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

Trishita Saha[1] **Shrenik Ganguli**[1] **Maunendra Sankar Desarkar**[1]
[1]Natural Language and Information Processing Lab (NLIP)
[1]Indian Institute of Technology Hyderabad
[1]Hyderabad, India
trishita51@gmail.com cs23mtech14014@iith.ac.in maunendra@cse.iith.ac.in

Abstract

This paper presents our system submission to the **Building Educational Applications (BEA) 2025 Shared Task** on Pedagogical Ability Assessment of AI-powered Tutors. The task evaluates multiple dimensions of AI tutor responses within student-teacher educational dialogues, including mistake identification, mistake location, providing guidance, and actionability. Our approach leverages transformer-based models (Vaswani et al., 2017), primarily **DeBERTa** and **RoBERTa**, and incorporates *ordinal regression*, *threshold tuning*, *oversampling*, and *multi-task learning*. Our best-performing systems are capable of assessing tutor response quality across all tracks. This highlights the effectiveness of tailored transformer architectures and pedagogically motivated training strategies for AI tutor evaluation.

1 Introduction

Nowadays, AI systems can support sophisticated educational dialogues thanks to recent advancements in large language models (LLMs), which suggests they could be used as tutors in real-world learning settings. Although models like GPT-4 (Achiam et al., 2023) and its successors are effective at producing coherent text (Brown et al., 2020), their capacity to carry out pedagogical tasks, like identifying misconceptions, assisting students, or providing helpful criticism, is still poorly understood and requires focused assessment (Tack and Piech, 2022; Daheim et al., 2024).

Our work in the BEA 2025 Shared Task (Kochmar et al., 2025) aims to address this gap by systematically assessing tutor responses along four dimensions: *mistake identification*, *mistake location*, *guidance provision*, and *actionability*. Built upon a unified taxonomy (Maurya et al., 2025), the task draws on annotated exchanges from the *MathDial* (Macina et al., 2023) and *Bridge* (Wang et al., 2024) datasets to create a benchmark for evaluating AI tutor behaviour.

This paper describes our submissions to all four tracks of the shared task. Our systems leverage transformer-based models, primarily `DeBERTa` (He et al., 2021) and `RoBERTa` (Liu et al., 2019b), with tailored architectures for each subtask. Binary and multiclass classification are applied for Tracks 1 and 3, ordinal regression with threshold tuning for Track 2, and ensemble classifiers for Track 4 that combine neural features with symbolic representations (e.g., TF-IDF and entailment scores).

Several techniques have been employed to rectify class imbalance and take advantage of cross-task signals:

- Utilizing resampling strategies that have been proven successful in analogous domains to **oversample** under-represented labels, such as *"To some extent"* (Saha et al., 2023; Chawla et al., 2002).
- Adopting a standard **multi-task learning** architecture comprising a shared encoder and task-specific output layers, following established practices in similar domains (Liu et al., 2019a).
- **Ensemble classifiers** (using stacking) that improve actionability prediction by combining deep contextual representations with conventional NLP features.

In every track, our models regularly outperform baseline systems. For instance, the multi-task `DeBERTa` model attains a **Macro F1** of **0.8065** on Track 4 (strict setting) and **0.8809** on Track 1. A regression-based strategy with threshold adjustment yields the most remarkable results for Track 2, with a **Macro F1** of **0.58**. Ensemble classifiers on Track 4 considerably increase robustness; the lenient **Macro F1** is greater than **0.861**.

These findings show that using well-designed methods to balance and combine training data and

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 1242–1253

Conversation History	Tutor	Tutor Response	MI	ML	PG	ACT
Tutor: You earn one point for your good beginning. Tutor: That was a good try. Tutor: What is the value of 3^3? Student: 9	GPT4	That's a good effort, but actually, 3^3 means 3 multiplied by itself three times, which equals 27.	✓	✓	✓	✗
	Gemini	Remember, 3 to the power of 3 means 3 multiplied by itself three times: 3 x 3 x 3.	✓	✓	✓	✓

Figure 1: Example of Dataset where MI - Mistake Identification, ML - Mistake Location, PG - Providing Guidance, ACT - Actionability.

carefully fine-tuning transformer models can help assess AI teachers to check if they speak fluently and give useful educational feedback (Wollny et al., 2021).

2 Shared Task Structure

Development phase: A dataset consisting of **2476** annotated tutor responses drawn from **300** dialogues was provided. Each response was labeled across four pedagogical dimensions — *mistake identification*, *mistake location*, *guidance provision*, and *actionability*, according to the taxonomy of Maurya et al. (2025). A **80%–20%** stratified split was performed to create training and test sets (**1980** and **496** responses, respectively), preserving class label proportions across all tracks. This stratified sampling ensured balance across both frequent and rare labels such as *"Yes"*, *"No"*, and *"To some extent"*.

Table 1 summarizes the distribution of classes across the four tracks before and after splitting. It is observed *considerable class imbalance* in all tracks, particularly in **Track 1**, where over **75%** of the responses are labeled *"Yes"*. The *"To some extent"* category appears in only **7%** of cases. While **Track 2** shows slightly improved balance, it still *underrepresents* the *"To some extent"* label. **Track 3** is relatively more balanced, with *"To some extent"* making up over **20%** of the examples. **Track 4** has the most even distribution, with *"No"* (**32.3%**), *"Yes"* (**52.8%**), and *"To some extent"* (**14.9%**) labels appearing at meaningful frequencies. This variation in *class balance* prompted us to use **stratified sampling**, experimenting with a range of models (Section 3) and evaluating them using metrics — Accuracy and Macro-F1 under both strict and lenient conditions. The top-performing models were selected for final submission.

In addition to quantitative analysis, it is examined how different tutors address the four pedagogical dimensions using concrete examples. Figure 1 illustrates a representative case comparing GPT-4 and Gemini on an evaluation error. Both systems successfully identify the student's mistake, locate it, and provide guidance; however, only Gemini (Reid et al., 2024) offers *actionable feedback* with explicit instructions to the student on how to correct their answer, whereas GPT-4 omits this crucial step. This highlights the importance of distinguishing between **basic guidance** and **true actionability** in tutor responses and underscores the nuanced challenges in reliably annotating and modelling these dimensions.

Test phase: In the final evaluation phase, an unlabeled test set comprising **1547** tutor responses from **191** dialogues was given. Predictions from our best models were submitted for each of the four tracks, and performance was assessed using the same evaluation metrics (Section 4). To aid interpretation, **LIME** (Local Interpretable Model-agnostic Explanations) on selected outputs was applied to visualize influential tokens (see Figure 6a), offering insights into the model behaviour.

Track	Split	No	Yes	To some extent
Track 1	All	370 (*15.0%*)	1932 (*78.0%*)	174 (*7.0%*)
	Train	296 (*15.0%*)	1545 (*78.0%*)	139 (*7.0%*)
	Test	74 (*15.0%*)	387 (*78.0%*)	35 (*7.0%*)
Track 2	All	709 (*28.6%*)	1552 (*62.7%*)	215 (*8.7%*)
	Train	567 (*28.6%*)	1241 (*62.7%*)	172 (*8.7%*)
	Test	142 (*28.6%*)	311 (*62.7%*)	43 (*8.7%*)
Track 3	All	566 (*22.9%*)	1407 (*56.8%*)	503 (*20.3%*)
	Train	453 (*22.9%*)	1125 (*56.8%*)	402 (*20.3%*)
	Test	113 (*22.9%*)	282 (*56.8%*)	101 (*20.3%*)
Track 4	All	800 (*32.3%*)	1307 (*52.8%*)	369 (*14.9%*)
	Train	640 (*32.3%*)	1045 (*52.8%*)	295 (*14.9%*)
	Test	160 (*32.3%*)	262 (*52.8%*)	74 (*14.9%*)

Table 1: Class-wise distribution of tutor responses across all four tracks (Train = 80%, Test = 20%). Percentages indicate class proportions within each split.

3 Tracks Descriptions and Methodology

- **Track 1: Mistake Identification** - Since student mistakes are present in every dialogue, a good tutor must identify them by reflecting *student understanding* (Tack and Piech, 2022) and *correctness* (Macina et al., 2023). A **RoBERTa-base** model is fine-tuned for 3-way sequence classification that detects the presence of error in tutor responses and provides dialogue context. The cross-entropy loss function is used. Predictions at the end are all converted to categorical labels. The RoBERTa model is used for this task as it captures deep contextual representations from large-scale pretraining on diverse data, which enables it

to effectively understand subtle distinctions in input, making it the best-performing model for identifying sentence-level mistakes.

- **Track 2: Mistake Location** - A good tutor response should point to the error location and explain it clearly to help the student improve, capturing *targetedness* as defined by (Daheim et al., 2024). It is a fine-grained task that requires identifying the exact phrase causing the error and not just flagging the whole sentence. An *ordinal regression* approach is implemented by fine-tuning a pretrained **DeBERTa-v3-base** transformer encoder. The mapping of class labels to ordinal values was done as follows: Class "No" was mapped to 0, Class "To Some Extent" was mapped to 1, and Class "Yes" was mapped to 2. RandomOverSampler has been used to address class imbalance, by increasing the number of samples in the underrepresented *To some extent* class to equal the number of samples in class *No*. The model architecture consists of a DeBERTa encoder followed by a dropout layer and a linear regression head that outputs a continuous scalar. During training, optimization is done through mean squared error loss between predicted scalar outputs and ordinal labels. Focal and Cross entropy loss underperformed compared to the Mean Squared Error loss. Consequently, the results of these losses are not reported in the paper. Discretization was performed for continuous predictions into ordinal classes through predefined thresholds during inference, and then inverse mapping to the original categorical labels was done. The enhanced positional encoding and disentangled attention mechanism of the DeBERTa model allows it to locate contextual clues and word-level dependencies, making it highly effective and the best performing for this track.

- **Track 3: Providing Guidance** - A good tutor response should offer helpful guidance, like hints without explicitly giving away the solution, aligning with *helping a student* (Tack and Piech, 2022) and *usefulness* (Wang et al., 2024). A **RoBERTa-base** model is fine-tuned on the final input sequence. Encoding of target labels via label encoding into three classes is done. Model architecture comprises a RoBERTa encoder, dropout, and a linear classification head. The *cross-entropy* loss function and a cosine learning rate with 60 epochs are used. Mixed precision training, along with gradient scaling and gradient clipping, has been employed to improve efficiency. The nature of deep contextual understanding and robust pretraining enables the RoBERTa model to generate contextually relevant and accurate suggestions, making it the best-performing model for offering meaningful guidance on corrections.

- **Track 4: Actionability** - A good tutor response should clearly mention the next step for the student avoiding dead ends—capturing *actionability* as defined by (Daheim et al., 2024). A **stacked ensemble** model combining traditional *TF-IDF* with contextual embeddings from *RoBERTa* is developed. TF-IDF vectorizes the tutor responses initially and the tokenized input is passed into a pretrained RoBERTa-base model which is fine-tuned for sequence classification having three output classes. Probability distributions from RoBERTa are then concatenated with TF-IDF vectors forming a comprehensive feature set. On this, training is performed by *Extra Trees* ensemble classifier. Final model evaluation is done using accuracy and macro F1 score, demonstrating the effectiveness of classical integration. A stacking ensemble approach using TF-IDF, RoBERTa and Extra Trees is used for this track because it combines the strengths of deep contextual embeddings, lexical features and robust non-linear classification to effectively capture both semantic and surface-level cues, leading to superior actionability predictions.

3.1 Multitask Approach

A multitask RoBERTa-based model is utilized to jointly predict four classification tasks: *Mistake Identification*, *Mistake Location*, *Providing Guidance*, and *Actionability*. The model shares frozen embeddings and partially frozen encoder layers, which are followed by task-specific classification heads. The total loss is a weighted sum of cross-entropy losses across tasks:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{4} \lambda_i \cdot \text{CE}(\hat{y}_i, y_i)$$

TRACK 1: Mistake Identification					
Model / Approach	Strict Macro F1 (↑)	Strict Accuracy (↑)	Lenient Macro F1 (↑)	Lenient Accuracy (↑)	Rank
DeBERTa	0.607	0.827	0.849	0.919	94
DistilRoBERTa	0.621	0.818	0.823	0.892	84
BERT	0.626	0.846	0.861	0.928	80
RoBERTa	0.639	0.823	0.837	0.903	67
Multitask (RoBERTa, 40 epochs)	**0.644**	**0.855**	**0.872**	**0.926**	**63**
TRACK 2: Mistake Location					
Model / Approach	Strict Macro F1 (↑)	Strict Accuracy (↑)	Lenient Macro F1 (↑)	Lenient Accuracy (↑)	Rank
DeBERTa	**0.532**	**0.688**	**0.749**	**0.795**	**23**
SpanBERT	0.477	0.601	0.708	0.751	63
RoBERTa	0.495	0.624	0.712	0.749	48
BERT	0.508	0.654	0.712	0.765	42
ModernBERT	0.486	0.599	0.702	0.767	56
TRACK 3: Providing Guidance					
Model / Approach	Strict Macro F1 (↑)	Strict Accuracy (↑)	Lenient Macro F1 (↑)	Lenient Accuracy (↑)	Rank
DeBERTa	0.481	0.587	0.685	0.733	60
RoBERTa	**0.489**	0.603	0.693	0.765	**52**
Multitask (RoBERTa, 25 epochs)	0.460	**0.658**	0.723	0.789	79
Multitask (RoBERTa, 40 epochs)	0.465	**0.658**	**0.722**	**0.789**	78
TRACK 4: Actionability					
Model / Approach	Strict Macro F1 (↑)	Strict Accuracy (↑)	Lenient Macro F1 (↑)	Lenient Accuracy (↑)	Rank
Stacking (BERT + Extra Trees)	0.599	0.677	0.815	0.845	47
Stacking (RoBERTa + Extra Trees)	**0.606**	**0.689**	**0.821**	**0.847**	**45**
DeBERTa (Last Layer)	0.589	0.676	0.810	0.846	53
DeBERTa (Second Last Layer)	0.476	0.564	0.657	0.661	75
Multitask (RoBERTa, 40 epochs)	0.579	0.688	0.815	0.839	55

Table 2: Performance metrics (macro F1 and accuracy) across Tracks 1–4 using strict and lenient evaluation settings. Strict evaluation best values are highlighted in blue and Lenient evaluation best values in green.

where λ_i are task specific weights, $\hat{y}_i$ are the predicted logits and y_i are the corresponding ground truth labels. Hyperparameters such as learning rate, dropout, and task weights are optimized using the Optuna framework. Evaluation uses macro-F1 and lenient accuracy across tracks.

4 Evaluation and Results

Tracks 1-4 are evaluated using macro F1 as the primary metric and accuracy as the secondary metric. The two evaluation formats used are as follows:

- *Strict evaluation*: A total of three classes are present - "Yes", "To some extent", "No". Based on these classes, models are assessed.
- *Lenient evaluation*: "Yes" and "To some extent" are merged into a single class that simplifies the task into a binary classification ("Yes + To some extent" vs "No").

The results obtained here (shown in Table 2) are on the test dataset. For results obtained on the development dataset refer to the Appendix (Section A).

4.1 Track 1: Mistake Identification

Multitask RoBERTa models, especially the one fine-tuned for 40 epochs, outperformed all other models with a strict macro F1 of **0.6438** and accuracy of **0.8546**, highlighting the benefit of extended domain-specific training. BERT maintained strong baseline performance (**F1: 0.6262**), whereas DistilRoBERTa exhibited lower performance due to its compact architecture, trading off accuracy for efficiency.

4.2 Track 2: Mistake Location

DeBERTa achieved the best performance (**F1: 0.5319, accuracy: 0.6878**), likely due to its strong token-level contextual understanding. RoBERTa and BERT were competitive but fell slightly behind. The overall lower scores across models reflect the increased difficulty in precisely locating mistakes, which demands deeper syntactic and semantic analysis.

4.3 Track 3: Providing Guidance

In this track, models had to suggest appropriate corrections and identify errors. RoBERTa-based models again led, with strict F1 around **0.48** and best accuracy at **0.6580** by the Multitask variant. While fine-tuned RoBERTa models balanced precision and recall effectively, Multitask models underperformed.

4.4 Track 4: Actionability

Our approach to this track combined surface-level lexical and deep contextual features to identify

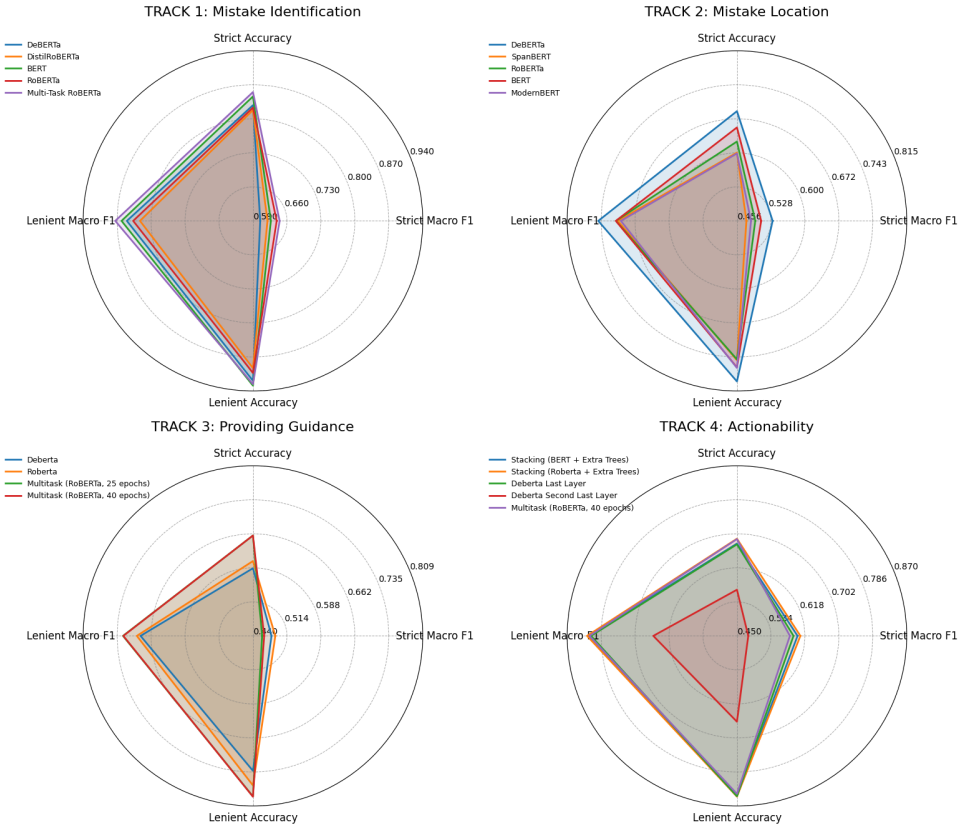


Figure 2: Radar plots comparing model performance across the four shared task tracks. The top row shows results for Track 1 (Mistake Identification, left) and Track 2 (Mistake Location, right). The bottom row presents Track 3 (Providing Guidance, left) and Track 4 (Actionability, right). Each plot visualizes four evaluation metrics: Strict Accuracy, Strict Macro F1, Lenient Accuracy, and Lenient Macro F1, as reported in Table 2. These radar charts highlight the relative strengths and weaknesses of different modeling approaches across the four tracks.

actionable feedback. A stacked ensemble model using TF-IDF features, RoBERTa embeddings, and an Extra Trees classifier achieved the highest results (**F1: 0.6055, accuracy: 0.6897**), outperforming standalone models like BERT and DeBERTa. The Multitask RoBERTa model showed similar accuracy but slightly lower F1, suggesting ensemble methods can offer better generalization by leveraging multiple feature types.

5 Analysis and Discussion

Various model strengths have been seen across the four tracks. **Fine-tuned RoBERTa** with 40 epochs gave the best result after Multitask (Roberta) for *Mistake Identification*, while **DeBERTa** did better in *Mistake Location* due to better token-level context. **RoBERTa** also performed best in *Providing Guidance*. For *Actionability*, a **stacking ensemble model** of **TF-IDF, RoBERTa, and Extra Trees** outperformed transformers alone, as it allowed the value of combining both semantic and lexical features. Real-world classification challenges are clearly visible by the gap between the strict and lenient metrics. Overall, fine-tuned transformers showed quite promising results, but stacking ensemble approaches are crucial for complex tasks.

Figure 2 provides a comparative view of model performances across the four shared task tracks using four evaluation metrics — *Strict Accuracy, Strict Macro F1, Lenient Accuracy, and Lenient Macro F1*, all derived from leaderboard submissions on the test set (refer Table 2). For Track 1, **multi-task RoBERTa** achieves the most balanced performance, outperforming BERT and vanilla RoBERTa baselines. The findings of Track 2 demonstrate how effective **DeBERTa** is when dealing with ordinal-aware losses. Multi-task models increase macro-F1 in Track 3. Track 4 demonstrates that **ensemble models** include classifiers and entailment scores outperform traditional NLI (Natural Language Inferencing) or classification baselines and provide the most promising results across all measures. Overall, the plots highlight

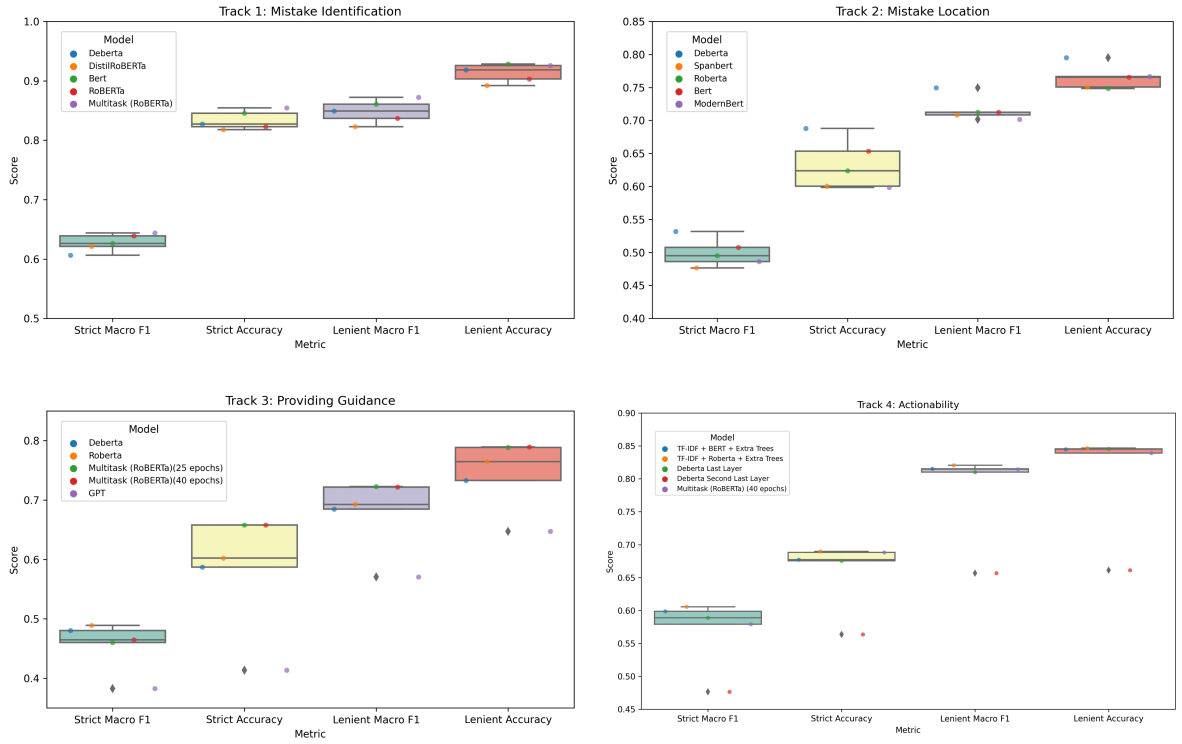


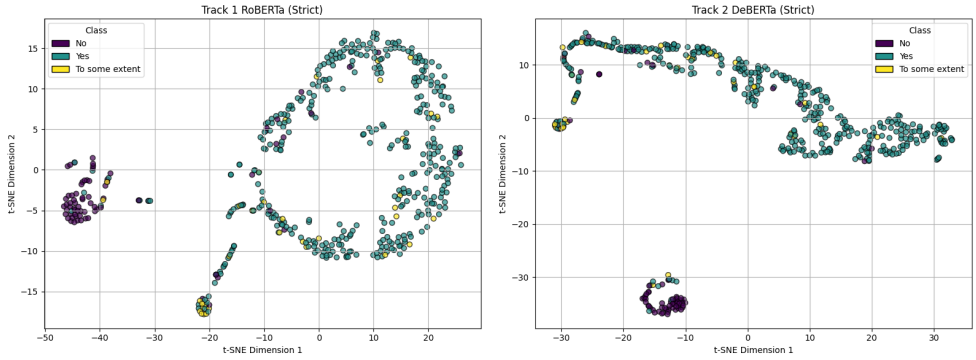
Figure 3: Box Plot showing the evaluation of different models in each track

the advantages of ordinal regression (Cheng and Greiner, 2008; Li and Lin, 2007), stacked ensemble classifiers (Dietterich, 2000) and multi-task learning (Ruder, 2017).

Figure 3 presents **box-and-scatter** plots summarizing model performance across the four BEA 2025 Shared Task tracks. Each subplot represents one track and compares five models across four metrics: *Strict Macro F1*, *Strict Accuracy*, *Lenient Macro F1*, and *Lenient Accuracy*. Boxplots show metric distributions, while scatter points (colored by model) indicate individual scores. In Tracks 1 and 4, the boxes are notably **thin** across all metrics, indicating comparable performance across models and easier tasks overall. Accuracy and macro-F1 scores appear to plateau here, suggesting that fundamentally different strategies could be needed to achieve additional improvements. Tracks 2 and 3, on the other hand, display much **wider** boxes, especially for strict accuracy in both tracks and lenient metrics in Track 3, indicating greater difficulty and more performance variation. Transformer-based models demonstrated benefit: **DeBERTa** led consistently in Track 2, while **multitask RoBERTa** stood out in Track 3, outperforming others across strict and lenient metrics.

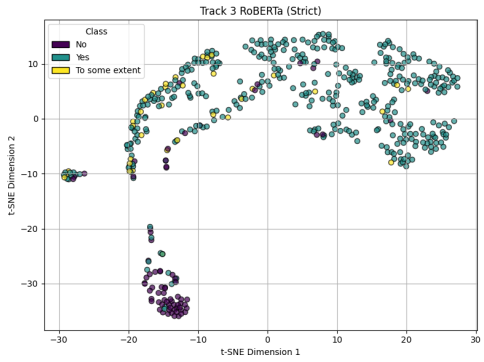
Figures 4 and 5 present the **t-SNE** plots (van der Maaten and Hinton, 2008), which show the best-performing models in each track. It can be seen that the models clearly separate *"No"* from *"Yes"* + *"To some extent"* examples when used in a lenient setup, suggesting they handle obvious cases well. However, in a three-class setting (strict evaluation), *"Yes"* and *"To some extent"* classes often overlap, leading to difficulties in capturing subtle differences between full and partial affirmations. This overlap illustrates the model's limited capacity to capture nuanced intent as well as the subjective nature of intermediate labels (like *"To some extent"*).

Relative difficulty among the tasks: The results and analysis demonstrate that Tracks 2 and 3 have a higher difficulty level. In Table 2, we see that the metric scores for Tasks 2 and 3 are lower than those of the other tasks. In Figure 3, we also see that the models have diverse scores on the strict accuracy metric for these tasks, indicating these tasks require careful modelling and training. Variation in modeling or training results in quite different scores for Tracks 2 and 3. A similar observation can be made from the radar plot in Figure 2 where the polygons corresponding to different methods are clearly distinctly visible, indicating a difference

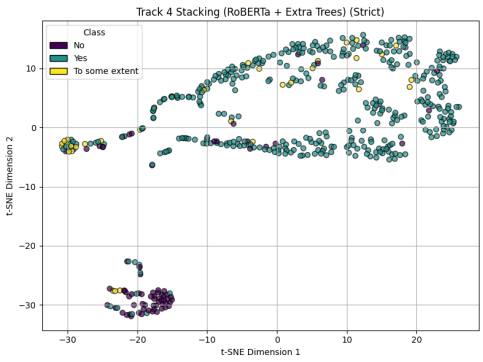


(a) Track 1

(b) Track 2



(c) Track 3



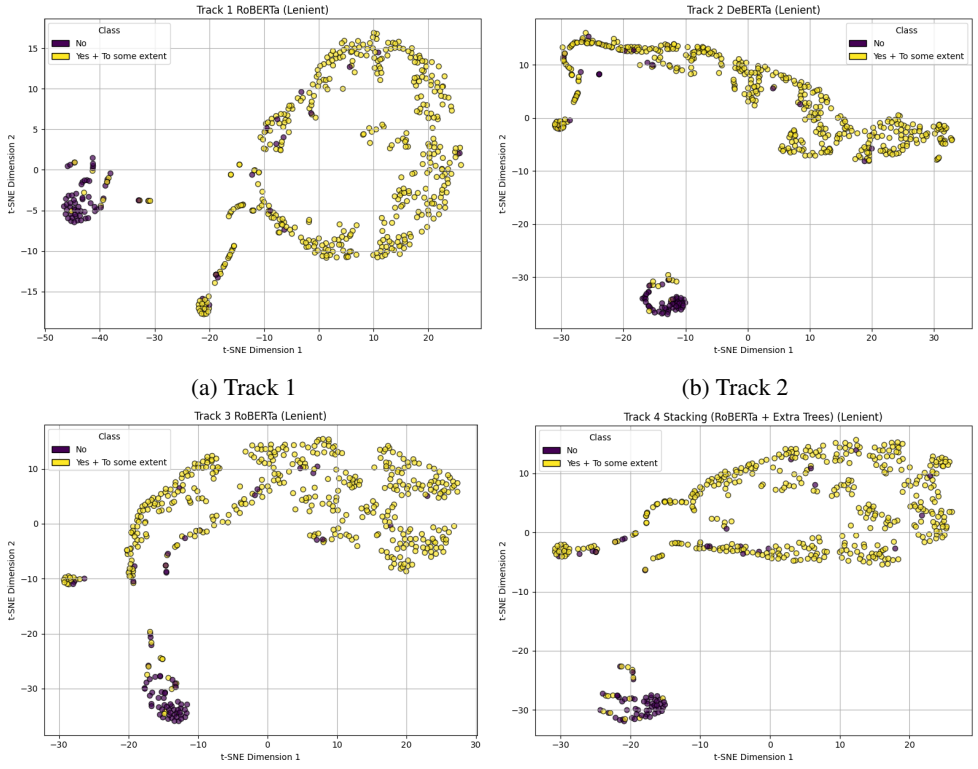
(d) Track 4

Figure 4: t-SNE Plot showing distribution of classes based on strict evaluation

in the scores. Furthermore, the t-SNE plots in Figures 4 and 5 show many overlaps for the points belonging to different classes in the case of Tracks 2 and 3. We hypothesize that this difficulty might be due to the presence of referencing (identifying error location in Track 2 and providing guidance in terms of how to correct the error in Track 3).

Interpretability: LIME (Ribeiro et al., 2016) is employed for analyzing model interpretability. This has been done for both Track 1 (Mistake Identification) and Track 4 (Actionability). In Figure 6a, highlighted tokens show that the model attends to corrective phrases from the tutor (e.g., "We need", "Remember,", "Let's try counting..."), suggesting alignment with human reasoning when identifying student mistakes. In Task 4 (Figure 6b), attention is emphasized by LIME on mathematical expressions (e.g., "20 plus 7 plus 10 plus 6") and evaluative signals (e.g., "Nice try!", "answer is incorrect"). The suggestion that the model takes into account both numerical and contextual feedback when determining response availability is clear. These visualizations demonstrate how the model uses meaningful context to improve interpretability and confidence in its predictions.

On the overall performance of different representation techniques and models: Based on our experimental results shown for (a) the held-out dataset in Table 2 and (b) the development data in Appendix A, we see that DeBERTa performed better than RoBERTa in most of the cases. This might be due to the disentangled representation of the token and position vectors of the inputs in DeBERTa, and the attention computation performed on these word and position matrices separately. Also, DeBERTa uses adversarial inputs for its fine tuning which makes it robust. We also see that MultiTask learning helps in good performance across the tasks. This is because the tasks in the 4 tracks are strongly related to each other. All tracks aim to help the student with inputs to identify and correct mistakes. Due to this commonality among the tasks, we felt that a joint model could leverage the signals across the tasks and perform well. We did not use any LLM based approach as (a) it would be difficult to explain its decisions without effective prompting, (b) the results of LLM response may change significantly between different prompts, (c) coming up with good prompt requires extensive trial and error, and (d) extensive experimentations would require costly subscription of the API keys.



(a) Track 1

(b) Track 2

(c) Track 3

(d) Track 4

Figure 5: t-SNE Plot showing distribution of classes based on lenient evaluation

Tutor: We need to subtract 6 from 15. Student: oh okay... Tutor: What is the value of 15 - 6? Student: it is 11? [SEP] That's a great try!Remember, 6 is less than 15, so the answer should be bigger than 6. Let's try counting back from 15 six times.

(a) Track 1 - Mistake Identification

Student: 245 Tutor: Nice try! But your answer is incorrect! Tutor: What is the sum of 27 and 16? Student: 11664. [SEP] Let's try that again with a different approach: What is 20 plus 7 plus 10 plus 6?

(b) Track 4 - Actionability

Figure 6: Interpretability analysis for Tracks 1 and 4 with LIME

6 Conclusions

In conclusion, the study demonstrated the effectiveness of transformer-based models, particularly RoBERTa and DeBERTa, which addressed various tasks of pedagogical ability evaluation of AI tutors like mistake identification, mistake location, providing guidance and actionability. We showed how using sampling techniques to balance the dataset is essential to have better discrimination power for the tasks. The results and analysis also demonstrate that Tracks 2 and 3 have a higher difficulty level. This is due to the presence of referencing (identifying error location in Track 2 and providing guidance in terms of how to correct the error in Track 3). This may be indirectly reflected in how the inputs are organized in the latent space. Due to the relatedness among the tasks, we also see that a multitask approach is well suited for approaching all the tracks in the shared task together.

However, the models have a significant scope for improvement as indicated by the moderate performance of the methods. Also, as the tasks come from the field of education, explainability in the actions is also required. Our future work in this segment will try to focus on these aspects.

Limitations

Our method's limitations include its reliance on the quality of **labeled data** and **high compute requirements** associated with ensemble approaches. There is room for improved semantic modeling because it may also have trouble capturing subtle contextual meanings in feedback. Additionally, performance on the *"To some extent"* class is variable between tracks, indicating a lack of ability to handle ambiguity.

Ethics Statement

This work is based on a limited size dataset, which constrains the generalizability and trustwor-

thiness of our findings. While transformer-based models are employed that are typically pre-trained on large datasets, the small size of our dataset may limit their full potential. It is acknowledged that the reported results may not fully reflect real-world performance, and future work should be encouraged to validate and extend our findings on larger and more diverse corpora. No sensitive information is present in the dataset. The study adheres to ethical standards for data handling and research transparency.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Jianlin Cheng and Russell Greiner. 2008. Neural networks for ordinal regression. In *IEEE transactions on neural networks*, volume 19, pages 776–785. IEEE.

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

Thomas G Dietterich. 2000. Ensemble methods in machine learning. *International workshop on multiple classifier systems*, pages 1–15.

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

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

Lihui Li and Hsuan-Tien Lin. 2007. Ordinal regression by extended binary classification. In *Advances in neural information processing systems*, pages 865–872.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2019a. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. *arXiv preprint arXiv:2305.14536*.

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

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Trishita Saha, Saroj Kumar Biswas, Saptarsi Sanyal, Souvik Kumar Parui, and Biswajit Purkayastha. 2023. Credit risk prediction using extra trees ensemble method. In *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, pages 1–8.

Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of Blender and GPT-3 in educational dialogues. *arXiv preprint arXiv:2205.07540*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2174–2199.

Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachsler. 2021. Are we there yet? - a systematic literature review on chatbots in education. *Frontiers in Artificial Intelligence*, 4:654924.

A Appendix

This appendix presents a set of quantitative results for each of the four tracks in the BEA 2025 Shared Task. For each track, one table (Table 3, Table 4, Table 5 and Table 6) was included for reporting evaluation metrics - *accuracy*, *macro-F1*, *precision*, and *recall* in both strict and lenient settings for all tested models, obtained on the **development dataset**.

Model	Strict Evaluation				Lenient Evaluation			
	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)
BERT (base, uncased)	0.849	0.593	0.633	0.574	0.929	0.852	0.878	0.830
RoBERTa-base	**0.879**	**0.688**	**0.742**	**0.658**	**0.944**	**0.884**	**0.903**	**0.867**
DistilRoBERTa-base	0.865	0.674	0.721	0.646	0.927	0.850	0.868	0.835
DeBERTa-v3-base	0.871	0.672	0.735	0.636	0.934	0.859	0.892	0.833
RoBERTa-base (Focal Loss)	0.827	0.593	0.591	0.597	0.911	0.831	0.821	0.842
MathBERT	0.845	0.596	0.633	0.581	0.919	0.836	0.848	0.825
Multitask (RoBERTa)	0.858	0.553	0.534	0.573	0.919	0.847	0.838	0.857
Multitask (DeBerta)	0.879	0.576	0.572	0.582	0.941	0.881	0.893	0.869
Multitask (Bert)	0.871	0.562	0.579	0.555	0.936	0.861	0.904	0.829

Table 3: TRACK-1: Mistake Identification performance across various transformer models using Strict and Lenient evaluation metrics. Colour codings - Blue (Strict), Green (Lenient)

Model	Strict Evaluation				Lenient Evaluation			
	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)
Without Oversampling								
DeBERTa-v3-base	0.729	**0.580**	**0.592**	**0.577**	0.813	0.753	0.779	0.738
SpanBERT-base-cased	0.684	0.533	0.568	0.527	0.817	0.745	0.803	0.722
Codebert-base	0.688	0.519	0.535	0.512	0.802	0.741	0.765	0.727
Modern-bert-base	0.671	0.564	0.599	0.599	0.813	0.739	0.796	0.717
Roberta-base	0.682	0.542	0.577	0.548	0.813	0.742	0.792	0.721
Bert-base-uncased	0.684	0.506	0.544	0.494	0.802	0.723	0.782	0.701
Multitask (RoBERTa)	0.729	0.476	0.489	0.489	0.809	0.739	0.783	0.719
Multitask (Deberta)	**0.739**	0.489	0.512	0.498	**0.831**	**0.765**	**0.823**	**0.739**
Multitask (Bert)	0.720	0.463	0.505	0.472	0.812	0.726	0.811	0.701
With Oversampling								
SpanBERT-base-cased	0.709	0.553	0.559	0.548	0.811	**0.759**	0.771	0.751
DeBERTa-v3-base	0.694	0.532	0.539	0.528	0.802	0.747	0.761	0.737
Codebert-base	0.659	0.521	0.528	0.525	0.786	0.726	0.739	0.717
Modern-bert-base	0.633	0.536	0.577	0.565	**0.813**	0.745	0.788	0.725
Roberta-base	0.686	**0.56**	**0.566**	**0.578**	0.798	0.744	0.755	**0.737**
Bert-base-uncased	**0.718**	0.533	0.565	0.521	0.807	0.733	**0.783**	0.713

Table 4: TRACK-2: Mistake Location Performance across various transformer models using Strict and Lenient Evaluation. Colour codings - Blue (Strict), Green (Lenient)

Model	Strict Evaluation				Lenient Evaluation			
	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)
Multitask (RoBERTa)	0.659	0.447	0.434	0.485	0.823	0.722	0.751	0.704
Multitask (Bert)	**0.667**	0.454	0.472	0.485	**0.847**	**0.742**	0.812	0.711
Multitask (Deberta)	0.664	0.458	**0.567**	0.486	0.836	0.730	0.784	0.704
BERT (Last layer predictions)	0.589	**0.503**	0.516	**0.497**	0.748	0.607	0.622	0.599
BERT (Second-last Layer + Linear Classifier)	0.581	0.453	0.507	0.448	0.732	0.594	0.602	0.589
RoBERTa (Last layer predictions)	0.655	0.593	0.611	0.582	0.825	0.733	0.754	**0.718**
RoBERTa (Second-last Layer + Linear Classifier)	0.282	0.288	0.731	0.439	0.841	0.688	**0.901**	0.654
DeBERTa (Last layer predictions)	0.601	0.524	0.539	0.520	0.760	0.611	0.637	0.601
DeBERTa (Second-last Layer + Linear Classifier)	0.615	0.418	0.671	0.425	0.813	0.611	0.847	0.598
DistilRoberta (Last layer predictions)	0.601	0.522	0.543	0.517	0.778	0.636	0.672	0.622
DistilRoberta (Second-last Layer + Linear Classifier)	0.479	0.376	0.525	0.427	0.561	0.529	0.568	0.597

Table 5: TRACK-3: Providing Guidance performance across various transformer models using Strict and Lenient evaluation metrics. Colour codings - Blue (Strict), Green (Lenient)

Model	Strict Evaluation				Lenient Evaluation			
	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	Macro F1 (↑)	Precision (↑)	Recall (↑)
Multitask (RoBERTa)	0.669	0.449	0.469	0.472	0.801	0.722	0.784	0.701
Multitask (Bert)	0.669	0.449	0.493	0.469	0.815	0.731	0.826	0.705
Multitask (Deberta)	0.715	0.505	0.484	0.536	0.844	0.807	0.814	0.799
Stacking (BERT + Extra Trees)	0.754	**0.655**	**0.675**	**0.648**	0.867	0.849	0.847	0.851
Stacking (BERT + Logistic Regression)	0.744	0.637	0.654	0.632	0.873	0.855	0.854	0.856
Stacking (RoBERTa + Extra Trees)	0.744	0.632	0.646	0.628	0.875	0.857	0.858	0.855
Stacking (RoBERTa + Logistic Regression)	**0.756**	0.662	0.674	0.657	**0.879**	**0.862**	**0.862**	**0.862**
Stacking (DeBERTa + Extra Trees)	0.734	0.647	0.651	0.645	0.881	0.861	0.869	0.855
Stacking (DeBERTa + Logistic Regression)	0.726	0.629	0.637	0.623	0.873	0.850	0.863	0.841

Table 6: TRACK-4: Actionability performance across various models for Strict and Lenient evaluations. Colour codings - Blue (Strict), Green (Lenient)