Averroes at BEA 2025 Shared Task: Verifying Mistake Identification in Tutor, Student Dialogue

Mazen Yasser[1], Mariam Saeed[1], Hossam Elkordi[1], Ayman Khalafallah[1]
[1]Applied Innovation Center,
Correspondence: m.yasser, m.saeed, h.elkordi, a.khalafallah@aic.gov.eg

Abstract

This paper presents the approach and findings of Averroes Team in the BEA 2025 Shared Task Track 1: Mistake Identification. Our system uses the multilingual understanding capabilities of general text embedding models. Our approach involves full-model fine-tuning, where both the pre-trained language model and the classification head are optimized to detect tutor recognition of student mistakes in educational dialogues. This end-to-end training enables the model to better capture subtle pedagogical cues, leading to improved contextual understanding. Evaluated on the official test set, our system achieved an exact macro-F_1 score of 0.7155 and an accuracy of 0.8675, securing third place among the participating teams. These results underline the effectiveness of task-specific optimization in enhancing model sensitivity to error recognition within interactive learning contexts.

1 Introduction

Tutoring has long been recognized as one of the most effective educational interventions, significantly enhancing student learning outcomes. Notably, the 2 sigma problem Bloom (1984) illustrates that students receiving one-on-one tutoring perform two standard deviations better than those in conventional classroom settings, highlighting the profound impact of personalized instruction. However, the scalability of such individualized tutoring remains a challenge due to resource constraints.

Advancements in deep learning Lin et al. (2023) and the emergence of large language models (LLMs) Lieb and Goel (2024); Park et al. (2024) have paved the way for AI-powered tutors capable of delivering personalized, on-demand educational support. These intelligent tutoring systems leverage natural language processing and machine learning techniques to adapt to individual learner needs, providing real-time feedback and tailored instruction. AI-powered tutors can make quality education available to more people by offering the same benefits as one-on-one tutoring, but for many students at once.

Despite these advancements, evaluating the pedagogical effectiveness of AI tutors remains a significant problem. Traditional evaluation metrics, often adapted from domains like machine translation and summarization, fail to capture the nuanced educational interactions between AI tutors and students. Moreover, while human evaluations are considered the gold standard, they are time-consuming, costly, and lack scalability. This highlight the urgent need for automated, reliable, and pedagogically-informed evaluation frameworks.

Addressing this gap, the BEA 2025 Shared Task Kochmar et al. (2025) focuses on the Pedagogical Ability Assessment of AI-powered Tutors, aiming to develop standardized evaluation methods for AI tutor responses. The task includes four main tracks: Mistake Identification, determining whether the AI tutor correctly identifies student errors; Mistake Localization, pinpointing the exact location or nature of the student's mistake; Guidance Provision, offering constructive feedback or hints to guide the student; and Actionability, ensuring the response leads to a clear next step for the student. These tracks are intended to measure the tutor's effectiveness in supporting student learning and correcting misunderstandings.

This paper describes our contribution to the BEA 2025 Shared Task, in which we leverage large language models (LLMs) to create an automated evaluation method for AI tutors, primarily focusing on the mistake identification track. We investigate multiple strategies, assess their performance, and present a comprehensive ablation study, delivering a scalable, education-focused evaluation framework designed to enhance personalized learning.

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 1121–1126

2 Related Work

2.1 AI Tutoring Systems

Early Intelligent Tutoring Systems (ITS), developed in the late 1970s and 1980s Guo et al. (2021), employed explicit cognitive or knowledge-tracing models to monitor learners' progress and simulate personalized instruction . Pioneering systems like Anderson and Corbett's Cognitive Tutors Anderson et al. (1995) utilized model-tracing algorithms to instantly detect deviations from expert problem-solving pathways, allowing immediate corrective feedback and error-specific hints. This approach significantly boosted students' learning speed and post-test performance in experimental settings. However, studies of human expert tutors, such as Hume et al. (1996), suggest a more effective approach, using indirect prompts such as Socratic questions or reflective hints to help students independently identify and correct errors. This approach encourages deeper learning and self-reflection, showing a limitation of early ITS.

2.2 Advances in Large Language Models for Educational Dialogue

Recent advancements in large language models (LLMs) have significantly improved their capabilities, especially within educational contexts Lieb and Goel (2024); Kasneci et al. (2023); Nye et al. (2023). Modern LLMs facilitate personalized, interactive tutoring experiences, creating customized content such as quizzes and lesson plans tailored to specific curricula and student proficiency. Furthermore, these models support educators by automating administrative responsibilities, enabling teachers to devote more time to direct instruction and student engagement.

2.3 Evaluation Methods for AI Tutoring Systems

Evaluating AI tutors in education primarily relies on human judgment that score responses on dimensions like mistake identification, clarity, and tone. While expert annotation remains the gold standard, it suffers from inconsistency and lacks a unified protocol, prompting studies such as Tack & Piech Tack and Piech (2022), and Maurya et al.Maurya et al. (2025) propose standardized taxonomies. Pairwise comparisons simplify evaluation by focusing on relative pedagogical effectiveness. However, automatic metrics remain limited: Traditional nautural language generation metrics such as BLEU Papineni et al. (2002) or ROUGE Lin (2004) poorly reflect pedagogical quality. Recent advances use reference-free approaches such as trained scorers (e.g., DialogRPT Gao et al. (2020)) and LLMs like GPT-4 [1] to evaluate tutor responses, though their reliability depends heavily on prompt design. Hybrid evaluation methods that combine LLMs and correctness checks are emerging to improve consistency and scalability.

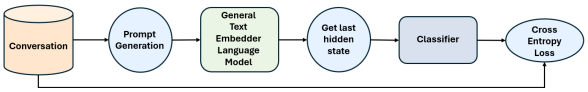

Figure 1: Model Architecture

3 System Overview

This section presents the complete methodology adopted for the task. We first formalize the problem, then detail the shared backbone architecture, followed by dedicated subsections describing each experimental variant. Finally, we present our quantitative analysis and comparison between different approaches in 4.3.

3.1 Problem Definition

We address the task of assessing whether an AI tutor's feedback in a dialogue setting correctly identifies a student's mistake. Given a multi-turn conversation between a student and an AI tutor, along with the tutor's final response, the objective is to classify that response as correctly identifying the mistake, to some extent identifying, or failing to do so. This is formulated as a sequence classification problem, where a contextual understanding of the conversation is required for an accurate prediction.

3.2 System Backbone

We employ, as shown in Figure 1, a sequence classification approach. To effectively capture the contextual dependencies in the dialogue, we prepend a task-specific system prompt to the conversation history and the tutor's final turn. The system prompt is defined as:

[1] https://openai.com/index/gpt-4

System Prompt

You are tasked with evaluating a multi-turn conversation between a math teacher and a student. The conversation is about a mathematical problem and in the form of a dialogue aimed at helping the student arrive at the correct solution.
The student initially provides an incorrect answer. The teacher then engages in follow-up exchanges to help the student uncover and understand the mistake.
You will be given:
- The full conversation up to the student's most recent turn, enclosed within '<CONV>' tags.
- The math teacher's immediate next response, enclosed within '<RESP>' tags.
Your task:
- Determine whether the teacher's response in '<RESP>' effectively contributes to identifying or addressing the student's mistake.
- Explain your reasoning clearly and concisely based on the content of the teacher's response and how it relates to the mistake and the original question. Then, provide your final judgment.
A teacher's response is considered a **mistake identifier** if it includes:
- A follow-up question, explanation, or prompt that targets the student's misunderstanding or errors in reasoning,
- Or if it guides the student toward re-evaluating key steps relevant to solving the original math problem.
You must output one of the following judgments based on the above criteria:
- **A** → If the teacher's response is clearly focused on the student's mistake and relates directly to the solution steps.
- **B** → If the response is unrelated to the mistake, irrelevant to the solution steps, or potentially confusing/misleading.
- **C** → If the response is only partially relevant or offers indirect guidance that might help the student reflect on the mistake.
Put Your Output In The Following Format: <think>The complete reasoning process</think><answer>Your final judgment from the choices (A, B, or C)</answer>

This input is passed through a decoder, where the last hidden-state representation is extracted. A lightweight classification head, implemented as a feed-forward linear layer, is then applied to predict how the tutor response identifies the mistake among three classes (Yes, No, To some extent). This design leverages the model's pretrained contextual embeddings, enhancing its capacity to discern nuanced dialogue interactions.

3.3 GTE-based Sequence Classification Models

We investigate three variants that use the *General Text Embedding* (GTE) family to obtain sentence-level representations, followed by lightweight feed-forward (FF) classification heads:

1. **GTE-ModernBERT-Base** [2] Zhang et al. (2024): the `gte-modernbert-base` encoder feeds into a single FF layer with a softmax output for prediction.

2. **GTE-Qwen2-1.5B-1FF** Li et al. (2023): embeddings from `gte-qwen2-1.5B-instruct` [3] are passed through one FF layer identical to (1).

3. **GTE-Qwen2-1.5B-2FF**: the same as in (2) but followed by a two-layer FF head before the final softmax output.

Unless otherwise stated, these models are fine-tuned with the optimization settings described in §4.2.

3.4 Qwen2.5-based Sequence Classification Models

We benchmark five instruction-tuned *Qwen2.5* language models, varying both model size and the depth of the feed-forward (FF) classification head that replaces the original causal-LM head:

1. **Qwen2.5-7B-1FF** Team (2024): 7B parameters variant of `Qwen2.5` [4]; a single FF layer with software output. Fine-tuned via LoRA adapters Hu et al. (2022) with rank 16 on all attention and MLP projection layers.

[2] https://huggingface.co/Alibaba-NLP/gte-modernbert-base
[3] https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct
[4] https://huggingface.co/Qwen/Qwen2.5-7B

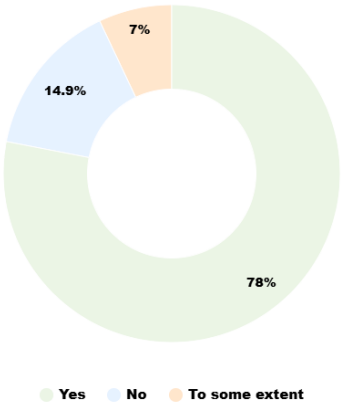

Figure 2: Class Distribution of the dataset.

2. **QWEN2.5-1.5B-2FF**: 1.5B parameters [5]; a two-layer FF head preceding the final softmax layer; full-parameter fine-tuning.

3. **QWEN2.5-MATH-1.5B-1FF**: math specialized 1.5B variant [6]; one FF layer; full-parameter fine-tuning.

4. **QWEN2.5-0.5B-1FF**: 0.5B parameters; one FF layer; full-parameter fine-tuning.

5. **QWEN2.5-0.5B-2FF**: same 0.5B backbone as (4) but with a two-layer FF head as in (2).

Unless otherwise stated, optimization hyperparameters follow the settings in §4.2.

4 Experiments

4.1 Dataset and Metrics

We conduct our experiments on **MRBench**, an annotated collection of 192 multi-turn student-AI tutor dialogues (1596 tutor responses) released by Maurya et al. (2025). Each tutor's response is labeled to indicate whether the feedback correctly identifies the student's error. Figure 2 shows the class distribution in the provided dataset. For model development, we divide the official development data into training and validation splits, retaining 15% of the dataset for validation during fine-tuning while maintaining the same class distribution of the train split. We follow the shared-task protocol and report strict *macro-averaged* F_1 and strict *accuracy* over the MISTAKE-IDENTIFICATION labels of the official test set.

[5]https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct

[6]https://huggingface.co/Qwen/Qwen2.5-Math-1.5B-Instruct

4.2 Training Setup

Each model was fine-tuned for no more than ten epochs using AdamW with a linearly decaying learning-rate schedule, reaching a maximum of 1×10^{-5}. We trained with an effective batch size of 64 in `bf16` mixed precision on a single NVIDIA RTX-A6000 GPU.

4.3 Results and Analysis

Model	Accuracy (%)	Macro-F_1 (%)
GTE-MODERNBERT-BASE	88.17	66.48
GTE-QWEN2-1.5B-1FF	**89.78**	**74.15**
GTE-QWEN2-1.5B-2FF	89.25	72.51
QWEN2.5-7B-1FF	85.48	64.06
QWEN2.5-1.5B-2FF	88.44	71.69
QWEN2.5-MATH-1.5B-1FF	88.44	67.95
QWEN2.5-0.5B-1FF	89.25	72.96
QWEN2.5-0.5B-2FF	88.44	71.15

Table 1: Accuracy and Macro-F_1 on our validation split.

4.3.1 Full fine-tuning wins

Training the entire decoder-only model `GTE-QWEN2-1.5B` with a single feed-forward head (**1FF**) achieves the best results on our validation split at 74.15 macro-F_1.

4.3.2 Small-but-efficient models keep pace

The smaller fully fine-tuned `QWEN2.5-0.5B-1FF` achieved our second best results at 72.96 macro-F_1 with only 1.2 points difference from our best model while cutting memory and latency.

4.3.3 More head depth is not always better

Adding a second feed-forward layer (**2FF**) to the backbone reduces performance.

4.3.4 Domain pre-training helps but not enough

The math-specialized `QWEN2.5-MATH-1.5B-1FF` outperforms the larger variant `QWEN2.5-7B- 1FF` by 3.89 F_1 with only 20% of its parameter size. However, increasing parameter count of non-specialized models surpasses the benefit of domain-specific training. In our case, `QWEN2.5-0.5B-1FF` outperforms the trained model by 5.01, `QWEN2.5-1.5B-2FF` by 4.74, and `GTE-QWEN2-1.5B-1FF` by 6.2.

4.3.5 Size alone isn't enough

The PEFT-tuned 7B `QWEN2.5-7B-1FF` achieves 6^{th} place at 64.06 macro-F_1, showing that the tuning was not effective.

5 Conclusion

This work benchmarked eight GTE- and Qwen2.5-based sequence-classification models on the MISTAKE-IDENTIFICATION task in AI-tutor dialogues. Full fine-tuning of a medium-sized decoder-only backbone (GTE-QWEN2-1.5B-1FF) achieved the strongest development performance at 74.1 macro-F_1, highlighting that carefully tuned 1.5 B models can outperform much larger 7B LoRA baselines.

These findings indicate that compact instruction-tuned LLMs can rival, or even surpass, their larger counterparts in pedagogical mistake detection, offering a resource-efficient pathway toward scalable AI tutors. Future work should expand the dialogue corpus, diversify subject matter and languages, incorporate richer pedagogical labels, and pair automatic metrics with human and learning outcome evaluations to approach genuinely effective educational dialogue systems.

Limitations

Our study is constrained by several factors that temper the generality of its findings. First, the evaluation corpus, MRBench, comprises only 1596 labelled tutor responses drawn from a single English, mathematics-focused dataset. Such limited scale and topical focus may bias the models toward the annotation style and error distribution specific to this domain, leaving their behavior untested in other subjects, proficiency levels, or languages.

Second, the present metrics provide only a partial view of the educational effectiveness. Moreover, we rely exclusively on automatic accuracy and macro-F_1; the absence of human judgments or learning-gain measurements means that the impact in real-world scenarios remains uncertain.

References

John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207.

Benjamin S Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking-training with large-scale human feedback data. In *EMNLP*.

Lu Guo, Dong Wang, Fei Gu, Yazheng Li, Yezhu Wang, and Rongting Zhou. 2021. Evolution and trends in intelligent tutoring systems research: a multidisciplinary and scientometric view. *Asia Pacific Education Review*, 22(3):441–461.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Gregory Hume, Joel Michael, Allen Rovick, and Martha Evens. 1996. Hinting as a tactic in one-on-one tutoring. *The Journal of the Learning Sciences*, 5(1):23–47.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, and 4 others. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Ekaterina Kochmar, Kaushal Kumar Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. Findings of the bea 2025 shared task on pedagogical ability assessment of ai-powered tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Anna Lieb and Toshali Goel. 2024. Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8.

Chien-Chang Lin, Anna YQ Huang, and Owen HT Lu. 2023. Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. *Smart Learning Environments*, 10(1):41.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.

Benjamin D Nye, Dillon Mee, and Mark G Core. 2023. Generative large language models for dialog-based tutoring: An early consideration of opportunities and concerns. In *LLM@ AIED*, pages 78–88.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Minju Park, Sojung Kim, Seunghyun Lee, Soonwoo Kwon, and Kyuseok Kim. 2024. Empowering personalized learning through a conversation-based tutoring system with student modeling. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–10.

Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *Preprint*, arXiv:2205.07540.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.