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

Fatima Dekmak
American University of Beirut
Beirut, Lebanon
fkd04@mail.aub.edu

Christian Khairallah
Aralects
Abu Dhabi, United Arab Emirates
christiank@aralects.com

Wissam Antoun
INRIA
France
wissam.antoun@gmail.com

Abstract

In light of the growing adoption of large language models (LLMs) as educational tutors, it is crucial to effectively evaluate their pedagogical capabilities across multiple dimensions. Toward this goal, we address the Mistake Identification sub-task of the BEA 2025 Shared task, aiming to assess the accuracy of tutors in detecting and identifying student errors. We experiment with several LLMs, including GPT-4o-mini, Mistral-7B, and Llama-3.1-8B, evaluating them in both zero-shot and fine-tuned settings. To address class imbalance, we augment the training data with synthetic examples, targeting underrepresented labels, generated by Command R+. Our GPT-4o model fine-tuned on the full development set achieves a strict macro-averaged F1 score of 71.63%, ranking second in the shared task. Our work highlights the effectiveness of fine-tuning on task-specific data and suggests that targeted data augmentation can further support LLM performance on nuanced pedagogical evaluation tasks.

1 Introduction

The increasing integration of large language models into educational applications has sparked significant interest in their potential as AI tutors capable of engaging students in meaningful learning dialogues. A critical component of effective tutoring lies in the ability to identify and address student misconceptions or errors. While recent studies have explored the capabilities of LLMs in simulating tutor-like behaviors, there remains a pressing need for systematic frameworks to evaluate their pedagogical effectiveness.

The BEA 2025 Shared Task (Kochmar et al., 2025) introduced a structured evaluation of AI tutors' responses, focusing on four pedagogical dimensions: mistake identification, mistake location, providing guidance, and actionability. In this work, we focus on the Mistake Identification sub-task, which involves determining whether a tutor's response acknowledges a student's error within a given conversational context. The task builds upon the unified evaluation taxonomy proposed by (Maurya et al., 2025), which defines key pedagogical dimensions for assessing the effectiveness of AI tutors in mistake remediation scenarios.

In our participation in this task, under the team name TutorMind, we explore the effectiveness of multiple LLMs, including GPT-4o-mini (OpenAI, 2024), Mistral-7B (Mistral-AI, 2023), and Llama-3.1-8B (Meta, 2024), in both zero-shot and fine-tuned settings. To address class imbalance in the dataset, we introduce a data augmentation strategy using the Command-R-plus model (Cohere, 2024) to generate synthetic examples targeting underrepresented classes. Our best-performing model, a fine-tuned variant of GPT-4o-mini trained on the full development dataset, achieved a strict macro-averaged F1 score of 71.63%, ranking second place in the competition.

This study contributes to the growing body of research on AI-assisted education by demonstrating how targeted fine-tuning can enhance LLMs' ability to evaluate the pedagogy of tutor LLMs. Our findings underscore the importance of aligning model training with domain-specific evaluation criteria. All fine-tuning scripts, evaluation pipelines, and data augmentation prompts, are publicly available for reproducibility and further research.[1]

[1] https://github.com/fatimadekmak/TutorMind-BEA2025

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 1203–1211

2 Related Work

LLM-Powered AI Tutors in Education: Large language models are being increasingly used as AI tutors capable of engaging students in natural dialogue and providing real-time feedback (Wang et al., 2024). In particular, domains like mathematics and programming have seen significant interest due to the structured nature of problems and the importance of identifying student misconceptions early (Daheim et al., 2024).

However, while LLMs demonstrate impressive fluency and general question-answering capabilities, their effectiveness as pedagogical models remains limited. For instance, GPT-4 often reveals answers prematurely, undermining its role as a supportive tutor. Similarly, Gemini and Phi3 struggle with coherence and actionable guidance, highlighting the need for targeted evaluation frameworks that go beyond traditional natural language generation (NLG) metrics (Jurenka et al., 2024).

Tutor LLMs Evaluation Frameworks: Traditional NLG metrics such as BLEU, ROUGE, and BERTScore are insufficient for evaluating AI tutors because they do not account for pedagogical values such as mistake identification, scaffolding, or encouraging tone. Several studies have proposed domain-specific evaluation criteria tailored to educational dialogues.

(Tack and Piech, 2022) introduced a framework assessing AI tutors based on conversational uptake, understanding, and helpfulness. (Wang et al., 2024) extended this by incorporating dimensions such as care, human-likeness, and usefulness. (Daheim et al., 2024) focused on actionability and correctness.

In contrast, (Maurya et al., 2025) proposed a unified taxonomy comprising eight distinct pedagogical dimensions: Mistake Identification, Mistake Location, Revealing of the Answer, Providing Guidance, Actionability, Coherence, Tutor Tone, Human-likeness, The authors also released MR-Bench, a benchmark dataset containing annotated responses from both human and LLM-based tutors, which is a previous version of the dataset being used in the current task.

Use of LLMs as Evaluators: Researchers have explored the use of LLMs themselves as critics or evaluators. Several studies have demonstrated that LLMs like GPT-4 can assess the quality of educational dialogues with moderate agreement compared to human annotators (Koutcheme et al., 2024). In particular, GPT-4 has been used as an automatic judge to evaluate feedback quality in programming education, showing reasonable correlation with expert human evaluations, although it tends to be overly optimistic in its ratings.

Other studies have leveraged LLMs to score classroom instruction or provide actionable insights for teacher coaching (Wang and Demszky, 2023). These works suggest that LLMs can offer scalable and cost-effective evaluation solutions, although they are not yet fully reliable substitutes for human judgment.

Recent efforts underscore both the growing interest in deploying LLMs as AI tutors and the challenges involved in evaluating their pedagogical effectiveness. While LLMs are proficient at generating fluent and coherent responses, their ability to function as effective tutor agents remains limited. Building on the work of (Maurya et al., 2025), we focus on a single pedagogical dimension, mistake identification, and investigate how fine-tuning LLMs can enhance their ability to evaluate tutor responses within this context.

3 Methodology

This section describes the models, dataset preparation, training setup, and augmentation strategy used to address the Mistake Identification sub-task of the BEA 2025 Shared Task.

3.1 Task Setup & Dataset

We utilized the labeled development set provided by the shared task organizers, focusing specifically on the Mistake Identification dimension of AI tutor responses. The dataset contains three class labels indicating whether the tutor's response addressed a student mistake: Yes (1932 instances), No (370), and To some extent (174). This distribution presents a significant class imbalance, with the "Yes" class significantly overrepresented compared to the other two categories (see Appendix A for a breakdown). We observed that this imbalance negatively impacted model performance during initial experiments. This motivated us to implement targeted data augmentation strategies, as discussed in Section 3.4.

To evaluate model behavior under constrained supervision, we partitioned the development set into two subsets using stratified sampling: a **Training Subset** (80%) and an **Validation Subset** (20%). All initial zero-shot and fine-tuning experiments

were conducted using the training subset, while the validation subset served as a held-out test set to guide model selection.

Additionally, all final system submissions were evaluated by the organizers on a separate **Blind Test Set**, for which ground-truth labels were not released. This Blind Test Set was used to compute the official leaderboard scores for the shared task.

3.2 Model Selection

We evaluated the use of multiple large language models as tutor evaluators. GPT-4o-mini (OpenAI, 2024) was chosen for its strong performance and availability for fine-tuning. Mistral-7B (Mistral-AI, 2023) and LLaMA-3.1-8B (Meta, 2024) instruct models were selected as competitive open-source baselines. Larger models were excluded from this study due to computational constraints.

3.3 Fine-tuning Setups

Fine-tuning experiments on the Mistral-7B and LLaMA-3.1-8B models were carried out using the Unsloth framework, which enables optimized and efficient training through 4-bit quantization and the integration of LoRA adapters. Both models were trained for a total of three epochs with a learning rate of 2e-4 and the AdamW optimizer. The training process was conducted on Google Colab [2], leveraging the range of available GPU resources, including A100 and T4 GPUs with high memory capacity, to ensure stable and efficient execution.

GPT-4o-mini, in contrast, was fine-tuned via the OpenAI platform[3] using supervised fine-tuning (SFT). The training data was formatted into JSONL files with role-tagged messages and associated classification targets (Yes/No/To Some Extent), following OpenAI's SFT guidelines. Prompt templates and formatting details for all models are provided in appendix C and D.

3.4 Data Augmentation

After initially fine-tuning our selected models on the training subset, we observed a noticeable discrepancy between strict and lenient scores (see Section 5 for further discussion). The models frequently confused the No and To some extent classes with Yes, indicating that class imbalance was a limiting factor. This motivated a data augmentation step focused on these underrepresented classes. We generated additional training examples for the No and To some extent classes using the Command R+ model (Cohere, 2024). This model was selected because it was neither involved in generating the original tutor responses nor used in the evaluation pipeline, and was capable of producing high-quality tutor response that follow the given instruction.

We created 100 synthetic examples per underrepresented class. Each instance was manually reviewed for label correctness and consistency with the shared task's annotation guidelines. These examples were added to the training subset and used in a second round of fine-tuning. We refer to this expanded dataset as the **augmented training subset** throughout the paper.

During manual inspection, most generated responses appeared to match the intended labels. The "To some extent" examples typically followed the prompt instructions, using cautious or indirect language like "maybe," "I think," or "let's double-check", without clearly identifying a mistake. For the "No" class, most responses were affirming and feedback-neutral, as expected. However, some responses included subtle hints that could be interpreted as uncertainty, making them closer in tone to the "To some extent" label. These cases were not filtered out as we prioritized maintaining class coverage. In retrospect, these borderline cases introduced some mild label noise, which highlights the need for more precise quality control in future augmentation steps.

The original and augmented training setups shared identical hyperparameter settings. The prompt used with Command R+ to generate data is documented in appendix E.

4 Results

We report results on both the held-out dev test set and the official shared task test set. Table 1 summarizes the accuracy and macro F1 scores under both strict and lenient settings. Our discussion focuses on strict F1, which was the official evaluation metric.

Zero-shot results show that both GPT-4o and Mistral-7B performed reasonably well out of the box (strict F1: 52.13% and 51.73% respectively), while LLaMA-3.1-8B struggled in the absence of fine-tuning, scoring only 19.03%. These results highlight the limitations of zero-shot prompting, particularly for minority class detection.

Fine-tuning on the initial training subset signif-

[2] https://colab.google/

[3] https://platform.openai.com/docs/overview

icantly improved performance across all models. GPT-4o achieved 68.20% strict F1 on the dev test set and was submitted as our first system, scoring 67.70% on the official blind test set. Mistral-7B and LLaMA-3.1-8B achieved 62.61% and 41.52%, respectively. Based on these results, we selected GPT-4o for further fine-tuning on the full development set. GPT-4o fine-tuned on the full development set scored 71.63% on the blind test, ranking second in the competition.

To evaluate the impact of data augmentation, we fine-tuned both GPT-4o-mini and Mistral-7B on the augmented training subset, which included synthetic examples targeting the underrepresented "No" and "To some extent" classes. LLaMA-3.1-8B was excluded from this stage, as it consistently underperformed compared to the other models in earlier experiments. Both GPT-4o-mini and Mistral-7B showed further gains: GPT-4o-mini reached 70.34% strict F1 on the dev test set, while Mistral-7B improved from 62.61% to 70.08%. These configurations were submitted as additional runs, with GPT-4o-mini achieving 70.76% on the blind test set. Notably, the augmented GPT-4o-mini model outperformed all other models trained only on the training subset. However, it was never fine-tuned on the full development set due to time constraints. As a result, it was not submitted in its optimal form. We hypothesize that combining data augmentation with full-devset fine-tuning would have yielded even stronger results, potentially surpassing our best-performing submission (GPT-4o-mini fine-tuned on the full devset without augmentation), which scored 71.63% on the blind test. The relatively lower leaderboard score of the augmented GPT-4o-mini model reflects the limitation of training on a smaller portion of the data, rather than a shortcoming of the augmentation strategy itself. The complete comparison is presented in Table 1.

5 Analysis

The Mistake Identification task was evaluated under two settings: strict and lenient. In the strict setting, the model deals with the three classes, Yes, No, or To some extent, separately. On the other hand, the lenient setting merges the Yes and To some extent labels into a single positive class. This reduces the penalty for confusing pedagogical distinctions, specifically partial vs. full mistake recognition.

As shown in Table 2, lenient scores were consistently higher than strict scores across all models and configurations. For instance, our GPT-4o-mini model fine-tuned on the training subset achieved a strict F1 of 68.20% but a lenient F1 of 87.53%, suggesting that the model often detected the presence of a mistake but occasionally failed to clearly distinguish between full and partial mistake identification. Similarly, Mistral-7B's results reinforce this observation, with 62.61% strict F1 and 85.71% lenient F1.

These results, along with careful examination of model predictions, had two key implications during system development. First, they highlighted that model failures were frequently due to confusion between Yes and To some extent, rather than between positive and negative classes (Yes/To some extent vs. No). This informed our decision to generate targeted augmentations specifically for the No and To some extent classes, which were both underrepresented and prone to misclassification. Second, the wide gap between strict and lenient scores helped us judge whether model improvements were actually sharpening pedagogical judgment, or simply boosting overall correctness.

To better understand the effect of data augmentation, we compare confusion matrices under both strict and lenient settings for the GPT-4o-mini model (Appendix B). In the lenient setting, slight improvements are observed after augmentation, but the gains are minimal—likely due to the small scale of augmentation relative to the underlying class imbalance. Under the strict setting, a few additional instances from the "No" and "To some extent" classes were correctly classified, confirming that the augmentation was directionally helpful. However, we also observe increased confusion within the "Yes" class, suggesting that the added synthetic data may have introduced mild noise. These trends indicate that while small-scale augmentation can be beneficial, its impact is limited and should be expanded or refined in future work.

6 Conclusion

In this work, we addressed the Mistake Identification sub-task of the BEA 2025 Shared Task, which evaluates whether AI tutors recognize student errors within educational dialogues. We explored both zero-shot and fine-tuned settings across several LLMs, including GPT-4o-mini, Mistral-7B, and LLaMA-3.1-8B. Our best-performing submit-

Model	Method	Validation Subset		Blind Test Set
		Strict F1	Strict Acc.	Strict F1 (submission)
GPT-4o-mini	Zero-shot	52.13	82.86	–
GPT-4o-mini	Fine-tuned on training subset	68.20	88.71	67.70
GPT-4o-mini	**Fine-tuned on Full development set**	–	–	**71.63**
GPT-4o-mini	Fine-tuned on augmented training subset	**70.34**	**88.91**	70.76
Mistral-7B	Zero-shot	51.73	70.16	–
Mistral-7B	Fine-tuned on training subset	62.61	87.10	–
Mistral-7B	Fine-tuned on augmented training subset	70.08	88.51	60.59
LLaMA-3.1-8B	Zero-shot	19.03	29.44	–
LLaMA-3.1-8B	Fine-tuned on training subset	41.52	84.48	–

Table 1: Performance comparison of all models under strict evaluation: The table reports strict macro-F1 and accuracy scores on the internal validation set and the official blind test set. The best-performing submitted model was GPT-4o-mini fine-tuned on the full development set, achieving a strict F1 score of 71.63% on the blind test.

Model	Method	Strict F1	Strict Acc.	Lenient F1	Lenient Acc.
GPT-4o-mini	Zero-shot	52.13	82.86	77.57	89.52
GPT-4o-mini	Fine-tuned on training subset	68.20	88.71	87.53	93.95
GPT-4o-mini	Fine-tuned on augmented training subset	**70.34**	**88.91**	**88.36**	**94.35**
Mistral-7B	Fine-tuned on training subset	62.61	87.10	85.71	92.74
Mistral-7B	Fine-tuned on augmented training subset	70.08	88.51	87.15	93.55

Table 2: Macro-F1 and accuracy scores are shown for both strict (3-way classification) and lenient (binary classification: Yes/To some extent vs. No) settings on the validation subset. GPT-4o-mini fine-tuned on the augmented training dataset performs best on the validation subset in both settings.

ted system, a GPT-4o-mini model fine-tuned on the full development set, achieved a strict macro-F1 score of 71.63% on the official blind test set, ranking second in the competition. These results highlight the value of lightweight fine-tuning in enhancing LLMs' pedagogical sensitivity. Our findings support the ongoing effort to make LLM-based tutors not only fluent but diagnostically effective, capable of recognizing learner misconceptions and delivering instruction that aligns with educational goals.

7 Limitations

While our approach yielded strong results on the Mistake Identification sub-task, several limitations remain. First, the scale of training data, particularly for the "No" and "To some extent" classes, was limited, Although synthetic augmentation improved model calibration, manual inspection of the generated examples was relatively permissive. In particular, some "No" examples included subtle guidance or hints that could blur the boundary with the "To some extent" class, introducing mild label noise. These were not filtered out during data selection and may have affected label consistency. Future work should explore more grounded augmentation strategies, along with stricter validation procedures to ensure correct label alignment.

Moreover, the models we used for evaluation in our study were also among those used to generate tutor responses for the development data. This overlap introduces potential bias, as models could be more inclined to align with responses produced by themselves or closely related variants. This

type of alignment can lead to overestimation of pedagogical quality of the tutor response.

References

Cohere. 2024. Command r+ documentation. Accessed: 2025-05-21.

Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. *Preprint*, arXiv:2407.09136.

Irina Jurenka, Markus Kunesch, Kevin R. McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, Ankit Anand, Miruna Pîslar, Stephanie Chan, Lisa Wang, Jennifer She, Parsa Mahmoudieh, Aliya Rysbek, Wei-Jen Ko, Andrea Huber, and 55 others. 2024. Towards responsible development of generative ai for education: An evaluation-driven approach. *Preprint*, arXiv:2407.12687.

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

Charles Koutcheme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, and Paul Denny. 2024. Open source language models can provide feedback: Evaluating llms' ability to help students using gpt-4-as-a-judge. *Preprint*, arXiv:2405.05253.

Kaushal Kumar Maurya, KV Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors. *Preprint*, arXiv:2412.09416.

Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Mistral-AI. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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

Rose E. Wang and Dorottya Demszky. 2023. Is chatgpt a good teacher coach? measuring zero-shot performance for scoring and providing actionable insights on classroom instruction. *Preprint*, arXiv:2306.03090.

Rose E. Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. *Preprint*, arXiv:2310.10648.

A Development Set Class Distribution

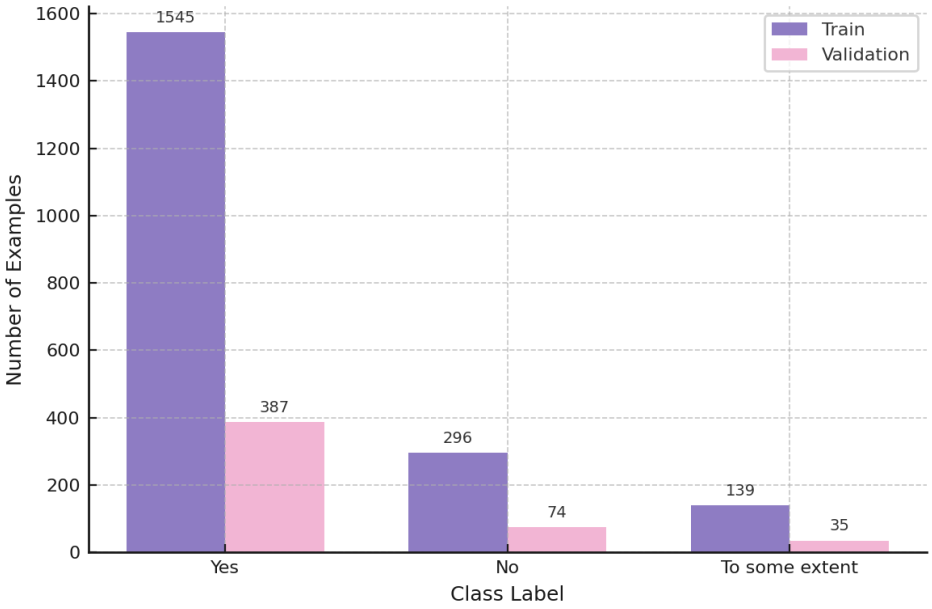

Figure 1: Class distribution in the original development set, split by training and validation subsets. This shows the class imbalance in the provided training data, motivating data augmentation.

B Confusion Matrices

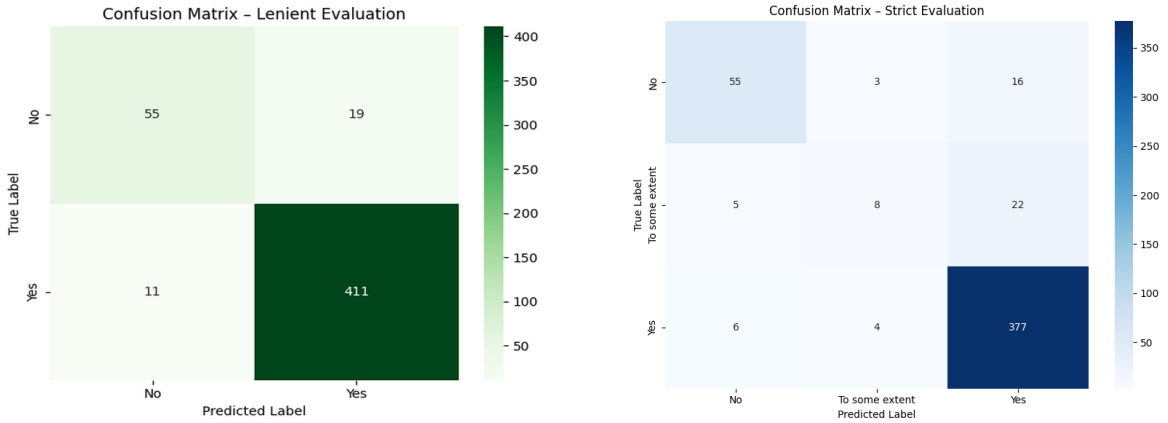

Figure 2: Confusion matrices for GPT-4o-mini fine-tuned on the original training subset.

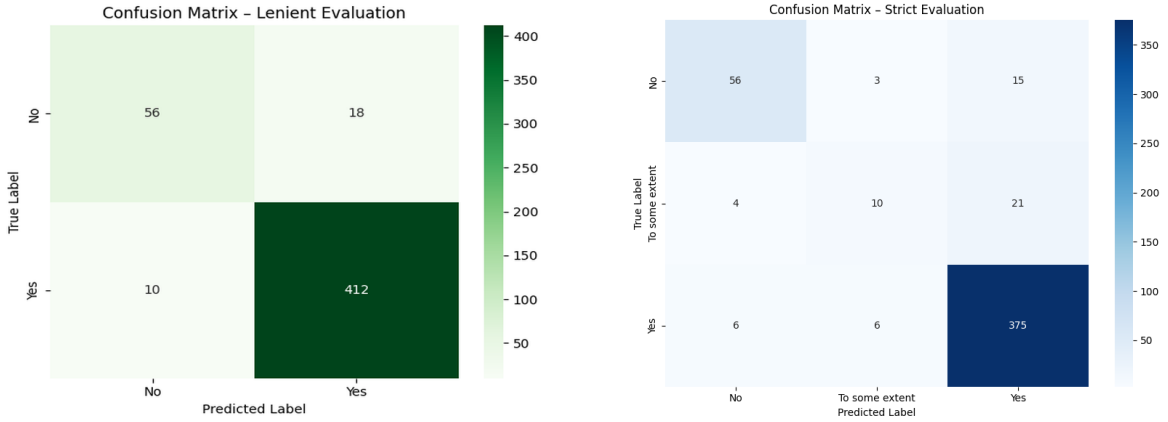

Figure 3: Confusion matrices for GPT-4o-mini fine-tuned on the augmented training subset.

C Prompt for Llama3.1 8B Instruct and Mistral 7B Instruct

Prompt Template

Instruction:
Evaluate the tutor's response based on whether they identified a mistake in the student's response or not. Mistake Identification: Has the tutor identified a mistake in the student's answer? Options:Yes, To some extent, No. Yes means the mistake is clearly identified or recognized in the tutor's response. No means the tutor does not recognize the mistake (e.g., they proceed to simply provide the answer to the asked question). To some extent means the tutor's response suggests that there may be a mistake, but it sounds as if the tutor is not certain. You should answer by Yes, No or To some extent strictly in the following format: Evaluation: (Yes, No, To Some Extent). It is very important to have the word Evaluation: before your answer, while also sticking to the criteria of evaluation.

Input:
{Conversation History + Tutor Response}

Response:
Evaluation: {Yes, No, or To Some Extent}

D Prompt for GPT-4o-mini

Prompt Format

System Message:
Classify the tutor's response to the student's answer based on whether the tutor has identified a mistake. Use the following labels: 'Yes' means the mistake is clearly identified; 'No' means the tutor does not recognize the mistake; 'To some extent' means the tutor suggests a mistake but is unsure. Respond strictly in the format: Evaluation: [Yes/No/To Some Extent].

User Message:
{Conversation History + Tutor Response}

Expected Output:
Evaluation: {Yes, No, or To Some Extent}

E Prompt for Data Augmentation with Command R+

Prompt for Generating "To Some Extent" Responses

Instruction:
You are a math tutor giving feedback to a student. Based on the conversation, write a single-sentence response that gently suggests the student may have made a mistake, but without clearly identifying what the mistake is. Your tone should sound uncertain, cautious, or exploratory. Do not explicitly say what is wrong. Do not state that something is definitely incorrect. Keep your response to ONE short sentence.

Input:
{Conversation History}

Output:
A single-sentence tutor response labeled "To some extent"