistance (Puech et al., 2025).

1.2 Constraining LLM behavior

The critical need to prevent inappropriate assistance in assessment contexts makes methods for constraining LLM behavior essential. While model tuning can modify behavior through weight updates, prompt engineering (PE) offers a more accessible approach using carefully crafted instructions and code-based techniques (Vijayan and Vengathattil, 2025).

Among PE techniques for behavioral constraint (Sahoo et al., 2024), self-critique shows particular promise. This technique uses the LLM to critique and revise its own responses (He et al., 2025), demonstrating effectiveness at reducing hallucinations (Dhuliawala et al., 2023) and performing well as a self-critic for short inputs (He et al., 2025), making it well-suited for assessment applications where responses are typically brief.

1.3 Evaluation methodology

Rigorous measurement is essential for evaluating LLM behavior in assessment contexts. Evaluating whether an LLM gives inappropriate hints requires measurement methodology borrowed from

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 79–85

social science (Ameli et al., 2024; Wallach et al., 2024). The process begins with construct definition and task development (Wallach et al., 2024), followed by evaluation using multiple human raters and assessment of interrater reliability (Belur et al., 2021).

To enable evaluation at scale, researchers increasingly employ LLM judges that complement human evaluation. While requiring careful validation against human judgments (Li et al., 2024), LLM judges have demonstrated accuracy in educational contexts including standards alignment (Lucy et al., 2024), response scoring (Frohn et al., 2025; Morris et al., 2024), and content refinement (Clark et al., 2025). This dual approach – combining human ground truth with validated LLM evaluation – enables rapid testing and experimentation during development of assessment safeguards.

1.4 Research questions

Intermediate-stakes CBA faces a critical tension: leveraging LLMs' engagement benefits while preventing their tendency to provide inappropriate assistance. This paper addresses this challenge through the following research questions:

1. How accurately can an LLM judge detect inappropriate hints when validated against human expert judgments?

2. Can self-critique mechanisms effectively reduce inappropriate hints in LLM-based CBA?

3. Does self-critique performance generalize from synthetic development data to real-world student conversations?

To address these questions, we develop and evaluate a self-critique mechanism where the LLM evaluates and revises its own responses before delivery. Through two studies – one using synthetic conversations for development and validation, and another using real student pilot data – we demonstrate that prompt engineering can successfully constrain LLM behavior while maintaining the engagement benefits that make CBA valuable for intermediate-stakes assessment.

2 Research

To evaluate whether self-critique can effectively prevent inappropriate hints in CBA interactions, we conducted two complementary studies. Study 1 used synthetic conversations between LLM-simulated students and the assessment chatbot (hereafter "ProctorBot") to develop and validate our self-critique mechanism under controlled conditions. Study 2 validated these findings using real student conversations from high-school math assessment pilots, demonstrating the practical effectiveness of self-critique in authentic assessment contexts.

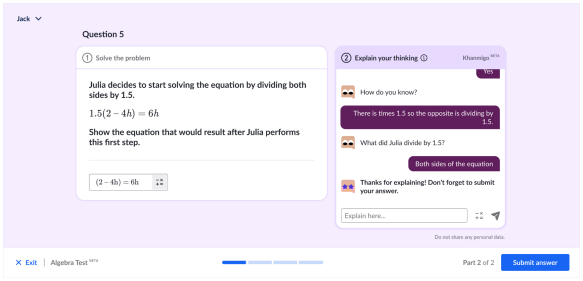

Figure 1: Screenshot of the Explain Your Thinking CBA item type. The student first answers a math problem (left), and then has a conversation about the problem (right) which is designed to assess their conceptual understanding.

2.1 Study 1: Pre-pilot development and validation using synthetic data

Study 1 developed and evaluated the self-critique mechanism under controlled conditions. Using synthetic conversations between LLM-simulated students and ProctorBot, we: (1) collected human expert labels to establish ground truth, (2) validated an LLM judge for detecting inappropriate hints, and (3) conducted an A/B test comparing baseline ProctorBot against a self-critique version.

2.1.1 Methods

Definition of inappropriate hint. For this study, we define an "inappropriate hint" as a ProctorBot response that reveals a concept from the assessment criteria that students are expected to demonstrate. Unlike a response that would draw out a student's thinking and reveal what they know (e.g., a Socratic question), an inappropriate hint would state or strongly hint at one of the criteria concepts – making it difficult to assess what they know. For example, say we wanted to assess if a student understood the concept of inverse operations: If the student solved the problem `1.5x = 3` by dividing, and then ProctorBot asked "How does dividing undo the multiplication?", this would be an inappropriate hint because it reveals the inverse operations concept.

Synthetic data generation. We generated synthetic conversation data using two LLM agents: (1) ProctorBot, designed to assess and question students about their conceptual understanding of math problems, and (2) a student simulator ("StudentBot") designed to express adversarial behaviors (asking for help, expressing uncertainty, refusing to answer) expected to increase the likelihood of inappropriate hints.

Using a Python script to orchestrate conversations between the two agents, we generated 200 synthetic conversations (50 conversations × 4 math problems). Of these, 62 ended early when ProctorBot determined that StudentBot had immediately satisfied the assessment criteria. From the remaining 138 conversations, we systematically extracted 597 test cases at various conversation depths for later experimental use.

The synthetic conversations reflected realistic assessment interactions: StudentBot responses had a median length of 11 words, ProctorBot responses averaged 18 words, and full conversations had a median of 7 turns. To increase response diversity, we varied several StudentBot parameters across simulation runs (see Appendix A).

From this corpus, we sampled 120 ProctorBot responses for human labeling, with some conversations contributing multiple responses from different points in the interaction.

Data labeling and ground truthing. Three subject-matter experts labeled each ProctorBot response as containing an inappropriate hint or not. We presented each response with full context: conversation history, the math problem, assessment criteria, and the inappropriate hint definition.

Initial inter-rater agreement was slight (Fleiss' kappa, denoted κ_F = 0.191 [0.070, 0.314]), with only 53 of 120 items (44%) achieving unanimous agreement. The 67 items with disagreements underwent group discussion and arbitration, resolving 59 cases. This process increased agreement to almost perfect (κ_F = 0.884 [0.798, 0.954]), establishing a reliable ground truth dataset for subsequent analyses.

LLM judge development and validation. To develop an LLM judge capable of detecting inappropriate hints, we tested three prompt variations that differed only in how the target behavior was specified:

1. **Baseline-prompt**: Provided only a simple definition stating that an inappropriate hint "gives away KEY information from the Criteria that has not already been mentioned"
2. **Enhanced-specificity**: Added clarification that "simply mentioning KEY concepts from the Criteria… IS ENOUGH to be considered leading"
3. **Example-based**: Supplemented the baseline definition with six annotated examples (three inappropriate hints, three appropriate responses)

All configurations used GPT-4o with temperature=0 and included the variables in Table 1 as context for the LLM judge. We ran each configuration 20 times on our 120-item dataset to ensure stable estimates, then calculated Cohen's kappa (denoted κ_C) for two-rater agreement and confidence intervals using bootstrap resampling (N=1000) to account for clustering.

Context Element	Description
Problem	The math problem that the student is having a conversation about
StudentAnswer	The student's answer to the problem
Criteria	The assessment criteria
BehaviorDefinition	The definition of inappropriate hints
ConversationHistory	The conversation between student ProctorBot so far
ProctorBotResponse	The ProctorBot response that is being judged, which immediately follows ConversationHistory

Table 1: Context elements provided to the LLM judge.

The enhanced-specificity configuration obtained substantial agreement with ground truth (Landis and Koch, 1977), and had the best balance of performance and simplicity (κ_C = 0.629 [0.611, 0.648]) – outperforming the baseline-prompt (κ_C = 0.553 [0.533, 0.569]), and performing comparably to the significantly more complex example-based prompt (κ_C = 0.612 [0.597, 0.628]). Thus, we decided to use the enhanced-specificity prompt for our implementation of self-critique.

Experiment to evaluate self-critique effectiveness. Having established a reliable automated method for detecting inappropriate hints through our validated LLM judge, we could now evaluate our proposed self-critique intervention at scale. The following experiment tests whether incorporating self-critique into ProctorBot's response generation process can effectively reduce the frequency

of inappropriate hints compared to the baseline system.

From our synthetic dataset of 597 test cases, we identified those with high propensity for inappropriate hints by screening each case 10 times with baseline ProctorBot. This yielded 179 conversation states that produced at least one inappropriate hint (as determined by our LLM judge).

For each of these 179 test cases, we generated responses using both baseline ProctorBot and a self-critique version, then evaluated each response using the LLM judge developed above. The self-critique mechanism employs a two-step process: (1) ProctorBot generates an initial response, then (2) a critic evaluates whether this response inappropriately reveals assessment criteria. If the critic detects an inappropriate hint, it generates a replacement response that avoids revealing assessment criteria. During development, we conducted informal qualitative review of the critic's replacement responses to ensure they maintained pedagogical appropriateness.

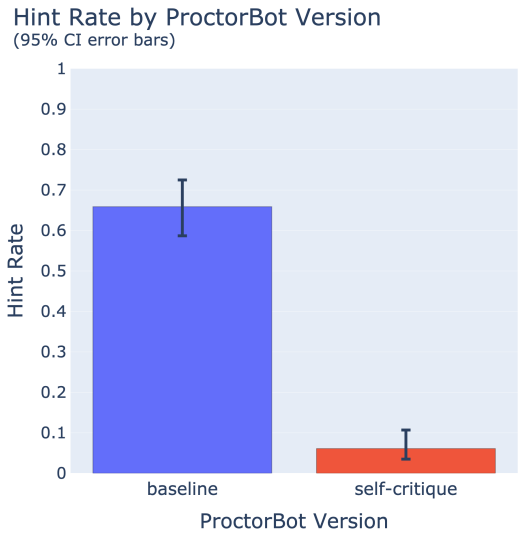

Figure 2: Results of the experiment showing the proportion of inappropriate hints with baseline and self-critique versions of ProctorBot. Self-critique dramatically reduced the rate of inappropriate hints from 65.9% to 6.1% – a 90.7% reduction.

2.1.2 Results

Self-critique dramatically reduced inappropriate hints from 65.9% (118/179) in the baseline to 6.1% (11/179), representing a 90.7% reduction. Figure 2 illustrates this substantial improvement.

To account for the paired nature of our data (same conversation states tested with both versions), we used McNemar's test, which revealed a highly significant difference ($\chi^2 = 101.23$, $p < 0.001$). Of the 111 test cases that showed different outcomes between versions (62% of all cases), 98.2% improved with self-critique: 109 cases changed from producing inappropriate hints to appropriate responses, while only 2 cases showed the opposite pattern.

These findings provided strong evidence for self-critique effectiveness in controlled settings, leading us to validate the approach with real-world data in Study 2.

2.2 Study 2: Post-pilot validation using real-world assessment pilot data

While Study 1 demonstrated self-critique effectiveness with synthetic data, validating this approach with authentic student interactions remained essential.

Study 2 validated the self-critique mechanism in authentic assessment contexts. Using real student conversations from high-school math assessment pilots, we: (1) collected human expert labels to establish ground truth, (2) validated an LLM judge for detecting inappropriate hints, and (3) conducted an A/B test comparing baseline ProctorBot against a self-critique version.

2.2.1 Methods

Data source and sampling. We analyzed conversation data from two high school math assessment pilots (algebra and geometry) conducted between April 14 and May 31, 2025, involving approximately 7,000 students and 9,000 conversations. From this corpus, we sampled 400 conversation states (specific points in conversations where ProctorBot responded), selecting 50 samples from each of eight Common Core standards problems.

To ensure sufficient positive examples given the expected low base rate of inappropriate hints, we performed stratified sampling: we pre-classified 25 examples per problem as likely containing inappropriate hints and 25 as likely not, using GPT-4.1 with the judge prompt from Study 1. We chose GPT-4.1 over GPT-4o for preliminary screening based on its superior agreement with our synthetic ground truth data.

Data labeling and ground truthing. Following the same protocol as Study 1, three subject-matter experts labeled each ProctorBot response. To reduce labeling burden, we employed a tie-break pro-

cess: two raters initially labeled each response, with a third rater resolving disagreements.

Inter-rater agreement (κ_F) was moderate during training (κ_F = 0.428 [0.305, 0.506]) and initially moderate for the main labeling session (κ_F = 0.571 [0.447, 0.682]).[1] Exercise-level analysis revealed that one problem achieved only chance-level agreement (κ_F = -0.004 [-0.389, 0.341]), likely due to ambiguous assessment criteria. Excluding this problem increased overall agreement to substantial (κ_F = 0.669 [0.537, 0.785]).

The final ground truth dataset comprised 350 items, with 82 responses (23.4%) labeled as containing inappropriate hints. Note that this rate reflects our stratified sampling strategy, not the population prevalence in actual student conversations.[2]

LLM judge validation. We validated an LLM judge against the ground truth, testing three models (GPT-4.1, GPT-4o, and GPT-5-mini) and two prompt configurations (baseline and chain-of-thought reasoning). GPT-5-mini without chain-of-thought achieved the strongest agreement with human judgments (κ_C = 0.596 [0.497, 0.688]), reaching a moderate level of agreement (see Appendix B for complete model comparison results).

Confirmatory experiment. To validate whether the self-critique effectiveness observed in Study 1 would generalize to real student conversations, we tested three models (GPT-5-mini, GPT-4.1, GPT-4o) implementing self-critique on our 350 conversation states. We compared these to the original ProctorBot responses from the assessment pilots (baseline), with all responses evaluated using the GPT-5-mini judge.

The self-critique implementation followed the same two-step process as Study 1, with all models operating at temperature=0 (except GPT-5-mini at fixed temperature=1).

2.2.2 Results

Self-critique proved effective with real-world data. All three models showed substantial reductions in inappropriate hints compared to the baseline rate of 27.4% (96/350): GPT-5-mini achieved a 75.0% reduction (to 6.9%), GPT-4.1 a 65.6% reduction (to 9.4%), and GPT-4o a 24.0% reduction (to 20.9%). Figure 3 displays these improvements across models.

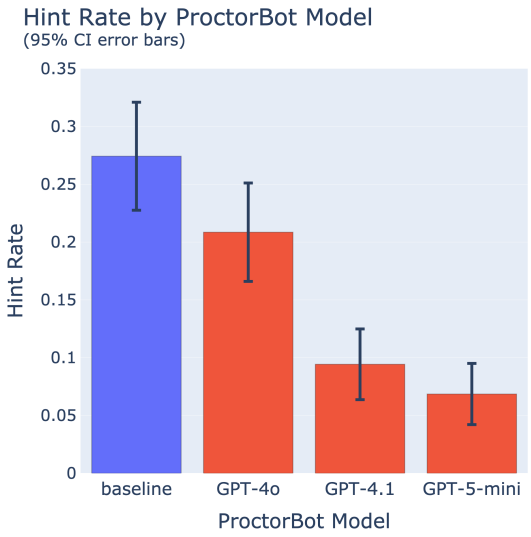

Figure 3: Inappropriate hint rates across model configurations on real-world pilot data. Self-critique implementations achieved reductions ranging from 24% (GPT-4o) to 75% (GPT-5-mini) compared to the null baseline, with all improvements being statistically significant. Error bars represent 95% confidence intervals.

McNemar's test confirmed highly significant differences for all models (all $p < 0.001$, significant after multiple comparison correction). The improvement pattern mirrored Study 1: of discordant pairs, the vast majority (89.1% for GPT-5-mini, 88.9% for GPT-4.1, 79.5% for GPT-4o) changed from inappropriate hints to appropriate responses with self-critique.

3 Conclusions

Our two-study investigation demonstrates that self-critique substantially reduces inappropriate hints in both synthetic and real-world CBA contexts.

Self-critique offers educational institutions a practical, immediately deployable solution for constraining LLM behavior in Conversation-Based Assessment. Organizations can implement this safeguard through simple prompt modifications, avoiding the costs and complexity of model fine-tuning. The technique's accessibility makes it particularly valuable for institutions with limited technical resources.

Our systematic evaluation methodology provides a template for assessing LLM behaviors in educational contexts. We progressed from synthetic to

[1] We excluded tie-break labels from agreement calculations as they are conditionally sampled only when initial raters disagree, violating assumptions for valid kappa statistics.

[2] The true population rate is likely substantially lower, as we deliberately oversampled conversations initially classified as containing inappropriate hints to ensure sufficient positive examples for analysis.

real-world data with rigorous human validation. The significant reductions in inappropriate hints across both studies validate self-critique as an effective safeguard. The same process could be used to attempt to reduce answer giving in other tutor scenarios where providing the answer is not desired. However, important limitations remain. Our evaluation focused specifically on mathematics assessment and hints that reveal assessment criteria. Generalization to other domains and types of assistance requires further investigation. Additionally, we did not systematically evaluate whether self-critique impacts overall response quality or educational value. Our focus remained exclusively on inappropriate hint reduction. While informal qualitative review during development suggested that responses remained pedagogically appropriate, quantifying any trade-offs between constraint effectiveness and response quality remains an open question.

Together with other emerging approaches for quality assurance in educational AI, self-critique offers a targeted solution for constraining LLM outputs through prompt engineering. Our contribution shows that even simple, immediately deployable techniques can meaningfully reduce inappropriate LLM behaviors and advance assessment validity when grounded in rigorous evaluation. As educational institutions navigate the integration of generative AI, this combination of theoretical frameworks, empirical validation, and practical tools will prove essential for maintaining the standards that make assessment meaningful.

4 Appendix A: StudentBot parameter variations

To increase the diversity of synthetic student responses in Study 1, we varied the following StudentBot parameters across simulation runs:

- **Model selection**: GPT-4o and Llama-3.1
- **Initial answer correctness**: Whether StudentBot provided a correct or incorrect answer to the initial math problem
- **Student persona traits**: Anxiety level, communication style (formal vs. informal), patience, and engagement level

These variations ensured that our synthetic dataset captured a range of student behaviors and interaction patterns, improving the robustness of our inappropriate hint detection and self-critique evaluation.

5 Appendix B: LLM judge model comparison results

Complete results from Study 2 LLM judge validation (κ_C with 95% confidence intervals):

- **GPT-5-mini**: κ_C = 0.596 [0.497, 0.688] (without chain-of-thought); 0.551 [0.445, 0.652] (with chain-of-thought)
- **GPT-4.1**: κ_C = 0.422 [0.304, 0.527] (without chain-of-thought); 0.437 [0.303, 0.550] (with chain-of-thought)
- **GPT-4o**: κ_C = 0.195 [0.086, 0.303] (without chain-of-thought); 0.320 [0.200, 0.431] (with chain-of-thought)

References

Siavash Ameli, Siyuan Zhuang, Ion Stoica, and Michael W. Mahoney. 2024. A Statistical Framework for Ranking LLM-Based Chatbots. *arXiv preprint*. ArXiv:2412.18407 [stat].

Isaac I. Bejar. 2017. A Historical Survey of Research Regarding Constructed-Response Formats. In Randy E. Bennett and Matthias von Davier, editors, *Advancing Human Assessment: The Methodological, Psychological and Policy Contributions of ETS*, pages 565–633. Springer International Publishing, Cham.

Jyoti Belur, Lisa Tompson, Amy Thornton, and Miranda Simon. 2021. Interrater Reliability in Systematic Review Methodology: Exploring Variation in Coder Decision-Making. *Sociological Methods & Research*, 50(2):837–865. Publisher: SAGE Publications Inc.

Hannah-Beth Clark, Margaux Dowland, Laura Benton, Reka Budai, Ibrahim Kaan Keskin, Emma Searle, Matthew Gregory, Mark Hodierne, and John Roberts. 2025. Auto-Evaluation: A Critical Measure in Driving Improvements in Quality and Safety of AI-Generated Lesson Resources. *The AI + Open Education Initiative*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-Verification Reduces Hallucination in Large Language Models. *arXiv preprint*. ArXiv:2309.11495 [cs].

Hanna Eklöf. 2010. Skill and will: test-taking motivation and assessment quality. *Assessment in Education: Principles, Policy & Practice*, 17(4):345–356.

Bridgid Finn. 2015. Measuring Motivation in Low-Stakes Assessments. *ETS Research Report Series*, 2015(2):1–17.

Scott Frohn, Tyler Burleigh, and Jing Chen. 2025. Automated Scoring of Short Answer Questions with Large Language Models: Impacts of Model, Item, and Rubric Design. In *Artificial Intelligence in Education*, volume VI of *Lecture Notes in Artificial Intelligence*, pages 44–51, Palermo, Italy. Springer.

Yancheng He, Shilong Li, Jiaheng Liu, Weixun Wang, Xingyuan Bu, Ge Zhang, Zhongyuan Peng, Zhaoxiang Zhang, Zhicheng Zheng, Wenbo Su, and Bo Zheng. 2025. Can Large Language Models Detect Errors in Long Chain-of-Thought Reasoning? *arXiv preprint*. ArXiv:2502.19361 [cs].

Cameron R. Jones and Benjamin K. Bergen. 2024. People cannot distinguish GPT-4 from a human in a Turing test. *arXiv preprint*. ArXiv:2405.08007 [cs].

J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods. *arXiv preprint*. ArXiv:2412.05579 [cs].

Li Lucy, Tal August, Rose E. Wang, Luca Soldaini, Courtney Allison, and Kyle Lo. 2024. Mathfish: Evaluating Language Model Math Reasoning via Grounding in Educational Curricula. *arXiv preprint*. ArXiv:2408.04226 [cs].

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*.

Marianne Perie, Scott Marion, and Brian Gong. 2009. Moving Toward a Comprehensive Assessment System: A Framework for Considering Interim Assessments. *Educational Measurement: Issues and Practice*, 28(3):5–13.

Romain Puech, Jakub Macina, Julia Chatain, Mrinmaya Sachan, and Manu Kapur. 2025. Towards the Pedagogical Steering of Large Language Models for Tutoring: A Case Study with Modeling Productive Failure. *arXiv preprint*. ArXiv:2410.03781 [cs].

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. *arXiv preprint*. ArXiv:2402.07927 [cs].

Resmi Vijayan and Sunish Vengathattil. 2025. Using the Right Tool: Prompt Engineering vs. Model Tuning. *International Journal of Innovative Science and Research Technology*, pages 274–284.

Hanna Wallach, Meera Desai, Nicholas Pangakis, A. Feder Cooper, Angelina Wang, Solon Barocas, Alexandra Chouldechova, Chad Atalla, Su Lin Blodgett, Emily Corvi, P. Alex Dow, Jean Garcia-Gathright, Alexandra Olteanu, Stefanie Reed, Emily Sheng, Dan Vann, Jennifer Wortman Vaughan, Matthew Vogel, Hannah Washington, and Abigail Z. Jacobs. 2024. Evaluating Generative AI Systems is a Social Science Measurement Challenge. *arXiv preprint*. ArXiv:2411.10939 [cs].

Seyma N. Yildirim-Erbasli and Okan Bulut. 2023. Conversation-based assessment: A novel approach to boosting test-taking effort in digital formative assessment. *Computers and Education: Artificial Intelligence*, 4:100135.

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. *arXiv preprint*. ArXiv:1909.08593 [cs].

Investigating Adversarial Robustness in LLM based Automated Essay Scoring

Renjith P Ravindran
ETS Assessment Services
rpravindran@ets.org

Ikkyu Choi
ETS Research Institute
ichoi001@ets.org

Abstract

Automated Essay Scoring (AES) is one of the most widely studied applications of Natural Language Processing (NLP) in education and educational measurement. Recent advances with pre-trained Transformer-based large language models (LLMs) have shifted AES from feature-based modeling to leveraging contextualized language representations. These models provide rich semantic representations that substantially improve scoring accuracy and human–machine agreement compared to systems relying on handcrafted features. However, their robustness towards adversarially crafted inputs remains poorly understood. In this study, we define adversarial input as any modification of the essay text designed to fool an automated scoring system into assigning an inflated score. We evaluate a fine-tuned DeBERTa-based AES model on such inputs and show that it is highly susceptible to a simple text duplication attack, highlighting the need to consider adversarial robustness alongside accuracy in the development of AES systems.

1 Introduction

Automated Essay Scoring (AES) is one of the earliest applications of Natural Language Processing (NLP) to educational assessment, with roots dating back to the 1960s (Page, 1967). Over the decades, AES systems have evolved from statistical models with shallow surface-level features to highly sophisticated neural architectures (Beigman Klebanov and Madnani, 2020). Traditional approaches often relied on handcrafted features designed to approximate lexical diversity, syntactic complexity, discourse organization, and stylistic control. For example, the use of connectives such as "therefore" or "in conclusion" could serve as proxies for argumentative structure, while measures such as type–token ratio or average sentence length are aimed at capturing lexical richness (Chodorow and Burstein, 2004). These approaches, although effective to some extent, are inherently limited: they depend heavily on feature engineering and are vulnerable to superficial manipulation by test takers (Powers et al., 2001; Chodorow and Burstein, 2004; Perelman, 2020).

The advent of deep learning (Goodfellow et al., 2016), and more recently pre-trained Transformer (Vaswani et al., 2017) based large language models (LLMs), has reshaped the AES landscape. Transformer-based models such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2020) learn contextual representations of text that capture lexical, syntactic, and semantic information simultaneously. When fine-tuned on essay scoring datasets, these models substantially increase the agreement between machine predictions and human raters, often measured using Quadratic Weighted Kappa (QWK) (Li and Ng, 2024). This leap in performance could lead to growing enthusiasm towards operational deployment of LLM-based AES in high-stakes testing environments.

Yet, the question of robustness remains underexplored (Ding et al., 2020; Kabra et al., 2022). Accuracy gains in typical test settings do not guarantee resilience under adversarial conditions. Adversarial attacks in NLP — ranging from synonym substitution in sentiment analysis (Zhou et al., 2021) to input perturbations in machine translation (Michel et al., 2019) — have shown that state-of-the-art models can be surprisingly fragile. In educational contexts, this fragility has serious implications. Unlike sentiment classification or translation, AES models directly influence student outcomes. If models can be "fooled" by trivial manipulations, such as artificially inflating essay length or inserting irrelevant but sophisticated-sounding sentences or words, the integrity of automated scoring is jeopardized. This is particularly concerning given the high stakes of standardized assessments, where

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 86–91

even a one-point increase in an essay score can affect admissions or scholarship decisions.

Prior work has begun to highlight these vulnerabilities. Ding et al. (2020) showed that content scoring systems can be misled by adversarial strings of meaningless characters. Kabra et al. (2022) proposed toolkits for systematically probing AES robustness, underscoring the need for adversarial evaluation. Jeon and Strube (2021) demonstrated that essay length continues to exert disproportionate influence on neural AES models, echoing concerns that date back to earlier systems (Chodorow and Burstein, 2004). Collectively, this line of work suggests that LLM-based AES models, despite their sophistication, may inherit structural weaknesses from both feature-based and neural predecessors.

In this preliminary study, we take a focused step toward systematically evaluating adversarial robustness of an LLM-based AES model. Specifically, we examine the behavior of a DeBERTa-based scoring system fine-tuned on the Persuade 2.0 corpus (Crossley et al., 2024). We design and test three adversarial scenarios that are both simple to implement and highly plausible in real testing conditions:

- Appending high-impact words, where the test taker simply append few words that are likely to be found in high scoring essays. If an automatic scoring model is overly relying on uni-grams such essays could see a boost in score.

- Fancy-language injection, where a short paragraph of complex, topic-agnostic sentences are appended to the essay to mimic advanced vocabulary and sentence structure.

- Text duplication, where a test taker repeats their essay once or twice to artificially inflate length. Scoring models often pick up essay length as a proxy to essay quality, duplication of text is the easiest way to increase essay length.

To provide additional context for robustness, we also examine noise-based perturbations such as scrambling words or sentence spans. These manipulations allow us to probe the model's reliance on lexical coherence versus discourse-level organization.

Findings from our preliminary study are twofold:

- We demonstrate that a DeBERTa-based AES model, while achieving strong baseline accuracy (QWK = 0.87), is highly vulnerable to text duplication, with systematic and substantial score inflation.

- We show that the model is relatively robust to high-impact lexicon and sentence insertions, suggesting that sophisticated vocabulary and structuring without semantic relevance does not easily fool the system.

Taken together, these findings highlight a central tension in AES research: while LLMs improve accuracy, they do not automatically confer robustness. Even trivial adversarial strategies can yield large score changes, raising fairness and validity concerns. We argue that adversarial robustness should be treated as a primary design criterion for AES, alongside scoring accuracy and reliability, and we hope this work stimulates further research in this direction.

2 Experiment Setup

For our experiments we use the Persuade Corpus 2.0 (Crossley et al., 2024), a large-scale dataset of approximately 25,000 student essays written by grades 6–12 in response to argumentative writing prompts. Each essay in this corpus has been scored holistically by human raters on a six-point ordinal scale (1 = weakest, 6 = strongest), reflecting overall writing quality rather than individual analytic traits. The dataset is particularly suitable for adversarial evaluation because it is both large enough to fine-tune LLMs effectively and realistic in content, covering authentic student writing with diverse levels of proficiency. In addition, Persuade 2.0 is a recent corpus explicitly designed to advance AES research, which makes it a valuable benchmark for studying not only predictive performance but also model robustness. For training and evaluation, we adopt the official splits, which contain 15,528 items in the training set and 10,356 items in the test set.

Our AES model is built on DeBERTa (He et al., 2020), a Transformer-based large language model that improves upon BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) through disentangled relative attention mechanisms. Specifically we used DeBERTa V3 base (He et al., 2021) via the Hugging Face Transformers Python Module. An important parameter that is relevant for this particular study is the token limit, which limits the size

transform	mean change (sd)
scramble-words	-2.10 (1.0)
scramble-sents	-0.06 (0.2)
add-low-words	-0.13 (0.2)
add-high-words	-0.09 (0.1)
add-smoke	-0.05 (0.2)
length-2x	0.93 (0.3)
length-3x	1.28 (0.6)

Table 1: Mean score change (SD) per transform (score range 1-6).

of the input text. Thanks to relative positioning bias in DeBERTa, the maximum number of tokens is given by $(2k-1)l$, where l is the number of layers and k is the maximum relative distance allowed between tokens. Since in DeBERTa base $l = 12$ and $k = 512$, we can have a maximum of 24,528 tokens in the input text. This is lower than the thrice the number of tokens in the longest essay (1902 tokens [1]) in the dataset. To adapt DeBERTa for essay scoring, we attach a simple regression head on top of the [CLS] token embedding to predict continuous essay scores in the range 1–6. The head consists of a two-layer feed-forward network trained jointly with the DeBERTa encoder, so that the model learns both task-specific features and general linguistic representations. Training is performed with mean squared error (MSE) loss, and model quality is evaluated using Quadratic Weighted Kappa (QWK), a standard metric for measuring agreement with human raters in AES research. When evaluated on the test-set our model gives a QWK of 0.87.

2.1 Attacks

2.1.1 Adding High Scoring Words

This attack tests whether the scoring model is relying on uni-grams to score the essay. High scoring essays are likely to have impactful vocabulary. Thus test takers may add such words out of context in the hope of triggering the scoring model to award a higher score. To find such words, we split the dataset (train) into two, a high scoring set which has essays having scores 4, 5, 6 and a low scoring set with essays having score 1, 2, 3. Now for each word we find its log-odds ratio of probability of the word occurring in the high scoring set over the probability of the word occurring in the low scoring set. This allows us to rank words based on how likely they are to be found exclusively in high scoring essays. The attack then is to append a sequence of 10 words sampled randomly from the top 100 of these words to each of the test essays. As this is a preliminary study the choice of 10 is informed intuitively as the likely number of words test takers may add. We defer testing a range of numbers to future work. Here is such a random sample: *dependence*, *traditional*, *theatre*, *platforms*, *extracurriculars*, etc. This attack is referred to as add-high-words.

2.1.2 Adding Fancy Language

Our next attack is to test the impact of adding a paragraph with impactful sentence structure and vocabulary. This simulates the situation where test takers may memorise a piece of fancy sounding text that could be added to any essay in order to trigger the machine to give a higher score. To study the effect of fancy-language injection, we transform the essay texts by adding the following to the end of each essay: "*Conceptual dynamics often emerge through the oscillations of undefined frameworks. This interaction, while nebulous, suggests a layered intentionality. Consequently, abstraction persists as both method and outcome.*". This is an arbitrary piece of text intended to add a dose of potentially high-impact vocabulary and sentence structure. As this is a preliminary study, we do not attempt to quantify what is meant by high-impact, and also limit the study to considering only a single instance. We refer to this attack as add-smoke.

2.1.3 Inflating Essay Length

Essay scores are often correlated to its length. One of the easiest ways a test taker can game this feature, without adding out of context text is to simply duplicate their essays. To study the effect of text-duplication we transform the essay text by duplicating it once, and twice, referred to as length-2x

[1] tokens here refer to lexical units after tokenisation

and length-3x, respectively.

2.1.4 Baselines

To understand the general robustness of our model we add two more transformations, scramble-words: in which words in the essay are scrambled, and scramble-sents: in which the spans of text separated by newlines are scrambled (note that this is not perfect sentence scrambling). To contrast with the add-high-words attack we include a add-low-words where we append words that are exclusively found in low scoring essays. The intention is to test if adding such words can lower the essay scores. A random sample from add-low-words : *luke*, *electrol*, *presendent*, *thay*, *negitive*, etc. We find that most of these are typos, and therefore can be expected to bring down scores when added to essays.

3 Experiments and Results

3.1 Average Score Change

Table 1 gives the mean and standard deviation of the score change induced by each transformation. The first clear observation is that scrambling words devastates performance (–2.10 average). This is expected: scrambling disrupts local coherence, making essays nonsensical.

In contrast, scrambling sentences produces almost no change (–0.06). This suggests that the DeBERTa-based AES model may be largely insensitive to discourse-level ordering of sentences. Although discourse coherence is a key aspect of human evaluation, our results imply that the model's reliance on the [CLS] embedding fails to adequately capture paragraph-level or argumentative flow. This insensitivity could become problematic if test takers deliberately manipulate essay structure while maintaining superficial lexical quality.

The more striking pattern emerges with duplication attacks. Doubling essay length (length-2x) increases average scores by +0.93, and tripling (length-3x) by +1.28. These are substantial gains considering the total score range is only 1–6. The effect size rivals the difference between adjacent holistic score levels as judged by human raters. Put differently, a mediocre essay rated 3 could be artificially boosted into the "proficient" range (4–5) simply by repetition.

Interestingly, adding high-scoring words or high impact vocabulary and sentence structure out of context doesn't increase the scores, instead marginally decreases the scores. This contrasts with anecdotal expectations that "sophisticated" vocabulary could fool models. Instead, the AES model appears somewhat robust to this type of lexical padding, possibly because embeddings capture topical mismatch between the appended text and the main essay body.

3.2 Score Change at each Human Score Level

Table 2 disaggregates score changes by human-assigned scores. This analysis yields three notable insights.

scramble-words degrades higher-quality essays more severely. Essays originally scored 6 lose over 4 points, while those scored 1 lose less than 1.

Duplication benefits mid-range essays the most. For length-2x and length-3x, the largest gains occur at human scores 3–4. For example, a 3-rated essay rises on average by +1.42 under length-3x. This reflects the model's tendency to conflate length with quality in borderline cases. Such vulnerabilities are particularly concerning because many operational decisions hinge on distinguishing "adequate" from "proficient" performance in this mid-range.

add-smoke and add-high-words has negligible effects across all bins. The consistency of near-zero changes suggests that superficial stylistic padding does not easily exploit this model.

3.3 Score Change Distribution

Average changes alone can obscure practical impact. Figure 1 therefore examines the distribution of rounded score differences under duplication.

For length-2x, nearly 80% of essays increase by at least +1 point, and around 10% gain +2 points. Such shifts could materially alter student outcomes: an essay initially rated 3 (marginal) may be reclassified as 4 (proficient).

For length-3x, the effects are even more dramatic: 50% of essays gain +1 point and 40% gain +2. In practice, this means almost every duplicated essay is rewarded, with a non-trivial fraction jumping two score categories.

Very few essays decrease in score, confirming that duplication is a high-reward, low-risk adversarial strategy.

These findings underscore the operational significance of duplication: if undetected, test takers can consistently and predictably exploit the scoring system.

transform	1	2	3	4	5	6
scramble-words	-0.39 (0.4)	-1.08 (0.4)	-1.86 (0.5)	-2.69 (0.5)	-3.59 (0.5)	-4.29 (0.3)
scramble-sents	-0.01 (0.1)	-0.04 (0.1)	-0.07 (0.2)	-0.07 (0.2)	-0.05 (0.2)	-0.02 (0.1)
add-low-words	-0.07 (0.1)	-0.10 (0.1)	-0.13 (0.2)	-0.17 (0.2)	0.17 (0.2)	0.09 (0.1)
add-high-words	-0.06 (0.1)	-0.08 (0.1)	-0.09 (0.1)	-0.12 (0.2)	0.08 (0.2)	0.00 (0.1)
add-smoke	-0.05 (0.1)	-0.07 (0.1)	-0.08 (0.2)	-0.07 (0.2)	0.03 (0.2)	0.09 (0.1)
length-2x	0.82 (0.4)	0.91 (0.3)	0.99 (0.3)	1.04 (0.3)	0.78 (0.4)	0.27 (0.3)
length-3x	1.12 (0.6)	1.42 (0.4)	1.49 (0.5)	1.33 (0.5)	0.71 (0.6)	0.01 (0.6)

Table 2: Mean score change (SD) at each human score level.

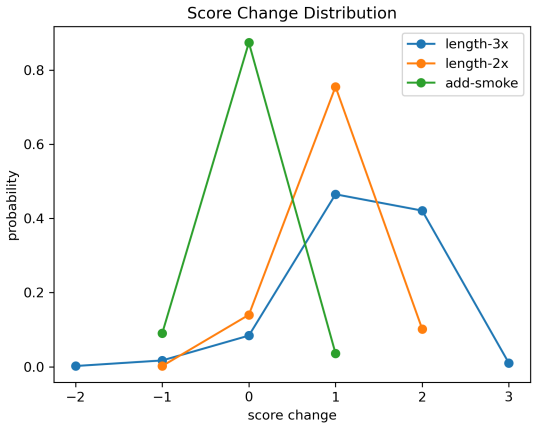

Figure 1: Score change distribution.

4 Conclusion

This study has examined the adversarial robustness of an LLM-based AES system trained on the Persuade 2.0 corpus. While the baseline DeBERTa model achieved strong agreement with human raters (QWK = 0.87), our experiments reveal that high scoring accuracy alone does not guarantee robustness to adversarially crafted responses. The most striking finding is the model's vulnerability to duplication: repeating an essay once or twice almost always leads to inflated scores, with gains of one or even two points on a six-point scale. Because such changes occur consistently across a large portion of the test set, they represent a genuine threat to the validity of AES in operational settings. Even if duplication is easy to detect with simple preprocessing, the fact that a trivial manipulation yields such predictable benefits underscores the importance of evaluating AES systems against adversarial input.

At the same time, the results also highlight areas where the model appears more robust. The insertion of sophisticated but irrelevant sentences ("smoke text") produced negligible effects, and the more systematic attempt to append vocabulary disproportionately associated with high- or low-scoring essays also failed to move predictions in a meaningful way. These negative results suggest that the model does not simply reward isolated lexical items, even when those items are correlated with writing quality in the training data. Instead, it appears to integrate vocabulary in context, discounting out-of-place words. This robustness to shallow lexical padding contrasts with the severe susceptibility to length manipulation, pointing to a specific structural weakness rather than a general fragility.

It is to be noted that these results are from our preliminary study along these lines. A major limitation of this study is that we have evaluated only one kind of model. A comprehensive evaluation is being planned as future work with multiple AES models, and to address other limitations.

More generally future studies should pursue two directions in parallel: developing systematic taxonomies of adversarial risks in AES (including semantic drift, coherence disruption, and targeted vocabulary injection), and exploring defenses that go beyond heuristic filters. Possibilities include explicit modeling of discourse, normalization against essay length, and the integration of adversarial training protocols.

Ultimately, if AES systems are to be trusted in high-stakes testing, adversarial robustness must be evaluated alongside accuracy and fairness. Our results provide early evidence that while certain manipulations are resisted, others remain alarmingly effective. Robustness cannot be assumed from model sophistication alone; it must be deliberately measured and built into the design of future AES systems.

References

Beata Beigman Klebanov and Nitin Madnani. 2020. Automated evaluation of writing – 50 years and counting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7796–7810, Online. Association for Computational Linguistics.

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.

Scott Andrew Crossley, Y Tian, P Baffour, Alex Franklin, Meg Benner, and Ulrich Boser. 2024. A large-scale corpus for assessing written argumentation: Persuade 2.0. *Assessing Writing*, 61:100865.

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*.

Yuning Ding, Brian Riordan, Andrea Horbach, Aoife Cahill, and Torsten Zesch. 2020. Don't take "nswvtnvakgxpm" for an answer –the surprising vulnerability of automatic content scoring systems to adversarial input. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 882–892, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Sungho Jeon and Michael Strube. 2021. Countering the influence of essay length in neural essay scoring. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 32–38, Virtual. Association for Computational Linguistics.

Anubha Kabra, Mehar Bhatia, Yaman Kumar Singla, Junyi Jessy Li, and Rajiv Ratn Shah. 2022. Evaluation toolkit for robustness testing of automatic essay scoring systems. In *Proceedings of the 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, CODS-COMAD '22, page 90–99, New York, NY, USA. Association for Computing Machinery.

Shengjie Li and Vincent Ng. 2024. Automated essay scoring: A reflection on the state of the art. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888, Miami, Florida, USA. Association for Computational Linguistics.

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*.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. 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 3103–3114, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Les Perelman. 2020. The babel generator and e-rater: 21st century writing constructs and automated essay scoring (aes). *Journal of Writing Assessment*, 13(1).

Donald E. Powers, Jill C. Burstein, Martin Chodorow, Mary E. Fowles, and Karen Kukich. 2001. Stumping e-rater: Challenging the validity of automated essay scoring. *ETS Research Report Series*, 2001(1):i–44.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. 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 5482–5492, Online. Association for Computational Linguistics.

Effects of Generation Model on Detecting AI-generated Essays in a Writing Test

Jiyun Zu and Michael Fauss and Chen Li
Educational Testing Service, Princeton, NJ
Correspondence: jzu@ets.org

Abstract

Various detectors have been developed to detect AI-generated essays using labeled datasets of human-written and AI-generated essays, with many reporting high detection accuracy. In real-world settings, essays may be generated by models different from those used to train the detectors. This study examined the effects of generation model on detector performance. We focused on two generation models – GPT-3.5 and GPT-4 – and used writing items from a standardized English proficiency test. Eight detectors were built and evaluated. Six were trained on three training sets (human-written essays combined with either GPT-3.5-generated essays, or GPT-4-generated essays, or both) using two training approaches (feature-based machine learning and fine-tuning RoBERTa), and the remaining two were ensembled detectors. Results showed that a) fine-tuned detectors outperformed feature-based machine learning detectors on all studied metrics; b) detectors trained with essays generated from only one model were more likely to misclassify essays generated by the other model as human-written essays (false negatives), but did not misclassify more human-written essays as AI-generated (false positives); c) the ensembled fine-tuned RoBERTa detector had fewer false positives, but slightly more false negatives than detectors trained with essays generated by both models.

1 Introduction

Generative artificial intelligence (AI) tools, such as ChatGPT, Copilot, and Gemini, have become increasingly capable at generating human-like text and are now more accessible. In education, AI has great potential at enhancing teaching, learning, and assessments (U.S. Department of Education, Office of Educational Technology, 2023). At the same time, there are also concerns about the misuse of AI in writing tasks (Lund et al., 2025). Writing assignments are routinely given in both K-12 and higher education. There are also many standardized writing tests designed to measure test takers writing proficiency, such as the ACT writing test, the Graduate Record Examinations (GRE) writing test, and the Writing assessment program (WrAP) for grades 3-12 students. These tests require test takers to write essays independently. If some test takers use generative AI tools to write essays and use these essays as their own, the validity and fairness of the writing assessment are compromised.

To address concerns about AI-generated text, many detectors have been developed to identify such content. For example, Grammarly (Grammarly Inc., 2025), Scribbr (Scribbr, 2025), and GPTZero (GPTZero, 2025) provide online tools that allow users to enter text and then output an estimated percentage of the text being AI-generated, although documentation on how they were trained is generally unpublished. Several research studies reported the training and evaluation of custom-built AI-generated essay detectors. For example, Yan et al. (2023) generated essays using GPT-3 for four writing items from a large-scale assessment. Using these essays and real human test takers' essays, the authors trained two detectors: one using supervised machine learning (ML) approach and the other by fine-tuning the pre-trained language model RoBERTa (Liu et al., 2019). The detection accuracy on a holdout test set was respectively 96% and 99.75% for these two detectors. Jiang et al. (2024) studied the accuracy and potential bias in detecting ChatGPT-generated essays. Using 10,000 essays generated by ChatGPT and 10,000 essays written by real test takers for 50 GRE writing items, the authors trained detectors using supervised ML with linguistic features extracted by e-rater (Attali and Burstein, 2006) and GPT-2-based perplexity features. Detection accuracy of the best performing detector was nearly 100% on a holdout test set, and showed no evidence of bias against non-native English speakers.

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 92–98

When AI-generated essay detectors are applied in real-world settings, several factors may affect their performances. For example, users may use a different generation model than the models used to generate the essays used for training the detectors. They may also use a different sampling temperature, different prompts, instruct the AI tool to paraphrase the generated essays to disguise its being AI-generated, or revise the generated output essays manually (themselves or other human).

Given the growing number of generative AI tools and the rapid release of newer AI models, understanding how different generation models affect the detection of AI-generated essay is an important research question. The study by Zhong et al. (2024) provides insights into this issue. The authors generated 200 essays using each of 10 different large language models (LLMs) and compared the essays in terms of linguistic features, textual similarities, and scores. They also trained a detector for each LLM using a feature-based ML approach relying on human-written essays and the 200 essays generated by that specific LLM. They found that while the detection accuracies for identifiying essays trained by the same LLM were higher than .9, when the detectors were applied to essays generated by the different LLMs, detection accuracy could be as low as .5. These findings showed the challenge of generalizing AI detectors to different generative models.

In this study, we investigate the effects of generation model on detecting AI-generated essays, expanding prior research to detectors trained on essays generated by more than one LLM as well as using a fine-tuning LLM approach. Specifically, we focus on two widely used generation models – GPT-3.5 and GPT-4 – and use writing items from a large-scale standardized English proficiency test for detector training and evaluation.

2 Method

2.1 Writing Items

We used 20 writing items from a standardized English proficiency test. The majority of the test takers are young adults. Each item asks test takers to write an essay expressing their opinion on a given topic with supporting details, with at least 100 words written within a 10-minute time limit. Essays were typed on a computer.

2.2 Data

Human-written essays We collected all test takers' responses to these 20 items when each item was administered for the first time and in test centers. The number of essays per item ranged from 192 to 6,438. For items with more than 300 essays, we randomly sampled them down to 300. The resulting total number of human-written essays used in this study was 5,745.

AI-generated essays We used GPT-3.5 turbo (version 0613) and GPT-4 (version 0613) to generate essays via the Azure OpenAI API. To generate a diverse sample of essays and match the length of human-written essays, we used 15 prompts per item – covering 5 levels of content (i.e., varying the amount of detail in the item stem or the direction of opinion to be expressed in the essay) and 3 levels for word count targets (100 words, 110 words, and 120 words). 20 essays were generated per prompt. Sampling temperature was set to 1.2 to balance text variance and text quality. Artificial typos were added to an average of 3.5% of the words using the python package *typo* (Kumar, 2022). 6,000 essays were generated using each generation model, resulting from 20 items $\times$ 15 prompts $\times$ 20 essays.

Training and test sets The 5,745 human-written essays were given a label of 0 (i.e., not AI-generated), and the 6,000 GPT-3.5-generated and 6,000 GPT-4-generated essays were given a label of 1. These essays and labels are referred to as the total dataset. Among them, we randomly selected 1,000 human-written, 1,000 GPT-3.5-generated, and 1,000 GPT-4-generated essays as the test sets for evaluating detectors' performances. From the remaining 4,745 human-written, 5,000 GPT-3.5-generated, and 5,000 GPT-4-generated essays, we created three training sets to build AI-generated essay detectors. All three training sets contain the same 4,745 human-written essays and differ by the AI-generated essays: respectively 5,000 GPT-3.5-generated, 5,000 GPT-4-generated, and a combination of randomly selected 2,500 GPT-3.5-generated and 2,500 GPT-4-generated essays. These training sets are named as Human + GPT-3.5, Human + GPT-4, and Human + GPT-3.5 + GPT-4.

2.3 Detector Training

We trained detectors for AI-generated-essay using a combination of two training approaches crossing the three training sets described in the previous

section. The two training approaches are feature-based machine learning (ML) approach and fine-tuning RoBERTa. Two additional detectors were ensembled from the detectors trained on Human + GPT-3.5 and Human + GPT-4, respectively for the ML and fine-tuning approach.

Feature-based ML approach Eleven features – 10 high-level linguistic features and the logarithm of GPT-2-based perplexity of an essay - were used in the ML approach. The 10 high-level linguistic features were extracted using e-rater (Attali and Burstein, 2006). These features represent grammatical errors, usage errors, mechanics errors, organization, development, word length, word frequency, collocation and preposition, sentence variety, and discourse coherence aspects of the essays. Perplexity (i.e., exponential of the cross-entropy loss) reflects how uncertain a language model is with predicting the next token given previous tokens. A higher value indicates the text sequence is less likely to be generated by the language model. It has been found to be contributing features for detecting AI-generated essays in previous research (see e.g., Yan et al., 2023; Jiang et al., 2024). Because the essays were relatively short, we only used perplexity for the entire essays. Although essays were generated by GPT-3.5 and GPT-4, those two models were proprietary and perplexity were not available via the API. Thus, we calculated the open-source GPT-2 perplexity using the transformers library (Wolf et al., 2020).

Four type of ML classifiers – random forest, gradient boosting, support vector machine and multilayer perception - were employed. Five-fold cross validation was used on the training set for hyperparameter tuning. The best classifier with hyperparameters that led to the highest cross-validation accuracy were selected to train the final models on the entire training set. Analyses were conducted using the Scikit-learn package (Pedregosa et al., 2011).

Fine-tuning approach We fine-tuned the base version of the pretrained language model RoBERTa (Liu et al., 2019) for classification. Batch size was fixed at 16. Five-fold cross validation was used on the training set for tuning hyperparameters, including the learning rate (in the range of $5e-4$, $1e-4$, $5e-5$, $1e-5$, $5e-6$, $1e-6$) and the number of epochs (from 2 to 5). Again, hyperparameters that led to the highest cross-validation accuracy were selected to train the final models on the entire training set. All fine-tuning was conducted using the transformers (Wolf et al., 2020) and PyTorch libraries(Paszke et al., 2019).

Ensemble Using the combined Human + GPT-3.5 + GPT-4 training set is one way to help detectors learn that essays generated by either GPT-3.5 or GPT-4 are AI-generated. An alternative way is to ensemble the detectors trained separately on Human + GPT-3.5 and Human + GPT-4 training sets. We obtained two additional detectors – one for the ML approach and another for the fine-tuning approach – by ensembling the predictions from the respective GPT-3.5 and GPT-4 detectors. Note that ensembling happened at inference time, without additional model training. For each essays to be classified, we averaged the predicted probabilities from the GPT-3.5 and GPT-4 detectors. If the resulting ensemble probability was higher than 0.5, the essays was classified as AI-generated (label = 1).

2.4 Detector Evaluation

A total of eight detectors were applied to the test sets for evaluation. We can organize these detectors into four conditions, each comprising two detectors trained using either a feature-based ML approach or a fine-tuned RoBERTa model. In the first two conditions, detectors were trained on AI-generated essays produced by only one LLM, either GPT-3.5 or GPT-4. The third condition used the combined Human + GPT-3.5 + GPT-4 training set. The fourth condition involved the ensembled detectors.

We used the number of correctly and falsely classified essays in each of the 1,000 human-written, GPT-3.5-generated, and GPT-4-generated essays as evaluation metrics. Given the goal of detecting AI-generated essays, the number of human-written essays that were misclassified as AI-generated essays are false positives, and the number of AI-generated essays misclassified as human-written essays are false negatives. We used frequencies as evaluation metrics instead of accuracy, precision and recall, which are affected by the ratio between human-written essays and AI-generated essays in the test set. This is because in our test set, the composition of human-written and AI-generated essays is 1:2, which is unlikely in real settings.

3 Results

3.1 Essay Similarities

We first examined pairwise text similarities among essays for each item, because the extent of similarities among human-written, GPT-3.5-generated, and GPT-4 generated essays can affect detector performances. Per item, within the same generation source (i.e., human-written, GPT-3.5, and GPT-4), the number of pairs was $n_k(n_k-1)/2$, where n_k is the number of essays in source k for this item. Across sources, the number of pairs was $n_i n_j/2$, where n_i and n_j are the number of essays for sources i and j. The number of pairs for GPT-3.5-generated, and GPT-4-generated essays was 897,000; for human-written essays was 834,074, for GPT-3.5-generated and GPT-4-generated essays for 1,800,000, for GPT-3.5/4-generated and human-written essays was 1,723,500. For each pair of essays, we calculated the cosine similarity of trigram term frequency-inverse document frequency (TF-IDF) vectors, and the edit similarity (Navarro, 2001). Both similarity measures are within 0 to 1, with a higher number indicating higher similarities. Box plots of pairwise similarities of essays for the same items within and between sources are provided in Figure 1. Sources with higher median similarities are located higher on the y-axis.

Essay similarity results revealed differences between GPT-3.5- and GPT-4-generated essays. Within the same source, essays generated by GPT-3.5 were the most similar as each others, while GPT-4 was able to generate essays with higher text variability, but not as diverse as human-written essays. Across sources, essays generated by the two LLMs were more similar to each other than with human-written essays. Human-written essays were more similar with GPT-3.5-generated essays than with GPT-4-generated essays.

3.2 Detector Performances

Eight detectors were applied to the three test sets consisting of respectively 1,000 human-written, GPT-3.5-generated and GPT-4-generated essays. The number of essays correctly and wrongly classified as human-written or AI-generated on the test sets by each detector are reported in Table 1. Among all the studied ML classifiers, SVM yielded the highest cross-validation accuracy in the three training sets. Thus, detectors obtained using SVM were used to represent the ML approach. When building detectors by fine-tuning RoBERTa, the following hyperparameters led to the highest cross-validation accuracy respectively for the three training sets, $lr = 5e-5$ and $epoch = 4$, $lr = 5e-5$ and $epoch = 5$, and $lr = 1e-5$ and $epoch = 4$.

First focus on detectors trained with AI essays generated by only one LLM (i.e., conditions 1 and 2 in Table 1). In the columns for human-written essays, we see that fine-tuned RoBERTa misclassified fewer number of human-written essays as AI-generated essays than SVM. While SVM misclassified 22 and 32 human-written essays as AI-generated essays, detectors based on fine-tuned RoBERTa misclassified fewer than 5 essays. For AI-generated essays that were generated by the same LLM as used in the training set (i.e., column GPT-3.5-generated for condition 1, and column GPT-4-generated for condition 2), detectors based on fine-tuned RoBERTa correctly classified all AI generated essays, while SVM missed 24 and 15 AI-generated essays. However, when detectors trained with AI-generated essays by one LLM were applied to essays generated by the other LLM (i.e., column GPT-4-generated for condition 1, and column GPT-3.5-generated for condition 2), the number of false negative cases increased. Fine-tuned RoBERTa trained with GPT-3-generated essays failed to identify 84 GPT-4-generated essays and fine-tuned RoBERTa trained with GPT-4 generated essays GPT-4 missed 192 GPT-3.5-generated essays. Performances of SVM detectors were worse. They failed to identify respectively 522 and 370 essays generated by the other LLM.

When both GPT-3.5- and GPT-4-generated essays were included in the training set (i.e., condition 3 in Table 1), the resulting detectors had lower number of false negatives cases for the combination of 1,000 GPT-3-generated and 1,000 GPT-4-generated essays. Fine-tuned RoBERTa identified all GPT-3.5 generated essays and only missed one GPT-4 generated essay, while SVM missed 29 GPT-3.5-generated and 26 GPT-4 generated. In terms of false positives, fine-tuned RoBERTa misclassified 9 out of the 1000 human-written essays (0.9%) as AI-generated essays, while SVM misclassified 41 (4%). These number of false positives were slightly higher than those for detectors trained with AI essays generated using only one model (i.e., conditions 1 and 2).

When GPT-3.5 detector and GPT-4 detector were ensembled (condition 4 in Table 1), the ensembled fine-tuned RoBERTa detector only misclassified 1 human-written essays as AI-generated essays and

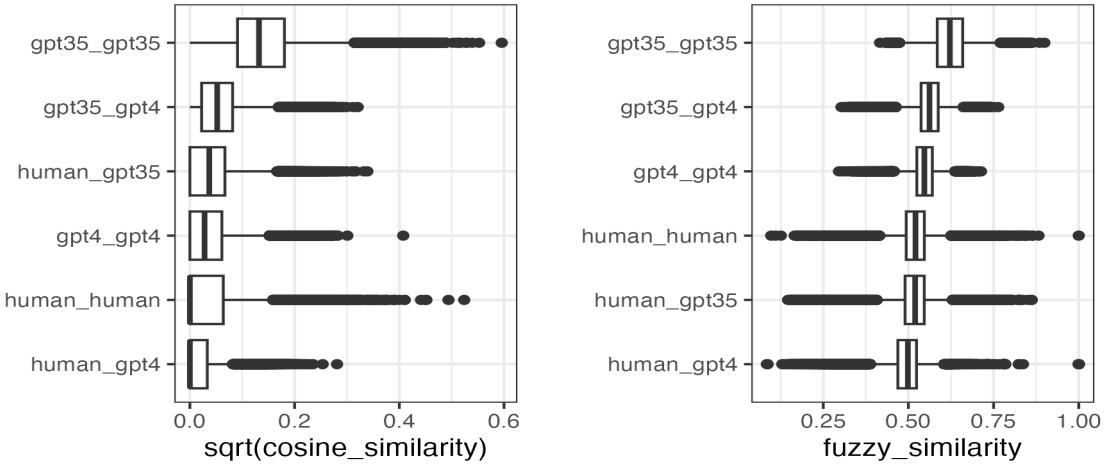

Figure 1: Box plots of the square root of cosine similarity and edit similarity among essays for the same items within and between sources.

Table 1: Number of Correctly and Falsely Labeled Essays in Test Sets by Different Detectors

Training condition	Approach	Human-written (n=1000)		GPT-3.5-generated (n=1000)		GPT-4-generated (n=1000)	
		Correct	Wrong	Correct	Wrong	Correct	Wrong
1. Human + GPT-3.5	SVM	978	22	976	24	478	522
	RoBERTa	996	4	1000	0	916	84
2. Human + GPT-4	SVM	968	32	630	370	985	15
	RoBERTa	997	3	808	192	1000	0
3. Human + GPT-3.5 + GPT-4	SVM	959	41	971	29	974	26
	RoBERTa	991	9	1000	0	999	1
4. Ensemble	SVM	986	14	894	106	928	72
	RoBERTa	999	1	993	7	993	7

failed to identify 7 GPT-3.5-generated and also 7 GPT-4 generated essays. Ensembled SVM misclassfied 14 human-written essays as AI-generated essays, but missed 106 GPT-3.5-generated and 72 GPT-4 generated. Comparing the conditions 3 and 4, in which detectors were given the information that both GPT-3.5- and GPT-4-generated essays are AI-generated, the ensembled detectors had lower number of false positives and higher number false negatives.

4 Discussion

In this study, we investigated the effects of generation model on performances of detectors for AI-generated essays. We studied two generation models (GPT-3.5 and GPT-4), two training approaches (feature-based ML and fine-tuning), and two ways of providing information from both generation models (including essays generated by both LLMs in the training set and ensembling detectors trained with only one LLM for essay generation). We found that a) fine-tuned detectors outperformed feature-based ML detectors on all studied metrics; b) compared to detectors trained with essays generated from both models, those trained with essays generated from only one model did not misclassify more human-written essays as AI-generated (false positives), but did misclassify more essays generated by the other model as human-written essays (false negatives); c) the ensembled fine-tuned RoBERTa detector had fewer false positives, but slightly more false negatives comparing to detectors trained with essays generated by both GPT-3.5 and GPT-4.

Fine-tuning pre-trained large language models has been found to be effective for many classification tasks, including natural language inference (Devlin et al., 2019), automated essay scoring (Fernandez et al., 2023), and AI-generated essay detection (Kaggle Community, 2025). Our findings are inline with these previous findings, suggesting superior performances of the fine-tuning approach comparing to the feature-based ML approach for AI-generated essay detection. However, the complexity of LLMs makes it difficult to explain the predicted results from fine-tuned LLMs. This posts challenges of using fine-tuned detectors in high-stakes situations, where false accusations against individuals can have serious consequences. Research to identify tokens or phrases that affecting the fine-tuned detectors' decisions, or the effects of adversarial inputs can be important future directions.

To detect essays generated by a wide range of AI models, the natural choice is to train a detector using essays generated by a diverse number of AI models. However, it can be resource-intensive to re-train the detector each time a new AI model is released. If the number of human-written essays don't increase, creating a balanced training set may mean not include all the AI-generated essays from previous AI models for training. This is the scenario we studied. Even though we generated 5,000 GPT-3.5-generated essays in conditions 1, and 5,000 GPT-4-generated essays in condition 2, we only used 2,500 from each generation model in condition 3. We found ensembling fine-tuned RoBERTa can be an effective alternative. It allows the use of the same number of AI-generated essays for each generation model as the number of human-written essays. Once detector is built for each generation model, one can flexibly adjust the contribution from each detector at inference, if there is evidence on the likelihood of essays from each generation model. Ensemble also allows easy adjustment of threshold. For example, if reducing false positives is more important, one may adjust the threshold to higher than .5.

5 Limitations

In this study, we generated essays using two AI models, built detectors with balanced sets of human-written and AI-generated essays, and studied detector performance in terms of detection accuracy. Results need to be generalized with caution beyond these conditions. As noted in the introduction, in the real-world, AI can be used in creating essays in many different ways. Other models that are more distant from the models in the OpenAI family, such as LLaMA or DeepSeek-R1, may produce more different essays, thus affect the detection performance. Essays may also be created by both humans and AI, with only a portion of the text generated by AI or humans revise AI-drafted essays. Moreover, fairness in detection across demographic groups is also an importance metric for evaluating detector performance. For future work, we plan to expand the study by including a broader range of generation models and also varying the proportion of AI generated text within essays.

References

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v.2. *The Journal of Technology, Learning, and Assessment*, 4(3):3–30.

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.

Nigel Fernandez, Aritra Ghosh, Naiming Liu, Zichao Wang, Benoît Choffin, Richard Baraniuk, and Andrew Lan. 2023. Automated scoring for reading comprehension via in-context bert tuning. *Preprint*, arXiv:2205.09864.

GPTZero. 2025. Gptzero - ai detector - the original ai checker for chatgpt & more. Accessed: 2025-06-18.

Grammarly Inc. 2025. Grammarly ai detector. Accessed: 2025-06-18.

Yang Jiang, Jiangang Hao, Michael Fauss, and Chen Li. 2024. Detecting ChatGPT-generated essays in a large-scale writing assessment: Is there a bias against non-native English speakers? *Computers & Education*, 217:105070.

Kaggle Community. 2025. LLM - Detect AI Generated Text: Discussion Post. https://www.kaggle.com/competitions/llm-detect-ai-generated-text/discussion/473295. Accessed: 2025-06-20.

Ranvijay Kumar. 2022. A python package to simulate typographical errors in english language. https://github.com/ranvijaykumar/typo. Accessed: 2025-06-20.

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.

Brady D. Lund, Tae Hee Lee, Nishith Reddy Mannuru, and Nikhila Arutla. 2025. Ai and academic integrity: Exploring student perceptions and implications for higher education. *Journal of Academic Ethics*.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Scribbr. 2025. Ai detector - trusted ai checker for chatgpt, copilot & gemini. Accessed: 2025-06-18.

U.S. Department of Education, Office of Educational Technology. 2023. Artificial intelligence and the future of teaching and learning: Insights and recommendations.

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.

Duanli Yan, Michael Fauss, Jiangang Hao, and Wenju Cui. 2023. Detection of AI-generated essays in writing assessment. *Psychological Testing and Assessment Modeling*, 65(2):125–144.

Yang Zhong, Jiangang Hao, Michael Fauss, Chen Li, and Yuan Wang. 2024. Evaluating ai-generated essays with gre analytical writing assessment. *Preprint*, arXiv:2410.17439.

Exploring the Interpretability of AI-Generated Response Detection with Probing

Ikkyu Choi
ETS, Princeton, NJ.
ichoi001@ets.org

Jiyun Zu
ETS, Princeton, NJ.
jzu@ets.org

Abstract

Multiple strategies for AI-generated response detection have been proposed, with many high-performing ones built on language models. However, the decision-making processes of these detectors remain largely opaque. We addressed this knowledge gap by fine-tuning a language model for the detection task and applying probing techniques using adversarial examples. Our adversarial probing analysis revealed that the fine-tuned model relied heavily on a narrow set of lexical cues in making the classification decision. These findings underscore the importance of interpretability in AI-generated response detectors and highlight the value of adversarial probing as a tool for exploring model interpretability.

1 Introduction

Modern foundation language models have demonstrated the ability to generate coherent, well-structured text across a wide range of domains (Li et al., 2024; Zhao et al., 2023). This capability has affected various aspects of writing, including assignments and assessments that require learners to write original responses. As a result, educators and assessment professionals have become increasingly interested in distinguishing between human-written and AI-generated responses (Jiang et al., 2024). This task, which we refer to as AI-generated response detection in this paper, is the focus of our study.

The growing interest in, and demand for, AI-generated response detection has led to the development of algorithmic detectors, many of which are themselves based on language models. Some utilize the in-context learning capability of these models combined with prompt engineering, while others employ supervised fine-tuning on custom datasets designed for the detection task (see, e.g., Fraser et al., 2025 and Wu et al., 2025). These detectors are often marketed as highly accurate, with reported classification accuracy routinely exceeding 90 percent. However, similar to other inferences made by language models, the decisions from these detectors are opaque and difficult to interpret. This lack of transparency is particularly concerning in AI-generated response detection, in which learners may face serious consequences (e.g., academic penalties, score cancellations) based on the detection outcome.

To address this knowledge gap, we investigated the decision-making process of a custom AI-generated response detector using probing techniques. Probing has proven effective in examining the internal mechanisms of language models across a variety of downstream tasks (Li et al., 2023a; Niven and Kao, 2019; Ohmer et al., 2024; Suau et al., 2020). In this study, we fine-tuned an open-source language model on a dataset including both human-written and AI-generated responses. We then identified and manipulated lexical cues to gauge their influence on the model's classification decisions.

Our findings show that the fine-tuned model achieved high accuracy in the detection task by relying heavily on a small set of lexical cues. While this reliance demonstrates the expressive capacity of language models, it also exposes their vulnerability to exploitation and manipulation. Overall, our results substantiate the need for understanding what AI-generated response detectors learn and for evaluating the trustworthiness of their decisions in real-world applications.

2 Background

Although the language generation capabilities of modern foundation models have provided opportunities and benefits, they also pose risks that can lead to undesirable outcomes. Crothers et al. (2023) proposed a taxonomy that classifies these into four high-level categories: (1) spam and harassment, (2)

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 99–106

online influence campaigns, (3) malware and social engineering, and (4) AI authorship exploitation. For our study, a particularly relevant form of AI authorship exploitation is academic fraud committed by learners and examinees. They may undermine the learning and assessment purposes of writing tasks by submitting responses that are generated by AI models.

To mitigate the authorship exploitation risk, researchers have explored various detection strategies and their effectiveness. A consistent finding across studies is that humans find it difficult to reliably detect AI-generated text. Multiple investigations have shown that human judges, including domain experts, often perform at near-chance levels when attempting to distinguish AI-generated text from human-written one (e.g., Li et al., 2023b; Soni and Wade, 2023; Uchendu et al., 2021), although training (Liu et al., 2023) and auxiliary information (Gehrmann et al., 2019) may improve their detection performance. The difficulty of manual detection, combined with the scalability of AI-generated text, has led researchers to algorithmic approaches. This pursuit has quickly accumulated into a sizable body of literature, for which multiple comprehensive surveys are available (e.g., Beresneva, 2016; Crothers et al., 2023; Dhaini et al., 2023; Jawahar et al., 2020; Fraser et al., 2025; Wu et al., 2025).

Wu et al. (2025) and Fraser et al. (2025) classified algorithmic detectors into three main categories based on the type of information leveraged: watermarks, manually engineered features, and language model-based text representations. The third category uses numerical embeddings derived from foundation language models as implicit features for classification; this allows researchers to circumvent the need for watermarks or manual feature development. Detectors based on this approach, particularly those that relied on fine-tuned language models, have demonstrated strong performance, with detection accuracies often exceeding 90% across diverse text types (e.g., Chen et al., 2023; Fagni et al., 2021; Guo et al., 2023; Wang et al., 2023). However, the complexity of their architecture makes it difficult to examine their decision making processes. Although there are various linguistic differences between AI-generated and human-written texts (e.g., Seals and Shalin, 2023), it is unclear whether and how these differences are utilized by classifiers.

An effective approach for investigating the internal mechanisms of language model classifiers involves the use of probing through adversarial examples: data points that are intentionally perturbed to challenge a model's decision boundaries while preserving the original semantic content. These examples function as diagnostic tools that can help identify the specific cues that language models rely on when making classification decisions. For example, Niven and Kao (2019) demonstrated that high classification performance can be achieved through reliance on superficial word-level statistical patterns alone rather than meaningful linguistic understanding. Their work demonstrated how adversarial probing can reveal vulnerabilities in a model's generalization capabilities and shed light on its interpretability. Subsequent studies have applied adversarial probing to better understand the decision-making processes of language models fine-tuned for a range of classification tasks (e.g., Li et al., 2023a; Ohmer et al., 2024; Suau et al., 2020).

In the domain of AI-generated text detection, adversarial examples have also been used to evaluate the robustness of detection systems. These adversarial "attacks" may operate at varying levels of granularity, including character-level perturbations (e.g., Wang et al., 2024), word-level substitutions (e.g., Pu et al., 2023; Wang et al., 2024), and paraphrasing techniques that maintain semantic meaning while altering surface form (e.g., Shi et al., 2024; Krishna et al., 2023). While these studies have effectively demonstrated the vulnerability of detectors to such attacks, they often focus primarily on evasion rather than on interpretability. As a result, the internal decision-making processes of these detectors remain largely opaque.

3 Methods

3.1 Data

Our dataset included both authentic responses written by human examinees and AI-generated responses. The authentic responses were collected from an essay writing task administered as part of a standardized English language proficiency assessment. In this task, examinees were asked to express their opinion or preference on a given topic, providing supporting details. We used 5,745 authentic responses on across 20 different topics submitted by examinees representing a diverse range of nationalities and first languages. The dataset also included 6,000 responses on the same 20 topics generated by GPT-3.5 (Ouyang et al., 2022) and

GPT-4 (Achiam et al., 2023). These synthetic responses were produced as part of a separate study (Zu et al., 2025), which provides a detailed description of the generation process.

AI-generated text typically lacks typographical errors, whereas such errors are common in human-written ones, including the authentic responses in our dataset. This discrepancy could easily be exploited by detection models, potentially reducing the task to a trivial problem. To address this issue, Zu et al. (2025) randomly imputed typographical errors into each AI-generated response, and we used the generated responses that included these imputed errors.

We allocated approximately 80% of the total dataset (9,396 out of 11,745 responses) for training and the remaining 20% (2,349 responses) for testing. The train-test split involved stratified random sampling, with generation status (authentic vs. AI-generated) as the stratification variable. This ensured that both the training and test sets maintained a similar proportion of generated responses (approximately 51%).

3.2 Fine-Tuning Detector

We fine-tuned the RoBERTa-base model (Liu et al., 2019) as our primary detector of AI-generated responses. The key hyperparameters for fine-tuning included learning rate and training epochs, which were tuned through a two-dimensional grid search using five-fold cross-validation on the training set. We then used the hyperparameter values that led to the best cross-validation performance to fine-tune the RoBERTa base model using the entire training set. More details about the fine-tuning process can be found in Zu et al. (2025).

The choice of RoBERTa-base was primarily motivated by convenience. To examine the robustness of our findings with respect to this model choice, we also fine-tuned three additional models: RoBERTa-large, and two DeBERTa models (He et al., 2021) of different sizes (base and large). These alternative models were fine-tuned using the same procedure as the main detector based on RoBERTa-base.

3.3 Examining n-gram Distributions

To identify linguistic cues that our detector would learn during fine-tuning, we analyzed the n-gram distributions in authentic and AI-generated responses within the training set. For an n-gram to be considered informative, it must satisfy two conditions:

1. It should exhibit a distinct distribution between authentic and generated responses.

2. It should occur with sufficient frequency in the training data.

To quantify these conditions, we adapted the π and ξ statistics introduced by Niven and Kao (2019). Let $\mathcal{A}$ and $\mathcal{G}$ denote the sets of authentic and generated responses in the training set, respectively. Let n_{ui} represent the count of an n-gram u in response i. Using this notation, we formally introduce the two adapted metrics below.

The asymmetry metric, adapted from the π statistic in Niven and Kao (2019), captures the relative difference in frequency of u between generated and authentic responses:

$$\text{Asymmetry}_u = \frac{\sum_{i\in\mathcal{G}} n_{ui} - \sum_{j\in\mathcal{A}} n_{uj}}{\sum_{i\in\mathcal{G}} n_{ui} + \sum_{j\in\mathcal{A}} n_{uj}}.$$

This metric ranges from -1 to 1. The value of -1 indicates that the n-gram appears exclusively in authentic responses. Similarly, the value of 1 indicates exclusive presence in generated responses.

The impact metric, adapted from Niven and Kao's (2019) ξ statistic, measures the average difference in frequency per response:

$$\text{Impact}_u = \frac{\sum_{i\in\mathcal{G}} n_{ui} - \sum_{j\in\mathcal{A}} n_{uj}}{(|\mathcal{G}| + |\mathcal{A}|)/2},$$

where $|\mathcal{G}|$ and $|\mathcal{A}|$ denote the number of generated and authentic responses (in the training set), respectively. The sign of the impact metric aligns with that of the asymmetry metric, indicating the direction of distributional difference.

We analyzed the distributions of unigrams, bigrams, and trigrams in the training set using the asymmetry and impact metrics, with the goal of identifying n-grams exhibiting both high asymmetry and high impact. For the unigram analysis, 129 stop words[1] were excluded. The bigram and trigram analyses were conducted twice: once including the stop words and once excluding them. The identified n-grams were used to construct adversarial examples for probing the behavior of the fine-tuned detector.

[1]We constructed this list by adding `may` and `would` to the 127 stop words from https://gist.github.com/sebleier/554280.

4 Results

4.1 Detector Performance

The fine-tuned RoBERTa-base detector achieved an overall test set accuracy of 0.991, with a precision of 0.983 and a perfect recall of 1.0. This strong performance was robust across different model choices: each of the three alternative fine-tuned detectors achieved similarly high accuracy, precision, and recall. Table 1 presents the confusion matrices for all four fine-tuned detectors. In addition, the detector's performance remained stable under basic text manipulations. For example, converting all characters to lowercase and removing punctuation had minimal impact on accuracy, precision, or recall.

		True Label	
		Aut.	**Gen.**
RoBERTa-base	Aut.	1134	0
	Gen.	21	1194
RoBERTa-large	Aut.	1145	0
	Gen.	10	1194
DeBERTa-base	Aut.	1151	0
	Gen.	4	1194
DeBERTa-large	Aut.	1154	0
	Gen.	1	1194

Table 1: Test set confusion matrices for the main and three alternative fine-tuned detectors. Aut: Authentic; Gen.: Generated

4.2 n-gram Distributions

The results from the unigram, bigram, and trigram analyses showed notable differences in their potential utility as classification cues. The bigram and trigram distributions included only a few sequences that stood out in terms of asymmetry and impact. Moreover, most such bigrams and trigrams were composed primarily of stop words. When stop words were excluded, the same analysis yielded few prominent sequences. The unigram distributions, on the other hand, showed greater potential for distinguishing between authentic and generated responses. While most unigrams in the training set had near-zero asymmetry and impact values, a small subset had large absolute values on one or both metrics, suggesting their potential as strong indicators. This overall pattern is illustrated in Figure 1 as a bivariate scatter plot of unigram asymmetry and impact values. In addition, Table 2 lists the top 10 unigrams in terms of their absolute impact metrics.

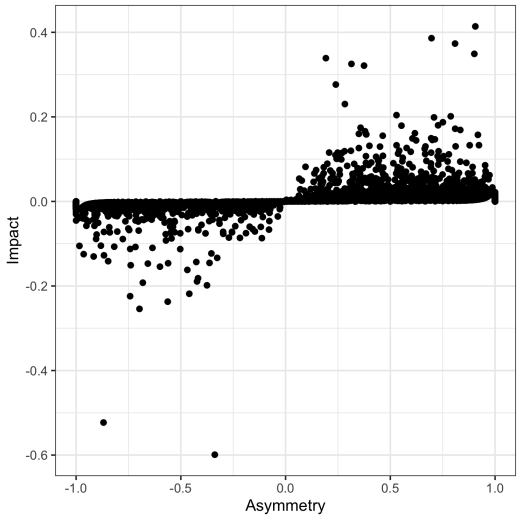

Figure 1: The asymmetry and impact metrics from the training set responses.

Table 2: Asymmetry (Asy.), impact (Imp.), and signal direction (Dir.) of the top 10 unigrams in the training set, in descending order of their absolute impact metrics.

	Asy.	Imp.	Dir.
`people`	-0.338	-0.599	Authentic
`think`	-0.869	-0.523	Authentic
`individuals`	0.906	0.414	Generated
`provide`	0.697	0.386	Generated
`overall`	0.809	0.373	Generated
`additionally`	0.901	0.349	Generated
`skills`	0.314	0.325	Generated
`learning`	0.375	0.321	Generated
`example`	-0.697	-0.254	Authentic
`good`	-0.563	-0.237	Authentic

A key distinction between unigrams associated with authentic versus generated responses was their lexical complexity or sophistication. Words that are long and typically used in formal settings tended to signal generated responses, whereas shorter, more informal ones were more indicative of authentic responses. This pattern manifested in the average length of unigrams signaling the two classes: among the 248 unigrams whose absolute asymmetry and impact values exceeded 0.05, the 72 unigrams signaling authentic responses were on average 5.0 characters long, whereas the corresponding average for the 176 unigrams signaling generated responses was 7.9. This also aligns with our prior expectations that human examinees in timed test-

ing contexts are more likely to produce draft-like responses, which may involve less frequent use of sophisticated and formal vocabulary, and that the training data for GPT 3.5 and GPT-4 are likely to consist primarily of final versions of texts rather than drafts.

4.3 Transforming Test Set Responses into Adversarial Examples

To probe the fine-tuned detector, we transformed test set responses into adversarial examples by replacing unigrams that signaled either authentic or generated responses with synonyms indicative of the opposite class. We focused on unigrams that met two criteria: (1) high absolute values on both asymmetry and impact metrics, and (2) availability of a synonym frequently used in the opposite class. For example, the unigram `people`, which strongly signaled an authentic response, was replaced with `individuals`, a word that appeared much more frequently in generated responses. To ensure meaningful substitutions, we allowed synonyms that were not unigrams, provided they occurred frequently in the opposite category. For instance, `additionally`, which appeared almost exclusively in generated responses, was replaced with `in addition`, a phrase more frequently used in authentic responses.

Applying these criteria to the training set yielded 149 unigrams to be replaced with their respective synonyms. The selected unigrams were a small subset of all unigrams in the training set, which included more than 30,000 unique unigrams. Among the 149 unigram-synonym pairs, 100 involved unigrams signaling generated responses paired with synonyms more frequent in authentic responses, while the remaining 49 involved the reverse pairing. Replacing the 149 unigrams with their synonyms resulted in, on average, 12 substitutions per response, affecting less than 10% of the average unigram count per response. This controlled transformation allowed us to evaluate the classifier's sensitivity to lexical shifts while preserving overall semantic content.

4.4 Detector Performance on Adversarial Examples

The transformation of test set responses into adversarial examples noticeably degraded the performance of the fine-tuned detector. Its overall accuracy dropped from 0.991 to 0.580. This accuracy is only slightly higher than that of a degenerate detector classifying every input into the most frequent category (whose accuracy would have been 0.508). In addition to the decline in overall accuracy, the number of responses classified as generated also dropped from 1,215 (from the original test set) to 210 on the post-transformation adversarial examples. Among those that were classified as generated, all but one response were indeed generated, resulting in a still high precision of 0.995. However, the reduced number of detected responses inevitably led to a sharp reduction in recall, which fell from being perfect (1.0) to extremely low ($209/1,194 = 0.175$), as can be seen the confusion matrix in Table 3.

		True Label	
		Aut.	**Gen.**
RoBERTa-base	Aut.	1154	985
	Gen.	1	209
RoBERTa-large	Aut.	1154	844
	Gen.	1	350
DeBERTa-base	Aut.	1154	898
	Gen.	1	296
DeBERTa-large	Aut.	1155	1097
	Gen.	0	97

Table 3: Confusion matrices for the main and three alternative fine-tuned detectors on the adversarial examples. Aut: Authentic; Gen.: Generated

The substantial decline in the frequency of responses classified as generated indicates that the detector classified much more of the adversarial examples as authentic ones than it did for the original responses. This in turn suggests that the performance change could primarily be attributed to the replacement of the 100 unigrams that were signaling generated responses. To further substantiate this conjecture, we did another transformation of the original test set responses, this time only replacing the 100 such synonym pairs while leaving the other 49 pairs unchanged. The results were quite similar as those from the full transformation involving all 149 unigrams (reported in Table 3), with the overall accuracy of 0.581 and recall of 0.175 as well as the same tendency of classifying only a small number of responses as generated. In contrast, when we did the opposite transformation of only replacing the 49 authentic-signaling synonyms, the results changed little compared to the original results (reported in Table 1): overall accuracy, precision, and recall of 0.966, 0.999, and

0.934, respectively. In sum, the performance declined primarily because the replacement of the 100 unigrams signaling generated responses tricked the detector into classifying generated responses as authentic ones.

These overall results were persistent against model choice. Table 3 also presents the confusion matrices from the three alternative detectors. All show the same pattern of substantial drop in overall accuracy, primarily attributable to the drop in the frequency of responses classified as generated and the accompanying drop in recall. This suggests that all four pre-trained language models mostly picked up unigrams signaling generated responses in the training set during fine-tuning and relied heavily on those unigrams to make their classification decisions.

5 Discussion & Conclusions

In this study, we probed an AI-generated response detector to understand how the model makes its decisions. The detector was built by fine-tuning the RoBERTa-base model (as well as three alternative language models) on a custom dataset, achieving 99.1% accuracy on a held-out test set. To identify influential lexical cues, we analyzed n-gram distributions in the training data and found 149 unigrams strongly associated with either class. By replacing these unigrams with synonyms indicative of the opposite class, we created adversarial test examples that reduced the detector's accuracy from 99.1% to 58.0%. This drop was primarily due to misclassification of AI-generated responses: altering only a small number of unigrams per response was sufficient to cause most AI-generated responses to be misclassified as authentic. The effect was consistent across all tested base models. These findings reveal the detector's strong reliance on a narrow set of lexical cues, which carries both promising and concerning implications.

On the positive side, pre-trained language models effectively identified and leveraged meaningful patterns in unigram distributions during fine-tuning, resulting in high performance on held-out data. Manually identifying these patterns would have been much more difficult and time-consuming. Moreover, such patterns can be used to build more interpretable and explainable classifiers with minimal loss in performance, assuming the patterns remain stable in future data.

However, the ease with which the detector's accuracy was reduced to near-chance levels raises concerns about its generalizability and robustness. If the small set of unigrams signaling AI-generated responses becomes widely known, malicious actors could evade detection by substituting a few words, as demonstrated in our adversarial examples. Therefore, a promising direction for future research is to devise ways to encourage detectors to learn more robust patterns. The identification of this major concern and promising future research step underscore the value of probing fine-tuned detectors in understanding what they learn, evaluating the trustworthiness of their decisions in real-world applications, and guiding improvements where necessary.

We acknowledge that this study was limited in its scope. All detectors were trained on responses from a single task type covering a relatively narrow set of 20 topics. Large-scale writing tests, on the other hand, may include multiple task types and a broader range of topics to ensure topical diversity and coverage. A training dataset drawn from such varied sources may exhibit different characteristics than those observed in our study, and the robustness of detectors trained on more diverse data cannot be reliably inferred from our findings. Furthermore, even within similar training contexts, the rapid evolution of generative AI raises uncertainty about whether the same lexical cues will remain effective indicators of AI-generated content. Therefore, our findings should be interpreted primarily as evidence of what fine-tuned detectors can learn, and how easily they can be compromised, rather than as prescriptive guidance for detection or evasion strategies.

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*.

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *International Conference on Applications of Natural Language to Information Systems*, pages 421–426. Springer.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. GPT-sentinel: Distinguishing human and ChatGPT generated content. *arXiv preprint arXiv:2305.07969*.

Evan N Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002.

Mahdi Dhaini, Wessel Poelman, and Ege Erdogan. 2023. Detecting ChatGPT: A survey of the state of detecting ChatGPT-generated text. *arXiv preprint arXiv:2309.07689*.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. TweepFake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Kathleen C Fraser, Hillary Dawkins, and Svetlana Kiritchenko. 2025. Detecting AI-generated text: Factors influencing detectability with current methods. *Journal of Artificial Intelligence Research*, 82:2233–2278.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

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. *arXiv preprint arXiv:2301.07597*.

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*.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*.

Yang Jiang, Jiangang Hao, Michael Fauss, and Chen Li. 2024. Detecting ChatGPT-generated essays in a large-scale writing assessment: Is there a bias against non-native english speakers? *Computers & Education*, 217:105070.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500.

Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, and 1 others. 2024. Fundamental capabilities of large language models and their applications in domain scenarios: A survey. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11116–11141.

Jiaxuan Li, Lang Yu, and Allyson Ettinger. 2023a. Counterfactual reasoning: Testing language models' understanding of hypothetical scenarios. *arXiv preprint arXiv:2305.16572*.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023b. MAGE: Machine-generated text detection in the wild. *arXiv preprint arXiv:2305.13242*.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023. ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models. *arXiv preprint arXiv:2304.07666*.

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*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*.

Xenia Ohmer, Elia Bruni, and Dieuwke Hupke. 2024. From form(s) to meaning: Probing the semantic depths of language models using multisense consistency. *Computational Linguistics*, 50(4):1507–1556.

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.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE symposium on security and privacy (SP)*, pages 1613–1630. IEEE.

Spenser M Seals and Valerie L Shalin. 2023. Long-form analogies generated by ChatGPT lack human-like psycholinguistic properties. *arXiv preprint arXiv:2306.04537*.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2024. Red teaming language model detectors with language models. *Transactions of the Association for Computational Linguistics*, 12:174–189.

Mayank Soni and Vincent Wade. 2023. Comparing abstractive summaries generated by ChatGPT to real summaries through blinded reviewers and text classification algorithms. *arXiv preprint arXiv:2303.17650*.

Xavier Suau, Luca Zappella, and Nicholas Apostoloff. 2020. Finding experts in transformer models. *arXiv preprint arXiv:2005.07647*.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296*.

Yichen Wang, Shangbin Feng, Abe Bohan Hou, Xiao Pu, Chao Shen, Xiaoming Liu, Yulia Tsvetkov, and Tianxing He. 2024. Stumbling blocks: Stress testing the robustness of machine-generated text detectors under attacks. *arXiv preprint arXiv:2402.11638*.

Zecong Wang, Jiaxi Cheng, Chen Cui, and Chenhao Yu. 2023. Implementing BERT and fine-tuned RoBERTa to detect AI generated news by ChatGPT. *arXiv preprint arXiv:2306.07401*.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Jiyun Zu, Michael Fauss, and Chen Li. 2025. Effects of generation model on detecting AI-generated essays in a writing test. In *Artificial Intelligence in Measurement and Education Conference*. NCME.

A Fairness-Promoting Detection Objective with Applications in AI-Assisted Test Security

Michael Fauss and Ikkyu Choi
ETS Research Institute, Princeton, NJ
{mfauss, choi001}@ets.org

Abstract

A detection objective based on bounded group-wise false alarm rates is proposed to promote fairness in the context of test fraud detection. The paper begins by outlining key aspects and characteristics that distinguish fairness in test security from fairness in other domains and machine learning in general. The proposed detection objective is then introduced, the corresponding optimal detection policy is derived, and the implications of the results are examined in light of the earlier discussion. A numerical example using synthetic data illustrates the proposed detector and compares its properties to those of a standard likelihood ratio test.

1 Introduction

Test security refers to the policies, procedures, and technologies used to protect the integrity and fairness of tests. A key component of test security is *test fraud detection*, that is, detection of unauthorized access to content, tools, or third-party assistance. Statistical methods for test fraud detection have been researched since at least the 1920s (Bird, 1927, 1929), with significant advances happening in the late 20th and early 21st century (Sotaridona and Meijer, 2002; Wollack, 2003; van der Linden and Sotaridona, 2004)—see (Kingston and Clark, 2014; Cizek and Wollack, 2016) for comprehensive overviews. In recent years, however, several developments have significantly expanded both the scope and urgency of test fraud detection efforts:

- The COVID-19 pandemic prompted a sudden shift from testing in tightly controlled test centers to remote testing in environments chosen by the test takers. While this transition offered significant convenience (Zheng et al., 2021; St-Onge et al., 2022), it also introduced numerous new opportunities for cheating (Bilen and Matros, 2021; Janke et al., 2021; Newton and Essex, 2023).
- Generative artificial intelligence (GenAI) models are now powerful enough to solve or assist with a wide range of item types, from simple multiple-choice questions to free-form essays and coding exercises, making them highly effective tools for cheating. (Yan et al., 2023; Susnjak and McIntosh, 2024)
- There is a movement towards more socioculturally responsive (Bennett, 2023) and personalized (Bennett, 2024; Sinharay et al., 2025) assessments to promote fairness and better capture the growing diversity of knowledge and abilities in increasingly heterogeneous test taker populations. This shift has led to greater item variety, resulting in fewer test takers responding to the same items.

These developments have made test fraud detection increasingly challenging: impostors and proxy test takers are more difficult to identify in remote settings than in test centers; AI-generated responses are harder to detect than content copied from traditional sources; and typical response times are difficult to establish for items that have been answered by only a handful of test takers. Consequently, test security reviews tend to require more time, expertise and data than they did in the past.

One approach to addressing these challenges is to delegate tasks to various AI systems, both generative and predictive. Building on the examples above: facial recognition could help detect impostors; typing pattern anomalies could signal proxy test takers; AI-content detectors could identify non-authentic writing or speech; and trained models, rather than empirical distributions, could be used to flag abnormal response times.

However, this approach typically and rightly raises questions regarding the reliability, accuracy and fairness of decision made by AI systems, especially in the context of high-stakes tests. (Weber-Wulff et al., 2023; Perkins et al., 2024) While con-

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 107–114

siderable research is being devoted to making AI fairer, more transparent and more reliable, biases and differential treatment continue to be observed in practice. (Stureborg et al., 2024; Bai et al., 2025; Maslej et al., 2025)

In contrast, many methods traditionally used in psychometrics and, more broadly, decision-making under uncertainty, have transparent objectives and strong accuracy and/or fairness properties. (Dorans and Cook, 2016; Johnson et al., 2022) In this paper, we propose addressing the uncertainty and potential biases of AI outputs not by using them directly, but by feeding them into a system that fuses and processes them. In essence, the idea is to delegate complex subtasks to advanced AIs while anchoring the final decision-making procedure in traditional statistical methods, thereby enabling the use of well-established techniques to define, measure, and promote fairness.

To clarify, this paper does not address the design or architecture of the complete system described above, which remains an ongoing research effort. (Fauss et al., 2025) Instead, it focuses on a specific subtask: designing a detector that flags test takers for potential fraud in a way that balances test integrity and group fairness. This task is formulated and analyzed as a standalone problem, meaning the proposed detector is largely agnostic to the specific detection context. As such, it may be of theoretical or practical interest beyond the use case discussed here. However, as will become clear throughout the paper, its design is explicitly guided by assumptions tailored to the intended application of AI-assisted test fraud detection.

2 Fairness in Test Security

In this section, we discuss some aspects and characteristics that set fairness in test security apart from fairness as a general concept in statistics and machine learning. Specifically, we will make and justify five *claims*. These claims are not intended to be "truths"; rather, we see them as important, sometimes overlooked aspects that can contribute to a more informed discussion of what constitutes fairness in test security applications.

Claim 1: *Fairness and performance are not in conflict.*

A concept commonly encountered in the literature on statistical fairness is the so-called *performance–fairness tradeoff* (Prost et al., 2019), which implies that a procedure's performance and fairness are often in tension with one another. The underlying idea is that in order to make a procedure fairer, additional *constraints* have to be introduced that shrink the space of feasible solutions, and, in turn, reduce the performance. While this is true from a purely mathematical perspective, we would argue that the idea of a performance-fairness tradeoff can be misleading in a test security context. This is the case because detecting test fraud is in itself an objective that, in principle, *promotes fairness*. Among other consequences, widespread, undetected cheating devalues the scores of honest test takers, potentially harming their future opportunities. In general, we consider the idea that a procedure can be "bad" at its dedicated task, yet still perfectly fair problematic. One can even argue that fairness issues are a *consequence* of performance issues. A fraud detector achieving perfect accuracy is not only highly performant, it is also fair by all common criteria. Fairness issues arise once a procedure starts making mistakes, and certain groups are more frequently or more severely affected by these mistakes. Therefore, we argue that in the context of test security fairness and performance should be considered two sides of the same coin—often, a *better* detector will also be a *fairer* detector.

Claim 2: *Equality $\neq$ fairness.*

This claim is closely related to Claim 1. We single it out to highlight the critical role that *equality* plays in virtually all fairness criteria in the literature. For example, *separation fairness* (Barocas et al., 2023) is defined in terms of equal true and false positive rates among all groups. Analogously, *sufficiency fairness* (Barocas et al., 2023) implies that the probability of predicted labels being correct is equal for all groups. Again, we would argue that this idea can be misleading in a test security context. For example, a fraud detector that randomly declare test takers cheaters is perfectly fair by many criteria, yet clearly dysfunctional and unfair in practice. Similarity, by most fairness criteria, a detector with groupwise false alarm rates of, say, $30\,\%$ and $35\,\%$ is fairer than a detector with groupwise false alarm rates of, say, $5\,\%$ and $15\,\%$. In reality, it is far from clear that test takers would view the higher false alarm rate of the first detector as fairer than the larger disparity in groupwise false alarm rates produced by the second.

Claim 3: *Fairness needs a concrete target.*

We argue that any nontrivial measure or intervention aimed at promoting fairness in test security must clearly specify the type of discrimination it

seeks to address and provide strong evidence that it effectively mitigates or eliminates it. While this may seem obvious, our experience suggests it is not consistently implemented in practice. Frequently, existing detectors or classifiers are made fair by picking an arbitrary or convenient fairness criterion, adding a corresponding penalty term to the training objective, and adjusting its weight until a "good performance-fairness tradeoff" is reached. We believe that promoting fairness in this manner can be superficial and ineffective. It will typically lead to a slightly more uniform distribution of the groupwise metric the fairness criterion considers important. However, showing that the combined effects on all groups and on the overall performance really address unfair treatment is usually difficult. In fact, the case for this kind of fairness measure is often made in a circular manner: it promotes fairness *because* it improves the fairness criterion underpinning its design.

Claim 4: *Fairness should not be a black box.*

While Claim 3 argues that it should be clear *what* a fairness-promoting procedure tries to accomplish, here we argue that it should also be clear *how* the procedure promotes fairness. This claim is based on the observation that, in particular in test security, fairness is closely connected to trust and transparency. To clarify, we do not claim that one should be able to explain every technical detail of a fairness-promoting procedure to a non-technical audience. However, we do believe that a sincere attempt at making a procedure fairer should be implemented in way that, at least conceptually, can be communicated to those affected by it. This also opens the door for broader discussions of what constitutes fairness and how it can be improved.

Claim 5: *Fairness should be measurable.*

Naturally, the vast majority of statistical fairness criteria are defined in terms of *probabilities*. However, these probabilities are typically unknown and must be *estimated* from data. This can lead to problems when certain events occur so infrequently that reliably assigning them an empirical probability becomes infeasible. This problem is more prominent the smaller the population and the more groups are considered. For example, in the context of test fraud detection, a fairness criterion that incorporates groupwise cheating rates might run into the problem that for some groups no cheaters have been observed yet. Does this mean that the respective cheating rates are low? Or that the detection rates are low? Can, often self-declared, group variables of cheaters be trusted in the first place? In a nutshell, we argue that fairness should be based on quantities that can accurately and reliably be inferred from the data.

In the next section, we present a fairness promoting detection objective that is informed by and largely aligned with the above claims.

3 A Fairness-Promoting Detection Objective

In this section, we propose a fairness-promoting detection objective, derive the corresponding optimal detector, and discuss its properties in light of the claims in Section 2. While the intended use case of the proposed detector is test fraud detection, it is not limited to this context and likely has applications in other areas.

A quick note on notation: In what follows, uppercase letters, X, denote random variables, lowercase letters, x, denote their realizations, and boldface, $\boldsymbol{x}$, indicates vectors. Probability distributions are denoted by P, and probability density functions (PDFs) by p.

3.1 Problem Formulation

Let $N \in \mathbb{N}_{\geq 1}$ be the number of test takers. For every test taker we observe a random vector $\boldsymbol{X}_n \in \mathbb{R}^M$, $M \in \mathbb{N}_{\geq 1}$, which is a collection of relevant observations and features. In this paper, we do not make further assumptions about the nature or meaning of $\boldsymbol{X}$ or its elements. However, as discussed above, in the intended application of (AI-assisted) test fraud detection, $\boldsymbol{X}$ is assumed to consist of high-level features that themselves are outputs of AI systems (likelihood of AI-generated content, likelihood of copy-typing, likelihood of impostor, etc.).

In addition to the feature vector, we assume that a discrete random variable, $G_n \in \{1, \ldots, N_G\}$, $N_G \in \mathbb{N}_{\geq 1}$ is observed for every test taker indicating membership in one of N_G groups. Every test taker is assumed to belong to exactly one group. These groups are typically defined by demographic attributes such as gender, race, age, or first language. However, depending on the application, one might also consider externally defined groups, such as test takers receiving a certain form or taking the test remotely versus in a test center.

Finally, we assume that every test taker is either fraudulent ("cheater") or honest ("non-cheater"). This is indicated by a binary random variable $C_n \in \{0, 1\}$, with $C_n = 1$ indicating a cheater

and $C_n = 0$ indicating a non-cheater. Naturally, C_n is assumed to be a latent variable.

Finally, we assume that the feature vectors of all test takers are independent, conditioned on their group membership and honesty. That is, there are random variables $\boldsymbol{X}$, G and C such that

$$\boldsymbol{X}_n \,|\, (G_n = g, C_n = c) \overset{d}{=} \boldsymbol{X} \,|\, (G = g, C = c)$$

for all $n \leq N$, where $\overset{d}{=}$ denotes equality in distribution. Therefore, the index n is omitted in what follows. The assumption may not always hold in practice, but it offers a useful approximation that suffices for the discussion at hand.

The detector we seek to design is assumed to generate a random variable $\hat{C} \in \{0, 1\}$ that indicates whether the respective test taker is classified as cheater ($\hat{C} = 1$) or non-cheater ($\hat{C} = 0$). It is defined by a function $f \colon \mathbb{R}^M \to [0, 1]$ that maps a feature vector to a probability of classifying the corresponding test taker as a cheater, that is:

$$P(\hat{C} = 1 \,|\, \boldsymbol{X} = \boldsymbol{x}, G = g, C = c) = f(\boldsymbol{x}). \quad (1)$$

for all $\boldsymbol{x}$, g and c. Note that f is a function only of the feature vector, $\boldsymbol{x}$, but not of the group variable, g, even though g is known. This is intentional, as incorporating group information into a detector is generally considered problematic. Most importantly, it can lead to cases in which two test takers with identical feature vectors are classified differently depending on which group they belong to.

We next present the proposed detection objective:

$$\max_f \; P(\hat{C} = 1 \,|\, C = 1) \quad \text{s.t.} \quad (2)$$

$$P(\hat{C} = 1 \,|\, G = g, C = 0) \leq \alpha \quad \forall g \leq N_G, \quad (3)$$

where $\alpha \in (0, 1)$ is a free parameter. The constraints in (3) enforce an upper bound on the false alarm rate (FAR) of each group. We refer to a detector that satisfies these constraints as fair in the sense of bounded FARs, or *BFAR-fair* for short. For a given α, the objective in (2) picks the BFAR-fair detector with the highest detection rate. This problem formulation will be discussed and justified in more detail shortly.

3.2 Optimal Detector

The main result of this paper, a detector that is optimal in the sense of BFAR fairness, is stated in the following theorem:

Theorem 1. *The detector that solves the problem in (2) and (3) is given by*

$$\hat{C}^* = \begin{cases} 0, & g_{\boldsymbol{\lambda}^*}(\boldsymbol{x}) \leq 0 \\ 1, & g_{\boldsymbol{\lambda}^*}(\boldsymbol{x}) > 0 \end{cases}, \quad (4)$$

where

$$g_{\boldsymbol{\lambda}}(\boldsymbol{x}) = p(\boldsymbol{x} \,|\, C = 1) - \sum_{g=1}^{N_G} \lambda_g p(\boldsymbol{x} \,|\, G = g, C = 0) \quad (5)$$

and $\boldsymbol{\lambda}^*$ is such that

$$P[\hat{C}^* = 1 \,|\, G = g, C = 0] = \alpha \quad (6)$$

if $\lambda_g^* > 0$ and

$$P[\hat{C}^* = 1 \,|\, G = g, C = 0] < \alpha \quad (7)$$

if $\lambda_g^* = 0$.

Proof. The statement in the theorem can be proven using standard arguments in constrained optimization. The Lagrange dual (Boyd and Vandenberghe, 2004, Ch. 5.2) of the problem in (2) is given by

$$\min_{\boldsymbol{\lambda} \geq 0} \; \max_f \; L_\alpha(f, \boldsymbol{\lambda}), \quad (8)$$

where

$$L_\alpha(f, \boldsymbol{\lambda}) = P[\hat{C} = 1 \,|\, C = 1] - \sum_{g=1}^{N_G} \lambda_g P[\hat{C} = 1 \,|\, G = g, C = 0] + \sum_{g=1}^{N_G} \lambda_g \alpha.$$

By conditioning and marginalizing over $\boldsymbol{X}$ we can write L_α as

$$L_\alpha(f, \boldsymbol{\lambda}) = \int f(\boldsymbol{x}) g(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \alpha \sum_{g=1}^{N_G} \lambda_g, \quad (9)$$

where $g_{\boldsymbol{\lambda}}$ is defined in (5) and we used (1) to write the relevant probabilities in terms of f. Since L_α in (9) is linear in f, the maximizer of the inner problem in (8) is given by

$$f^*(\boldsymbol{x}) = \begin{cases} 0, & g_{\boldsymbol{\lambda}}(\boldsymbol{x}) \leq 0 \\ 1, & g_{\boldsymbol{\lambda}}(\boldsymbol{x}) > 0 \end{cases}. \quad (10)$$

It remains to show that the optimal Lagrange multiplier satisfy (6) and (7). However, this property follows immediately from the complementary slackness condition of the KKT conditions (Boyd and Vandenberghe, 2004, Ch. 5.5). Finally, note that for $f = f^*$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$ the constraints in (3) are satisfied by construction, which in turn implies that the solution of the dual problem also solves the primal problem. (Boyd and Vandenberghe, 2004, Ch. 5.5) This completes the proof. □

3.3 Discussion

In this section, we discuss the problem formulation in (2) and (3) in more detail and explain why we consider BFAR fairness an appropriate and practical approach to promoting fairness in the context of (AI-assisted) test fraud detection.

1. BFAR fairness requires the detector to operate at a false alarm rate (type II error probability) below α for all groups. This means that, in the spirit of Claim 1, there is a *minimum performance level* that the detector needs to meet in order to be considered fair.

2. In the spirit of Claim 2, BFAR fairness *promotes* equality, but does not *enforce* it. As long as the error probabilities are acceptable for all groups, it does not detract from the detector's fairness if it performs better for some groups.

3. BFAR fairness deliberately constraints false alarm rates instead of alternative metrics, such as false discover rates or detection rates. This is in the spirit of Claims 3 and 4. BFAR fairness targets unfairness from the perspective of *honest test takers* and, consequently, can be communicated in a straightforward manner: For an honest taker, the probability of being falsely flagged by a BFAR-fair detector is at most α, irrespective of their race/age/first language etc. Appropriate values of α might be subject to debate, but we believe that both the target group and the concept of BFAR fairness are clear and transparent.

4. BFAR fairness does not require groupwise detection rates. This is in the spirit of Claim 5. For any reputable test, cheaters are a small minority of the test taker population. Therefore, as explained in the discussion of Claim 5, estimating groupwise detection rates is notoriously difficult for smaller groups. Moreover, groups can sometimes lose their meaning if the corresponding test taker committed fraud. For example, a native French speaker might copy an essay written by a native Mandarin speaker. Therefore, BFAR fairness avoids grouping cheaters in the first place.

5. By inspection of (4) and (5), the BFAR-fair detector is implemented via a modified likelihood ratio test. More specifically, it compares the likelihood of the observed feature vector under the cheater versus the honest hypothesis. However, while a standard likelihood ratio test marginalizes over the group variables using their true probabilities, the marginalization in the BFAR-fair test statistic in (5) is performed with custom weights, $\boldsymbol{\lambda}^*$, that do not necessarily reflect the actual group sizes. That is, the BFAR-fair detector is implemented by *re-weighting* or *oversampling* groups that would otherwise violate the false alarm rate constraints. While details on how to obtain these weights and how they enter the test statistic may be more intricate, the underlying idea of re-weighting or oversampling is well-established, conceptually simple, and easy to communicate—which aligns well with the spirit of Claim 4.

However, BFAR fairness also has its shortcomings. For example, two arguments against its use in operation are the following:

1. The detection rate and groupwise false alarm rates can not be observed directly, but have to be inferred based on some statistical model of the test taker population. This aspect can be argued to be in conflict with Claim 5. However, as discussed above, the quantities of interest were deliberately chosen to avoid problematic corner cases or small-sample scenarios, and we expect that they can typically be estimated with reasonable accuracy.

2. The focus on honest test takers can conflict with the goal of test integrity. Traditionally, fraud detectors are tuned to meet specific detection rate requirements, accepting potentially high false alarm rates as a necessary cost. From the BFAR perspective, one first determines a justifiable burden on honest test takers and then accepts the corresponding detection rates. On the one hand, this approach can be difficult to defend in practice. On the other hand, in the spirit of Claim 1, any detector that can only satisfy integrity standards by imposing an unacceptable burden on honest test takers may not be ready for operational use.

In summary, while BFAR fairness may not be suitable or implementable in every setting and application, we believe that it is a useful, transparent, practical and well-justified approach to promoting fairness in test fraud detection.

Table 1: Groupwise false alarm rates of likelihood ratio and BFAR-fair detector with detection rate of $\approx 87\,\%$.

	False Alarm Rate		
Detector	$G=1$	$G=2$	$G=3$
Likelihood Ratio	0.0416	0.1298	0.0825
BFAR-fair	0.0835	0.1011	0.0997

4 Numerical Example

In this section, we demonstrate the BFAR fair detector proposed in the previous section with a numerical example. Since it is merely supposed to provide a proof of concept, we deliberately keep this example simple. Specifically, we assume that the test taker population consists of three equally likely groups of interest ($N_G = 3$) and that two features ($M = 2$) are observed for each test taker. In line with the assumption that these features are themselves probabilities of a test taker having committed fraud, likely generated by large, high-level AI models, we assume that the feature vectors are distributed on the unit square. We model these features via a multivariate beta distribution in (Fauss, 2024). The exact parameters for each group are given in Appendix A.

In order to establish a baseline performance, and in light of Comment 5 in Section 3.3, we compare the proposed BFAR-fair detector to a standard likelihood ratio test, that is, a detector with decision rule

$$\hat{C} = \begin{cases} 1, & \frac{p(\boldsymbol{x}\,|\,C=1)}{p(\boldsymbol{x}\,|\,C=0)} \geq \nu \\ 0, & \frac{p(\boldsymbol{x}\,|\,C=1)}{p(\boldsymbol{x}\,|\,C=0)} < \nu \end{cases}, \tag{11}$$

where $\nu \in (0, 1)$ is a threshold that balances the detection and false alarm rates.

We set the parameter of the BFAR-fair test to $\alpha = 0.1$, that is, the false alarm rate must not exceed $10\,\%$ for any group. The corresponding weights, $\boldsymbol{\lambda}^*$, were determined by numerically solving the optimality conditions in Theorem 1 and are given by $\boldsymbol{\lambda}^* \approx (0, 0.7682, 0.4266)$. The probabilities on the left-hand sides of (6) and (7) were approximated by sampling from the specified distributions. The threshold ν was selected so that the detection rate of the likelihood ratio detector matches that of the BFAR-fair detector, which was evaluated to $87\,\%$ in this case. Again, we used sampling to approximate this rate. The resulting groupwise false alarm rates for both detectors are reported in Table 1.

By inspection, the false alarm rates of the likelihood ratio detector vary substantially across groups, ranging from just above $4\,\%$ for group 1 to nearly $13\,\%$ for group 2. In contrast, by design, the BFAR-fair detector keeps all false alarm rates below the $10\,\%$ threshold. Note that while the false alarm rates for groups 2 and 3 are close to this threshold, the rate for group 1 is lower by a margin that cannot be attributed to approximation errors alone. This gap is consistent with the first element of $\boldsymbol{\lambda}^*$ being zero, which indicates that the false alarm rate constraint for group 1 is non-binding. In fact, the BFAR detector uses effective group probabilities/sizes of $P(G=1) = 0$, $P(G=2) \approx 0.64$ and $P(G=3) \approx 0.26$. In words, the assumed probability of group 3 remains close to its true value of $\frac{1}{3}$, the probability of group 2 approximately doubles, increasing its influence on the test statistic, while the effective size of group 1 set to zero, effectively ignoring it in the calculation of the non-cheater likelihood. This implies that the false alarm rate constraint for group 1 is redundant given the constraints for groups 2 and 3.

The decision boundaries of the two detectors are shown in Figure 1. For illustration purposes, Figure 1 also shows samples of feature vectors drawn from the respective distributions. Both decision boundaries approximately follow the negative diagonal of the unit square, with a noticeable "bulge" in the region where the feature distribution of honest test takers in group 2 strongly overlaps with that of the cheaters. However, the bulge is much more pronounced in case of the BFAR-fair detector. This increased lenience towards test takers in group 2 is (partially) compensated by tightening the decision boundary in the upper left region, which is unlikely to contain members of group 2. This adjustment explains the observed increase in false alarm rates for test takers in groups 1 and 3.

In summary, at the same detection rate, the BFAR-fair detector admits a significantly more uniform false alarm rate profile compared to a standard likelihood ratio test and keeps the "worst case" false alarm rate across all groups below the targeted $10\,\%$. On the downside, the overall false alarm rate, which, in this case, is given by the average of the groupwise rates, increases from $8.5\,\%$ to $9.5\,\%$. Whether or not this drawback outweighs the benefits of the BFAR-fair detector has to be evaluated on a case-by-case basis. We hope that the discussions in Section 2 and 3.3 provide valuable guidelines for this evaluation.

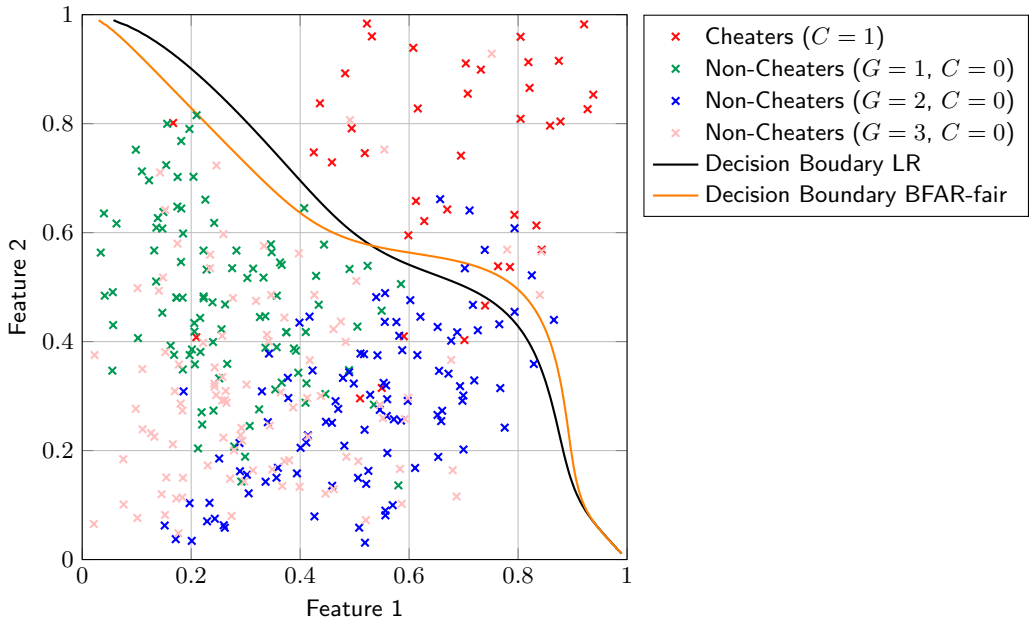

Figure 1: Feature sample and decision boundaries of BFAR-fair and likelihood ratio detector.

References

Xuechunzi Bai, Angelina Wang, Ilia Sucholutsky, and Thomas L. Griffiths. 2025. Explicitly unbiased large language models still form biased associations. *Proceedings of the National Academy of Sciences*, 122(8):e2416228122.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2023. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, Cambridge, MS, USA.

Randy E. Bennett. 2023. Toward a theory of sociocultur-ally responsive assessment. *Educational Assessment*, 28(2):83–104.

Randy E. Bennett. 2024. Personalizing assessment: Dream or nightmare? *Educational Measurement: Issues and Practice*, 43(4):119–125.

Eren Bilen and Alexander Matros. 2021. Online cheating amid COVID-19. *Journal of Economic Behavior & Organization*, 182:196–211.

Charles Bird. 1927. The detection of cheating in objective examinations. *School & Society*, 25:261–262.

Charles Bird. 1929. An improved method of detecting cheating in objective examinations. *The Journal of Educational Research*, 19(5):341–348.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Gregory J. Cizek and James A. Wollack, editors. 2016. *Handbook of Quantitative Methods for Detecting Cheating on Tests*. Routledge, New York City, NY, USA.

Neil J. Dorans and Linda L. Cook. 2016. *Fairness in Educational Assessment and Measurement*. Routledge.

Michael Fauss. 2024. tmvbeta: Truncated multivariate beta distribution on the unit hypercube.

Michael Fauss, Xiang Liu, Chen Li, Ikkyu Choi, and H. Vincent Poor. 2025. Bayesian selection policies for human-in-the-loop anomaly detectors with applications in test security. Under review for publication in Psychometrika.

Stefan Janke, Selma C. Rudert, Änne Petersen, Tanja M. Fritz, and Martin Daumiller. 2021. Cheating in the wake of COVID-19: How dangerous is ad-hoc online testing for academic integrity? *Computers and Education Open*, 2:100055.

Matthew S. Johnson, Xiang Liu, and Daniel F. McCaffrey. 2022. Psychometric methods to evaluate measurement and algorithmic bias in automated scoring. *Journal of Educational Measurement*, 59(3):338–361.

Neil Kingston and Amy Clark. 2014. *Test Fraud: Statistical Detection and Methodology*. Routledge Research in Education. Taylor & Francis, Milton Park, Abingdon-on-Thames, UK.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Tobi Walsh, Armin Hamrah, Lapo Santarlasci, and 4 others. 2025. Artificial intelligence index report 2025. Annual report, Stanford University Institute for Human-Centered AI.

Philip M. Newton and Keioni Essex. 2023. How common is cheating in online exams and did it increase during the COVID-19 pandemic? A systematic review. *Journal of Academic Ethics*, pages 1–21.

Mike Perkins, Jasper Roe, Binh H. Vu, Darius Postma, Don Hickerson, James McGaughran, and Huy Q. Khuat. 2024. Simple techniques to bypass genAI text detectors: implications for inclusive education. *International Journal of Educational Technology in Higher Education*, 21(1).

Flavien Prost, Hai Qian, Qiuwen Chen, Ed H. Chi, Jilin Chen, and Alex Beutel. 2019. Toward a better trade-off between performance and fairness with kernel-based distribution matching. *ArXiv*, abs/1910.11779.

Sandip Sinharay, Randy E. Bennett, Michael Kane, and Jesse R. Sparks. 2025. Validation for personalized assessments: A threats-to-validity approach. *Journal of Educational Measurement*.

Leonardo S. Sotaridona and Rob R. Meijer. 2002. Statistical properties of the k-index for detecting answer copying. *Journal of Educational Measurement*, 39(2):115–132.

Christina St-Onge, Kathleen Ouellet, Sawsen Lakhal, Tim Dubé, and Mélanie Marceau. 2022. COVID-19 as the tipping point for integrating e-assessment in higher education practices. *British Journal of Educational Technology*, 53(2):349–366.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *Preprint*, arXiv:2405.01724.

Teo Susnjak and Timothy R. McIntosh. 2024. Chatgpt: The end of online exam integrity? *Education Sciences*, 14(6).

Wim J. van der Linden and Leonardo Sotaridona. 2004. A statistical test for detecting answer copying on multiple-choice tests. *Journal of Educational Measurement*, 41(4):361–377.

Debora Weber-Wulff, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltýnek, Jean Guerrero-Dib, Olumide Popoola, Petr Šigut, and Lorna Waddington. 2023. Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*, 19(1).

James A. Wollack. 2003. Comparison of answer copying indices with real data. *Journal of Educational Measurement*, 21(4):307–320.

Duanli Yan, Michael Fauss, Jiangang Hao, and Wenju Cui. 2023. Detection of AI-generated essays in writing assessments. *Psychological Test and Assessment Modeling*, 65(1):125–144.

Meixun Zheng, Daniel Bender, and Cindy Lyon. 2021. Online learning during COVID-19 produced equivalent or better student course performance as compared with pre-pandemic: empirical evidence from a school-wide comparative study. *BMC medical education*, 21:1–11.

A Simulation Parameters

Let the parameters of the multivariate beta distribution in (Fauss, 2024) be denoted by $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{\Sigma}$.In our simulation, the feature distribution of the cheaters was assumed to be independent of the group and given by:

$C = 1$:

$$\begin{aligned}\boldsymbol{a}_1 &= \begin{bmatrix}4 & 4\end{bmatrix}\\ \boldsymbol{b}_1 &= \begin{bmatrix}2 & 2\end{bmatrix},\\ \boldsymbol{\Sigma}_1 &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\end{aligned}$$

The feature distribution of honest test takers was modeled groupwise with parameters:

$C = 0, G = 1$:

$$\begin{aligned}\boldsymbol{a}_{01} &= \begin{bmatrix}2 & 4\end{bmatrix}\\ \boldsymbol{b}_{01} &= \begin{bmatrix}6 & 4\end{bmatrix},\\ \boldsymbol{\Sigma}_{01} &= \begin{bmatrix}1 & -0.5\\ -0.5 & 1\end{bmatrix}\end{aligned}$$

$C = 0, G = 2$:

$$\begin{aligned}\boldsymbol{a}_{02} &= \begin{bmatrix}4 & 2\end{bmatrix}\\ \boldsymbol{b}_{02} &= \begin{bmatrix}4 & 6\end{bmatrix},\\ \boldsymbol{\Sigma}_{02} &= \begin{bmatrix}1 & 0.5\\ 0.5 & 1\end{bmatrix}\end{aligned}$$

$C = 0, G = 3$:

$$\begin{aligned}\boldsymbol{a}_{03} &= \begin{bmatrix}2 & 2\end{bmatrix}\\ \boldsymbol{b}_{03} &= \begin{bmatrix}4 & 4\end{bmatrix},\\ \boldsymbol{\Sigma}_{03} &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\end{aligned}$$

The Impact of an NLP-Based Writing Tool on Student Writing

Karthik Sairam and Amy Burkhardt and Susan Lottridge
Cambium Assessment
{karthik.sairam, amy.burkhardt, susan.lottridge}@cambiumassessment.com

Abstract

We present preliminary evidence on the impact of a NLP-based writing feedback tool, Write-On with Cambi! on students' argumentative writing. Students were randomly assigned to receive access to the tool or not, and their essay scores were compared across three rubric dimensions; estimated effect sizes (Cohen's d) ranged from 0.25 to 0.26 (with notable variation in the average treatment effect across classrooms). To characterize and compare the groups' writing processes, we implemented an algorithm that classified each revision as Appended (new text added to the end), Surface-level (minor within-text corrections to conventions), or Substantive (larger within-text changes or additions). We interpret within-text edits (Surface-level or Substantive) as potential markers of metacognitive engagement in revision, and note that these within-text edits are more common in students who had access to the tool. Together, these pilot analyses serve as a first step in testing the tool's theory of action.

1 Introduction

The writing feedback tool, Write-on with Cambi!, was designed to support students in revising their argumentative essays. It highlights key argumentative elements based on annotation guidelines aligned to standards, which have been shown to produce organizational patterns that correlate with rubric scores [1]. These annotations drive the tool's feedback in two primary ways. First, they provide students with a structured overview of their writing. Second, the absence of certain annotations in a student's essay triggers targeted feedback. [3] Beyond annotation-based feedback, the tool flags conventions-related errors (e.g., spelling, punctuation, grammar). It does not auto-correct; instead, it highlights each issue and provides guidance on how to revise it.

The tool is grounded in a theory of action that, at a high level, states: "Students who are guided through a structured review of their essays with immediate, annotated feedback that is well-aligned to teacher instruction will produce essays of higher overall quality."[4] To further theorize the causal mechanism that leads to this outcome, we posit that, by prompting students to examine potentially missing compositional elements and conventions-related errors, the tool elicits metacognitive processes (reviewing, evaluating, and editing) that, in turn, improve essay quality.

This study aims to begin to evaluate this theory of action by the way of the following two key research questions:

1. Do students with access to the tool achieve higher scores across all three dimensions of the scoring rubric? Additionally, how does the effect vary across different teachers in terms of both magnitude and direction?

2. How can we begin to analyze the differences in writing and revision strategies between students who have access to the tool and those who did not? How might we tie this back to the theory of action?

2 Methods

2.1 Randomized Pilot

At the end of the 2024 school year, 11 educators from two states volunteered to pilot the Write on with Cambi! (or Cambi!) tool in their grade 6 through 8th grade classrooms. This pilot study involved 262 seventh grade students within eleven classrooms, with 125 from State A and 137 from State B. During the test, students were randomly assigned access to the tool: 124 did not receive access (control group) and 138 students did receive access to the tool (treatment group).

To begin to assess the impact of the writing tool of Cambi! in student performance, students' essay responses were scored across three dimensions of

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 3: Coordinated Session Papers, pages 115–123

the rubric: Conventions, Elaboration and Organization by an automated scoring engine, Autoscore. All students answered the same writing prompt and were scored using the same rubric, which was common across the two states.

We report descriptive statistics (means/SDs), estimate effect sizes (Cohen's d), test group differences (two-sample t-tests; Wilcoxon rank-sum), describe score-point distributions, and examine heterogeneity in treatment effects by teacher/classroom.

2.2 Response Analysis

We collected the full text of each student's essay at 2-minute intervals throughout the writing session, which we will be referring to as "2-minute snapshots" or simply "snapshots."

This process yielded a primary corpus of 4,990 snapshots from 262 unique student participants. Each entry in this corpus contains the student's unique ID, their assigned group (treatment or control), a chronological snapshot sequence number, and the full text of their essay at that moment.

From a qualitative review of two-minute snapshots, we categorized essay revisions into two types: appending—adding new text to the end of the essay—and internal edits—changes made within the previously written text. We further distinguish two forms of internal edits:

1. Surface-level Edits: Minor corrections, oftentimes related to writing conventions, such as spelling, punctuation, and grammar.

2. Substantive Edits: Larger changes or additions within the previously written text of the essay.

Internal edits are of particular interest, as they may signify a deeper level of metacognition, suggesting a shift from automatic drafting to more deliberate and strategic composing.

To analyze revision patterns, we developed a custom algorithm to classify the changes between consecutive 2-minute snapshots. After tokenizing each snapshot's text using the NLTK library, we used a hierarchical classification logic to categorize every change into one of three mutually exclusive types:

1. **Appended Text**: Edits were first checked for location. Any change involving an addition of text at the very end of the previous snapshot was classified as an *Append*.

2. **Surface-level Edits**: If an edit was not an *Append*, its size was evaluated. Any internal change (an insertion, deletion, or replacement within the body of the text) involving 3 words or fewer was classified as a *Surface-Level* Edit.

3. **Substantive Edits**: Any internal edit involving more than 3 words was classified as a Substantive Edit.

This process yielded a count for each of the three edit types for every 2-minute interval. In the charts presented below, the average number of edits is calculated as the total number of edits of a specific type (e.g., surface-level) within a given writing stage, divided by the total number of students in that group.

The algorithms used for this classification of edits is detailed in Algorithm A.1

3 Results

3.1 Randomized Pilot

We analyzed the impact of thewriting tool on student writing by comparing a treatment group (acess to Cambi!) and a control group (without access to Cambi!) across three rubric dimensions: Conventions, Elaboration, and Organization.

3.1.1 Aggregate Results

Across all classrooms, essays written by students with access to Cambi! had higher mean scores on average, as outlined in Table 1. This corresponded to a Cohen's d ranging from 0.25 to 0.26. While this effect size may appear small, it should be noted that a review of over 700 k-12 intervention studies suggest an effect size of over .2 is considered large [2]

To test for statistical significance, we first ran two-sample t-tests, which assume scores are interval data. These tests, as shown in Table 2 confirmed the differences were statistically significant ($p<0.05$). To better account for the ordinal nature of the rubric scores (i.e., the distance between 1 and 2 may not equal the distance between 2 and 3), we also conducted a non-parametric Wilcoxon rank-sum test. The results of this test approached statistical significance at the $p<0.05$ level.

Analysis of the score point distributions revealed specific shifts for the those with access to the writing tool compared to the group without access to the tool, shown in Figure 1

Table 1: Comparison of Mean Scores and Effect Sizes by access to Cambi!

Access to Cambi!	Mean Score (SD)		
	Conventions	**Elaboration**	**Organization**
0 (n = 133)	1.50 (.72)	1.21 (.64)	1.46 (.72)
1 (n = 115)	1.67 (.60)	1.37 (.55)	1.63 (.58)
Effect Size (Cohen's d)	.26	.26	.25

Table 2: Two-sample *t*-test Results

Dimension	*t*-Statistic	*p*-value
Conventions	-2.056	0.0408
Elaboration	-2.044	0.0420
Organization	-2.016	0.0449

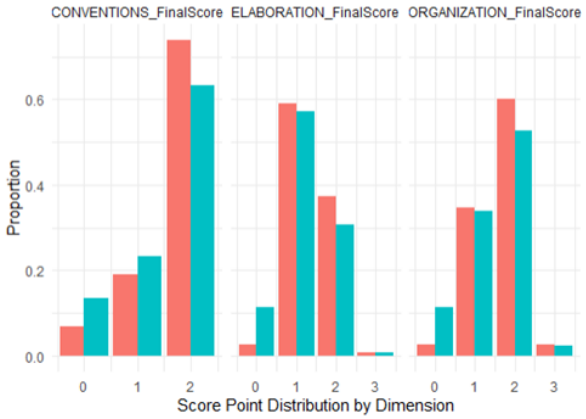

Figure 1: Score Point Distribution by treatment and control group

1. **Conventions**: The Cambi! group received more scores of 2 and fewer scores of 1 and 0.

2. **Elaboration**: The Cambi! group had fewer scores of 0, more scores of 1 and 2, and an equal proportion of 3s.

3. **Organization**: The Cambi! group earned fewer scores of 0, more scores of 1 and 2, and a slightly higher proportion of 3s.

Notably, for Elaboration and Organization, a score of 0 indicated a non-attempt, suggesting the tool helped students overcome initial writing inertia. Additionally, on Elaboration and Organization, no students in either group achieved the maximum score of 4.

3.1.2 Classroom-level Variability

Although aggregate results were positive, the estimated treatment effect varied across classrooms. We take a closer look at one dimension, Organization, to illustrate this variance in Figure 2. For this dimension, 6 of 11 classrooms showed a positive effect for Cambi!, 4 showed a negative effect, and 1 showed no difference.

The variation in results can be illustrated by examining the three largest classrooms, where teacher survey data helps interpret the quantitative findings:

1. **0F8C** *(d=0.32; N: 15 control, 22 treatment)*: This classroom showed a positive effect, but the teacher provided no comment on their implementation strategy.

2. **CCA5** *(d=-0.03; N: 24 control, 14 treatment)*: This classroom showed a negligible effect. The teacher noted that student engagement may have been skewed by low motivation, as the voluntary pilot took place after summative testing at the end of the year.

3. **5F4E** *(d=0.22; N: 29 control, 41 treatment)*: This classroom showed a positive effect. The teacher reported actively scaffolding the tool by going through each feedback tab with students to ensure they understood the suggestions and how to apply them.

In the next section, these pilot results are futher explored to understand how we can begin to analyze the differences in writing and revision strategies between students who have access to the tool and those who did not.

3.2 Response Analysis

In this section, we describe how the revision process—categorized into three edit types—differs between students with and without access to the writing tool.

3.2.1 Overview and Appended Text

An analysis of the overall composition of edits shows that appending new text was a common behavior in both groups. However, as seen in Figure 3, those without access to Cambi! dedicated a

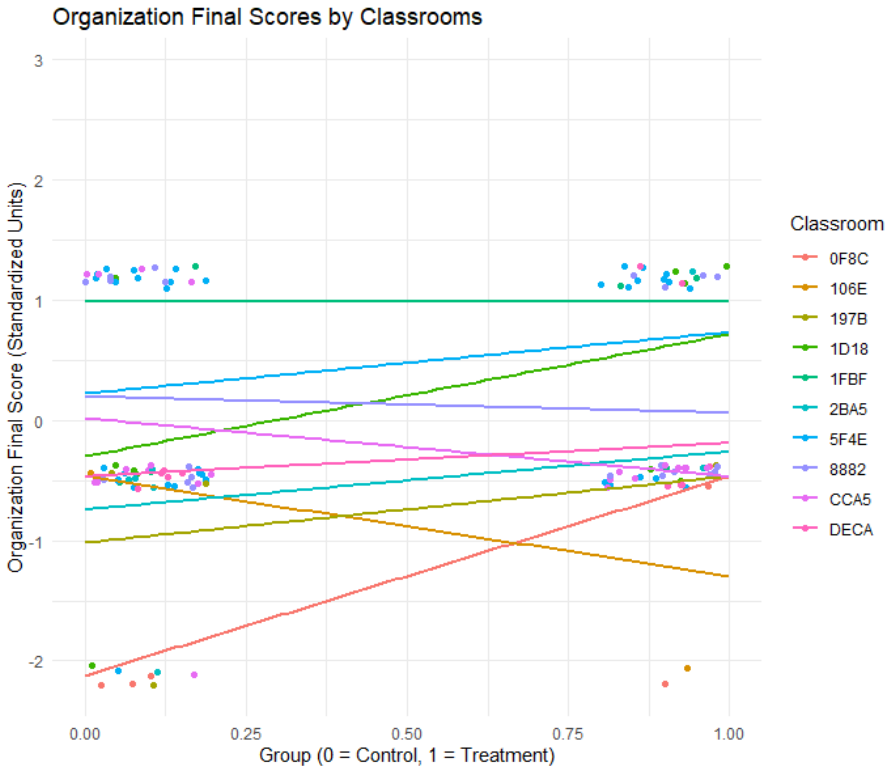

Figure 2: Organization Effect of Access to Cambi! by Teacher/Classroom

larger proportion of their total revision activity to appended edits compared to those who had access to the writing tool.

3.2.2 Surface-level Edits

The timing and frequency of small, surface-level edits revealed a notable difference in writing workflow between the groups.

1. **Overall Trend**: As shown in Figure 4, both groups steadily accumulated surface-level edits throughout the writing session. Notably, while those without access to the writing tool maintained a slightly higher cumulative edit count for the first three quarters, those with access showed a marked acceleration in editing during the final stage (76-100%). This timing aligns with the tool's feedback flow: conventions-related feedback is delivered only after students receive more substantive feedback focused on compositional elements.

2. **Analysis by Score Point**: This trend was most pronounced among students with high scoring essays on the Conventions score. However, the most striking difference was observed among students who ultimately scored a zero on Conventions (Figure 5). Students who had access to the tool but received a lower score showed a high and increasing level of cumulative surface-level edits. In contrast, the control group's essays that received zero scores show almost no cumulative editing activity.

3.2.3 Substantive Edits

The analysis of substantive edits (defined as internal edits involving more than three words) reveals a divergence in revision strategy between the two groups. As shown in Figure 6, students with access to Cambi! consistently accumulated more substantive edits than the control group throughout the entire writing session. The gap between the two groups widened over time, with the treatment group performing a substantially higher number of total substantive revisions by the end of the session.

This difference in behavior was most pronounced among the essays with the highest scores. Figure 7 illustrates the cumulative edits specifically for students who earned a score of 3 on the Organization rubric. For this tier, the treatment group's engagement in substantive revision was higher, with an average of nearly 11 cumulative edits by the end of the writing time. In contrast, their control group peers who also scored a 3 performed very few of these edits, averaging just over 2 by the session's

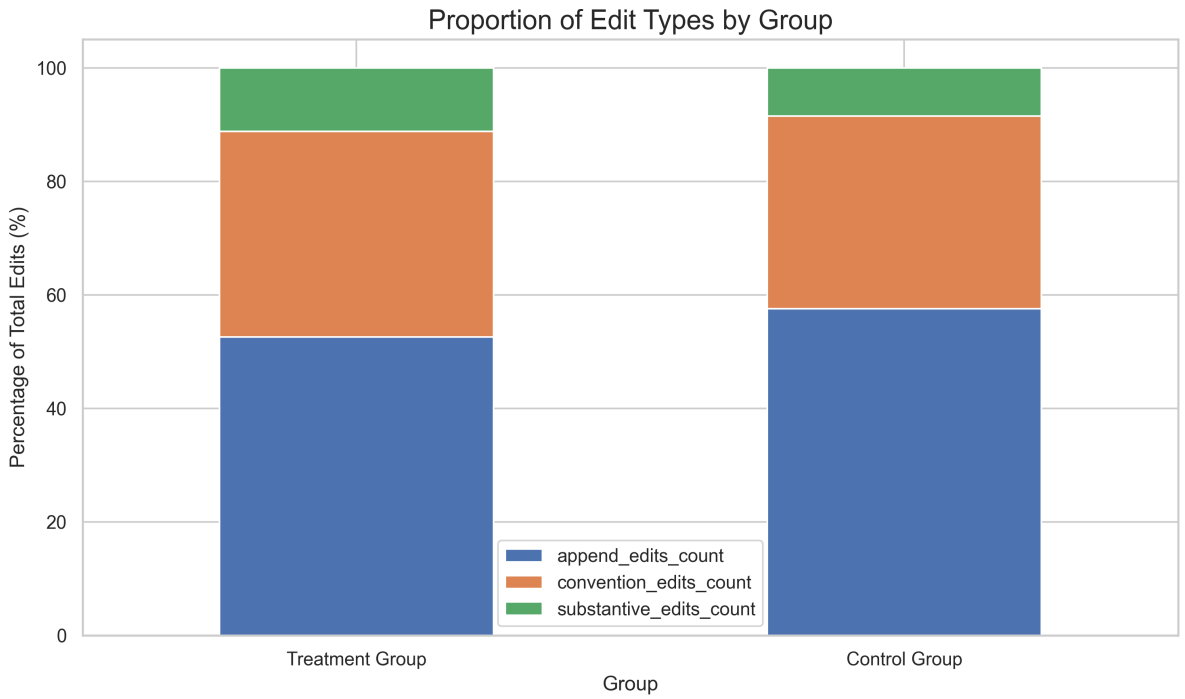

Figure 3: Proportion of Total Edit Types by group.

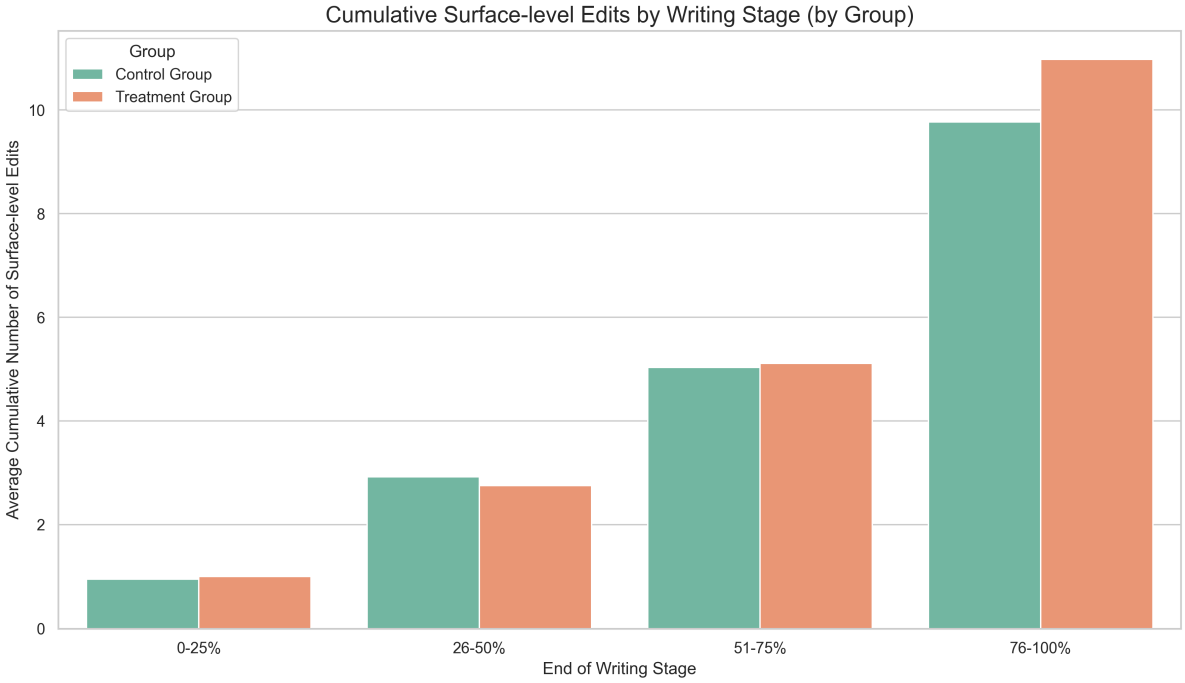

Figure 4: Cumulative Surface-level edits by Writing Stage

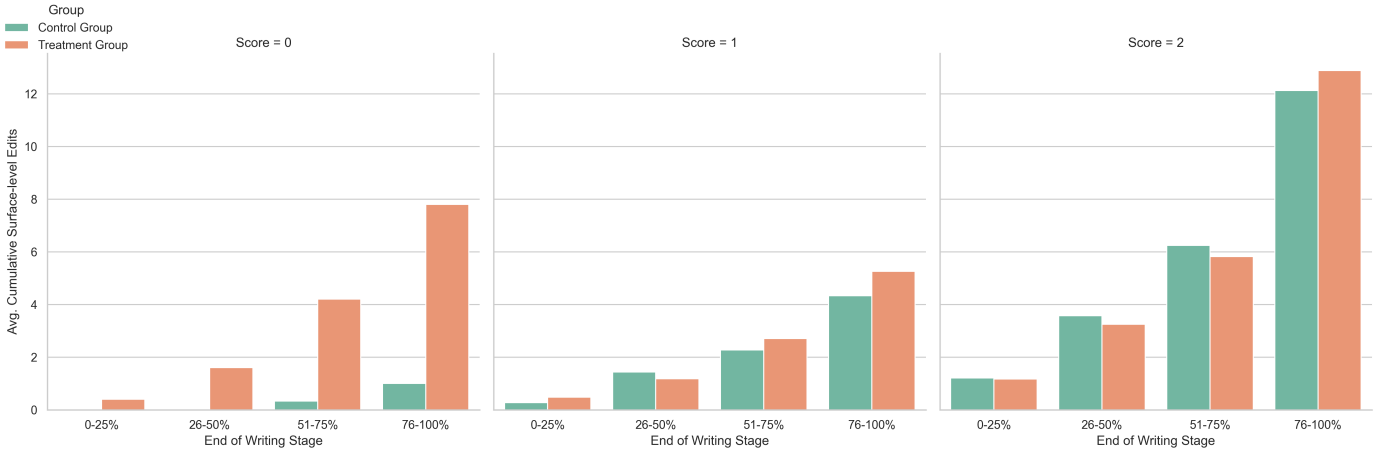

Figure 5: Cumulative Surface-level Edits by Conventions Score Tier

end. The graphs for Organization scores 1 and 2 are in A.2

4 Discussion and Limitations

This work provides preliminary evidence for the effectiveness of an AI-powered writing tool through a pilot study in which students were randomly assigned access. It also introduces methods for exploring the underlying mechanisms that explain how and why the tool influences writing behavior and outcomes, and ties back to the theory of action.

We observed a notable effect of using the tool on student rubric scores, as scored by Autoscore, in the aggregate. Across classes and states, the effect size for each rubric dimension ranged between .25 and .26. While this effect size is large in educational contexts, the outcome of the rubric scores is strongly aligned and scored immediately after the student wrote the essay. As such, with less aligned and further apart outcome variables, we may expect an effect size of a smaller magnitude. Furthermore, observed heterogeneity in the average treatment effect across classrooms, as expected given differences in implementation and baseline writing ability.

We offer several considerations when interpreting the results:

1. **Intent-to-Treat Study**: In this study, we only know if a student was granted access to the tool, but we did not track if the tool was used. The results should be interpreted accordingly. These results provide an estimate of the tool's effectiveness in a real-world setting, where not every student may utilize the tool, they have access to.

2. **Test Fatigue**: The pilot occurred after annual summative assessments, and teachers noted student fatigue. This low-stakes context may have suppressed scores across both groups and masked a larger potential effect.

3. **Control group Behavior**: The control group, aware they lacked access to a new AI tool, may have been less motivated, potentially inflating the observed difference between the groups.

4. **Treatment Diffusion**: Teachers reported helping students interpret Cambi! feedback. It's possible that this guidance was overheard by or shared with control group students, which would weaken the measured effect

In this paper, we also explored methods to begin to analyze differences in writing and revision strategies between students who have access to the tool and those who do not. First, we qualitatively reviewed and categorized the two-minute snapshots into three forms of revisions: appending, surface-level, and substantive. The latter two forms of in-text revisions are aligned with our theory of action that the tool may lead the student to engage in the metacognitive task of reviewing, evaluating, and editing their text.

The findings indicate that students with access to Cambi! tended to shift their efforts from simply appending text toward more internal revisions. This pattern varied across levels of student performance.

For surface-level edits, the tool's impact was most evident among students who received lower scores. As shown in Figure 5, students in the treatment group who ultimately scored a 0 on Conven-

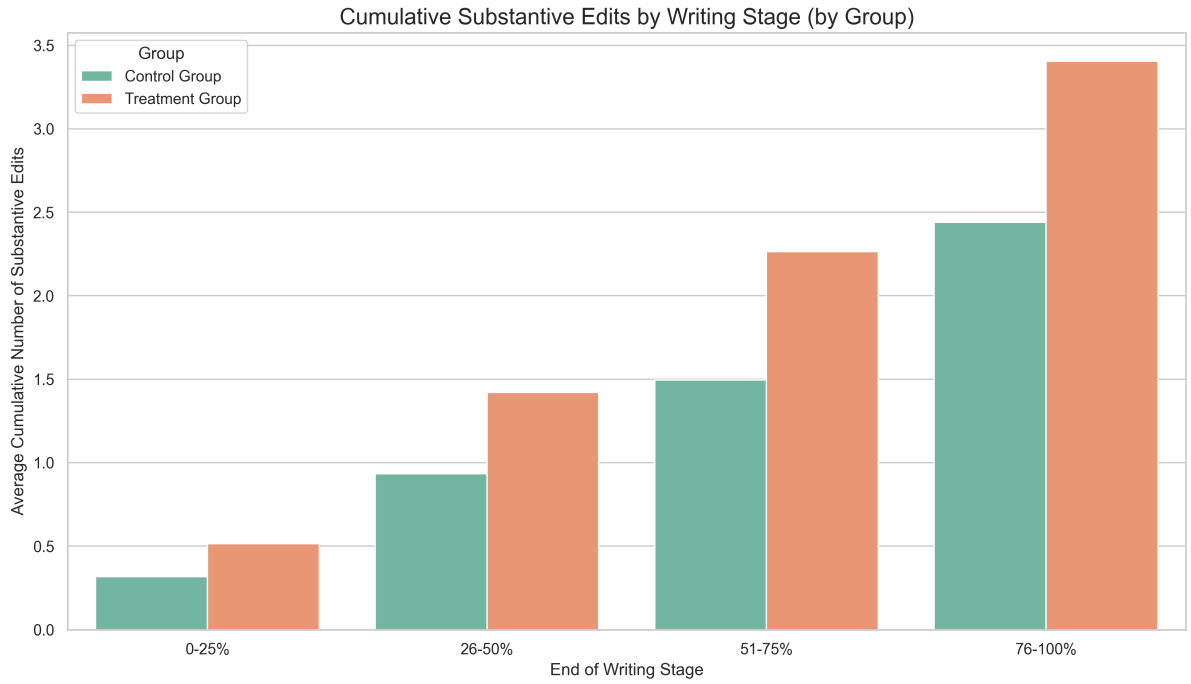

Figure 6: Cumulative Substantive Edits by Writing Stage

tions attempted a notable number of edits, whereas their counterparts in the control group made very few. While these edits did not raise their final scores in this instance, this finding indicates that the tool can prompt engagement from students who might otherwise remain passive.

For substantive edits, the effect was particularly notable among students with higher scores. The data from students who achieved a score of 3 on the Organization rubric shows a substantially higher number of substantive revisions for the treatment group compared to their control group peers (Figure 7). This suggests the tool may act as a scaffold, guiding students who already are capable writers to move beyond surface-level fixes and engage in more complex, structural revision. Future work could also explore different word-count thresholds for differentiating between surface-level and substantive edits.

4.1 Conclusion

Building the evidence base for a writing tool in argumentative writing is ongoing. This paper offers preliminary findings that Write-On with Cambi! can support students and proposes a path for analyzing the mechanisms behind observed score differences. These results serve as a first step in testing the tool's theory of action.

References

[1] Amy Burkhardt, Suhwa Han, Sherri Woolf, Allison Boykin, Frank Rijmen, and Susan Lottridge. 2025. Standards-aligned annotations reveal organizational patterns in argumentative essays at scale. *Frontiers in Education*, 10.

[2] Matthew A. Kraft. 2018. Interpreting effect sizes of education interventions. Technical report, Brown University.

[3] Sue Lottridge, Amy Burkhardt, Christopher Ormerod, Sherri Woolf, Mackenzie Young, Milan Patel, Harry Wang, Julius Frost, Kevin McBeth, Julie Benson, Michael Flynn, Kevin Dwyer, Scott Fitz, Radd Berkheiser, Henry Floyd, Dave Davis, Ben Godek, and Quinell Wilson. 2025. Write on with cambi: The development of an argumentative writing feedback tool. Technical report, Cambium Assessment, Inc.

[4] Sue Lottridge, Chris Ormerod, and Amy Burkhardt. 2025. Development and validation of an AWE system "Write On with Cambi!". In *Proceedings of the National Council on Measurement in Education (NCME)*, Denver, CO.

A Appendix

A.1 Algorithm for edits retrieval

The algorithm used for classifying the edits into three different categories, appended, surface-level and substantive, is outlined in Algorithm 1

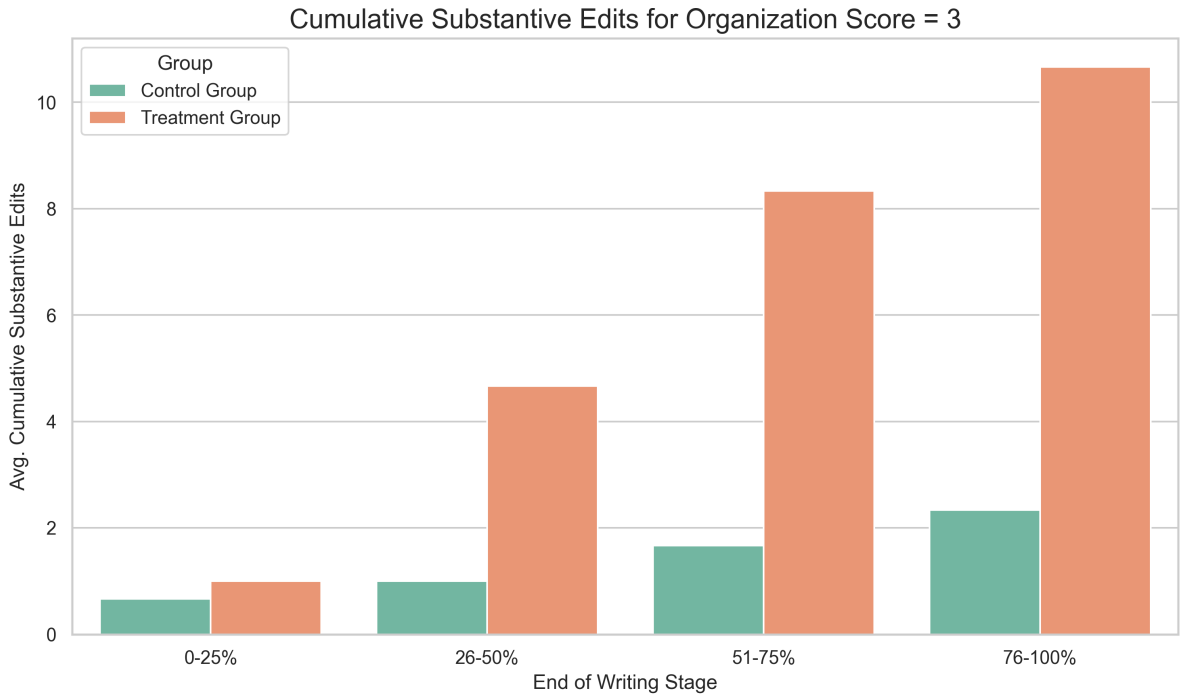

Figure 7: Cumulative Substantive Edits for Students with a Top Organization Score (3)

Algorithm 1 Classification of Revision Edits

procedure CLASSIFYREVISION($S_{before}, S_{after}, N_{threshold}$)
  Input: S_{before} (previous text), S_{after} (current text), $N_{threshold}$ (word count limit)
  Output: List of classification labels for each edit
  $W_{before} \leftarrow$ TokenizeAndClean(S_{before})
  $W_{after} \leftarrow$ TokenizeAndClean(S_{after})
  $W_{after} \leftarrow$ HandleChoppedWord(W_{before}, W_{after})
  $Opcodes \leftarrow$ Diff(W_{before}, W_{after})
  $Edits \leftarrow []$
  for all $(tag, i_1, i_2, j_1, j_2) \in Opcodes$ **do**
    if $tag =$ 'equal' **then continue**
    end if
    if $tag =$ 'insert' $\wedge\, i_1 = |W_{before}|$ **then**
      Append "Appended" to $Edits$
    else if $\max(i_2 - i_1, j_2 - j_1) \leq N_{threshold}$ **then**
      Append "Surface-level" to $Edits$
    else
      Append "Substantive" to $Edits$
    end if
  end for
  return $Edits$
end procedure

A.2 Graphs for Organization scores 1 and 2

Figures 8 and 9 illustrate the cumulative edits by students who earned a score of 1 and 2, respectively, on the Organization rubric

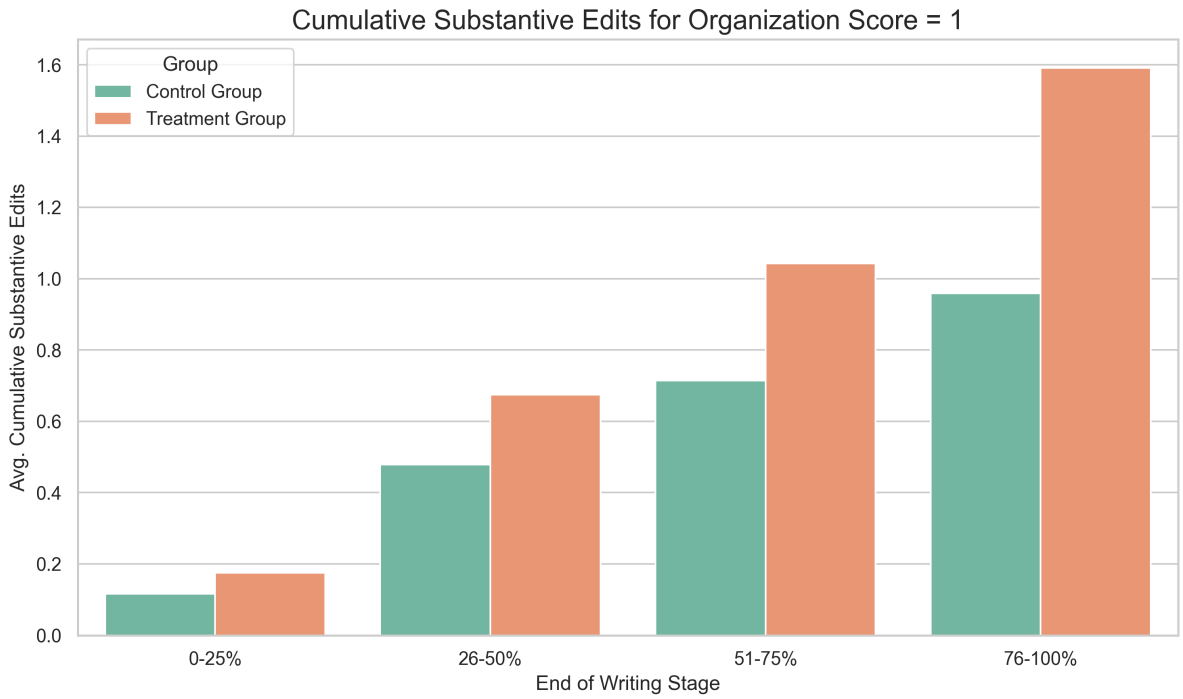

Figure 8: Cumulative Substantive Edits for Students with a Organization Score=1

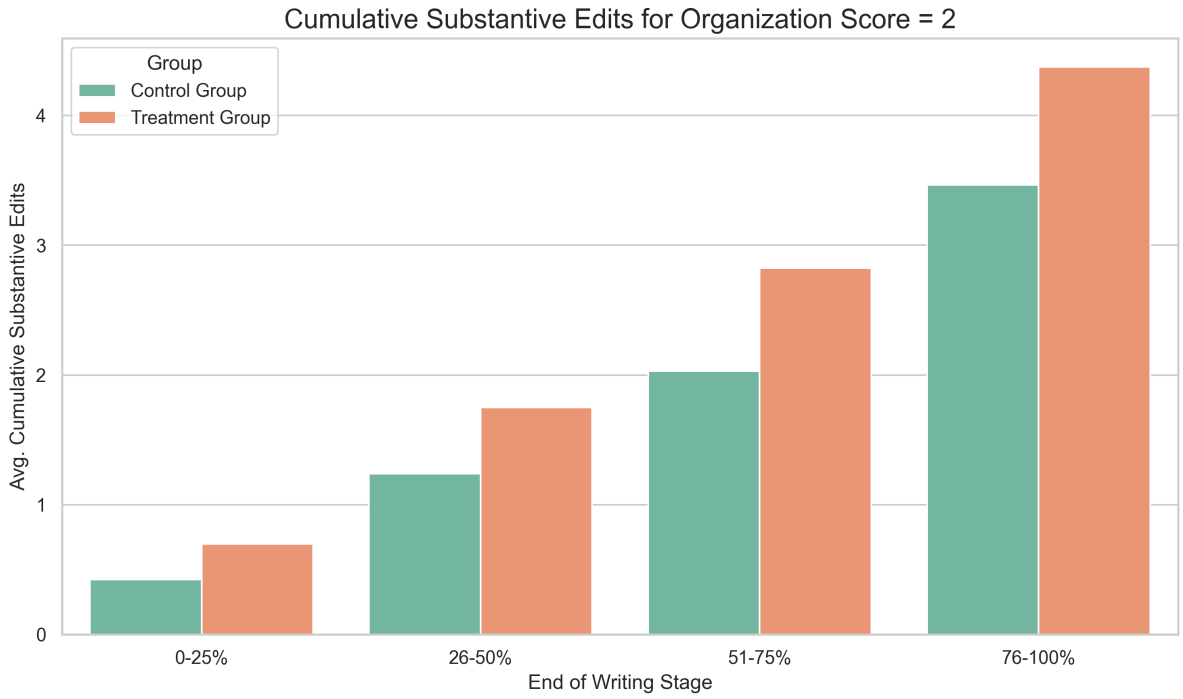

Figure 9: Cumulative Substantive Edits for Students with a Organization Score=2

Author Index