Synthetic Student Responses: LLM-Extracted Features for IRT Difficulty Parameter Estimation

Matías Hoyl[1]

[1]School of Education, Stanford University

Abstract

Educational assessment relies heavily on knowing question difficulty, traditionally determined through resource-intensive pre-testing with students. This creates significant barriers for both classroom teachers and assessment developers. We investigate whether Item Response Theory difficulty parameters can be accurately estimated without student testing by modeling the response process and explore the relative contribution of different feature types to prediction accuracy. Our approach combines traditional linguistic features with pedagogical insights extracted using Large Language Models (LLMs), including solution step count, required mathematical skills, cognitive complexity, and potential misconceptions. We implement a two-stage process: first training a neural network to predict how students would respond to questions, then deriving difficulty parameters from these simulated response patterns. Using a dataset of over 250,000 student responses to mathematics questions, our model achieves a Pearson correlation of 0.85 between predicted and actual difficulty parameters on completely unseen questions. These results suggest that our method can reliably estimate question difficulty without pre-testing, potentially accelerating assessment development while maintaining psychometric quality. Furthermore, our analysis demonstrates that LLM-extracted pedagogical features contribute significantly to prediction accuracy, suggesting new pathways for AI-assisted educational assessment.

1 Introduction

Educational assessment plays a vital role in teaching and learning, but creating high-quality questions requires knowing their difficulty level—how challenging they are for students with different abilities. Currently, determining question difficulty requires extensive pre-testing with students followed by statistical analysis, an approach that is effective but creates significant delays and costs in developing new assessments.

For teachers developing classroom assessments, this process is often impractical, leading to tests with unknown psychometric properties. For large-scale assessment programs, the pre-testing requirement consumes substantial resources and limits the speed at which new content can be developed. Both scenarios highlight the need for methods that can predict question difficulty without requiring actual student testing.

This study addresses two specific research questions:

- How effectively can we estimate Item Response Theory difficulty parameters without student testing by modeling the student response process?
- To what extent do different feature types (LLM-extracted pedagogical features, linguistic characteristics, and semantic embeddings) contribute to the accuracy of difficulty predictions?

Rather than treating difficulty prediction as a direct regression problem, which is what most existing research does, we take an approach that mirrors how difficulty emerges in practice—through patterns of student responses.

Our method combines conventional question features with pedagogical insights extracted using Large Language Models (LLMs). These AI tools analyze questions to identify solution steps, required mathematical skills, cognitive complexity, and potential student misconceptions—aspects typically assessed by human experts. By embedding this pedagogical knowledge in our predictions, we create a system that better reflects how questions function in educational settings.

The remainder of this paper is organized as follows. Section 2 reviews related work on difficulty prediction, highlighting existing approaches and gaps in the literature. Section 3 describes our dataset, which contains over 250,000 student responses to mathematics questions. Section 4 outlines our methodology, including feature engineering and our neural network architecture. Section 5 presents our experimental results, demonstrating the effectiveness of our approach. Finally, Section 6 discusses the implications of our findings, acknowledges limitations, and suggests directions for future research.

2 Related Work

2.1 Traditional Item Difficulty Estimation

Traditional approaches to difficulty estimation typically rely on either expert judgment or statistical analysis of student responses. Manual calibration by subject matter experts, while common, has been shown to be inconsistent and subjective [1, 2]. Items developed by experts often perform differently than expected when deployed in real settings.

Statistical calibration through pre-testing with actual students provides more reliable estimates but introduces significant delays before questions can be used operationally. This approach typically applies Item Response Theory (IRT) models to estimate difficulty parameters based on student response patterns [3]. While effective, this process is time-consuming and resource-intensive, creating a bottleneck in the development of high-quality assessments.

The limitations of these traditional methods have motivated research into automated approaches that can predict item difficulty from the question's textual content, potentially reducing or eliminating the need for extensive pre-testing.

2.2 Text-based Features for Difficulty Prediction

Research on text-based difficulty prediction has identified several categories of features that contribute to question complexity.

2.2.1 Linguistic Features

Linguistic characteristics of questions are strongly associated with their difficulty. Some studies have extracted lexical, syntactic, and semantic features to predict question complexity [4, 5]. Common linguistic features include:

- **Lexical features:** Word count, word frequency, word length, and vocabulary difficulty [6, 7]
- **Syntactic features:** Sentence length, syntactic complexity, and grammatical structures [5]
- **Semantic features:** Concept density, abstraction level, and semantic similarity between options [8]

Hickendorff [9] demonstrated that linguistic factors significantly impact student performance on mathematics assessments, even when controlling for mathematical complexity.

2.2.2 Domain-specific Features

Features specific to certain domains have been shown to improve prediction accuracy:

- **Mathematics:** Number of mathematical symbols, presence of graphs or figures, computation complexity [10]
- **Reading comprehension:** Text layout, presence of contextual cues, relationship between passages and questions [11]
- **Multiple-choice questions:** Similarity between options, plausibility of distractors [12, 13]

Toyama [14] emphasized that item difficulty is determined by the interplay of item characteristics, respondent characteristics, and the context in which assessment occurs.

2.3 Machine Learning Approaches for Difficulty Prediction

Early approaches to difficulty prediction relied on deterministic methods or simple regression models. Recent advances have introduced increasingly sophisticated machine learning techniques.

2.3.1 Classical Machine Learning Methods

Traditional machine learning algorithms have been widely applied to the challenge of difficulty prediction. Linear and logistic regression models have proven effective in establishing relationships between text features and difficulty levels, as demonstrated by Ha et al. [6] in their work on high-stakes medical exams. More sophisticated approaches include the R2DE model proposed by Benedetto et al. [10], which leverages random forests to simultaneously predict both IRT difficulty and discrimination parameters based on TF-IDF features extracted from question text. The literature also shows that feature-based ensemble methods have achieved notable success across various educational domains, with Ha et al. [6] demonstrating that combining different feature types can enhance prediction accuracy compared to single-feature approaches.

2.3.2 Neural Network Models

Deep learning approaches have increasingly been applied to difficulty prediction, with several notable innovations emerging in recent years. Recurrent Neural Networks, particularly those utilizing Long Short-Term Memory (LSTM) architectures, have demonstrated effectiveness in capturing sequential patterns within question text, as shown in the work of Huang et al. [15]. Building on this foundation, Qiu, Wu, and Fan [16] introduced a more sophisticated approach with their Document-enhanced Attention Network (DAN), which improves prediction accuracy by enriching questions with relevant domain documents. This contextual enhancement allows the model to better understand the relationship between questions and their subject matter. Further advancing the field, Xue et al. [17] explored transfer learning techniques, demonstrating that models pre-trained on related tasks such as response time prediction can be effectively fine-tuned for difficulty estimation, leveraging knowledge gained from one educational assessment task to improve performance on another.

2.3.3 Transformer-based Approaches

The most recent research has explored transformer-based models for difficulty prediction, leveraging their superior ability to capture contextual relationships. Benedetto et al. [18] compared BERT and DistilBERT models for difficulty estimation, finding that transformer models outperform previous approaches by up to 6.5% in terms of RMSE. Building on this work, Gombert et al. [19] employed scalar-mixed transformer encoders with specialized regression heads, showing significant improvements over baseline models. Further advancing the field, Kapoor et al. [20] incorporated embeddings from various LLMs (ModernBERT, BERT, and LLAMA) alongside linguistic features, achieving a correlation of 0.77 between true and predicted difficulty.

A key advantage of transformer models is their ability to capture complex semantic relationships within text without requiring extensive feature engineering. Aradelli [21] demonstrated that fine-tuning pre-trained transformers on domain-specific corpora further enhances prediction accuracy. These approaches represent the current state-of-the-art in difficulty prediction, combining the contextual understanding capabilities of transformer architectures with specialized training techniques to achieve unprecedented levels of accuracy in estimating question difficulty from text alone.

2.4 Feature Extraction Using Language Models

Recent studies have explored the use of Large Language Models (LLMs) not just for prediction but also for feature extraction:

- **Procedural complexity:** Using LLMs to quantify the number of steps required to solve problems Liu et al. [22]
- **Skill identification:** Extracting the specific skills required to answer questions Didolkar et al. [23]
- **Cognitive level assessment:** Classifying questions according to Bloom's taxonomy Scaria, Dharani Chenna, and Subramani [24]
- **Misconception analysis:** Identifying potential student misconceptions associated with questions Sadihin, Rodriguez, and Chen [25]

These approaches leverage the reasoning capabilities of LLMs to extract pedagogically meaningful features that might be difficult to capture through conventional feature engineering.

2.5 Gaps in Current Research

Despite significant advances in difficulty prediction methods, two important gaps remain in current research.

First, while transformer-based models have shown remarkable results, often outperforming traditional machine learning approaches by leveraging contextual relationships within question text, there has been limited exploration of using Large Language Models (LLMs) to extract pedagogically meaningful features from questions. Most studies rely on conventional linguistic features or embedding-based approaches, neglecting the potential of LLMs to identify more complex attributes, such as procedural complexity, specific mathematical skills, cognitive levels, and potential misconceptions.

These LLM-extracted features could provide richer signals about question characteristics that are difficult to capture through traditional feature engineering approaches. The few studies that have begun exploring LLM-based feature extraction have shown promising results, but this approach remains underutilized in the difficulty prediction literature.

Second, most current research approaches difficulty prediction as a direct regression problem, where models are trained to predict difficulty parameters directly from question text and features. This direct approach, while intuitive, bypasses a crucial intermediate step: modeling how students with varying ability levels would actually respond to these questions.

Few studies have explored the alternative approach of first simulating student responses to questions and then deriving difficulty parameters from these simulated response patterns. This simulation-based approach offers several advantages over direct prediction. It better mimics the actual process through which question difficulty is determined in real educational settings, allowing difficulty parameters to emerge naturally from student performance patterns rather than being directly predicted. Additionally, it provides a more interpretable model that captures both question characteristics and student abilities, while also producing additional useful metrics beyond difficulty, such as discrimination parameters and expected response patterns.

This gap presents an opportunity to develop more robust difficulty estimation methods. If a model can successfully simulate how students would respond to new questions, it could potentially be applied to any unseen question to estimate difficulty parameters without requiring actual student testing. Such an approach would significantly reduce the resource burden of developing high-quality assessments while maintaining psychometric validity.

The present study addresses both of these gaps by (1) leveraging LLMs to extract complex pedagogical features from questions and (2) developing a neural network model that first predicts student responses and then derives IRT difficulty parameters from these predictions, offering a more comprehensive approach to the difficulty estimation problem.

3 Data

3.1 Overview

The dataset used in this study originates from Zapien, an adaptive educational technology platform based in Chile, focused on helping students learn mathematics. The dataset contains 280,979 answers from approximately 2,000 students across 50+ schools. Each record represents an interaction between a student and the platform, providing information on the student's skill level, question attributes, and outcomes.

Key features of the dataset include:

- Student demographic information and skill levels
- Question characteristics (options, text) and difficulty metrics
- Response data including correctness and timing
- Topic and subject categorization
- School and grade-level information

3.2 Data Preprocessing

Data preprocessing involved several key steps to ensure data quality and prepare the dataset for analysis:

- Merging multiple data sources including student response data (`master_translated.csv`), question information (`questions_master.csv`), and IRT difficulty metrics (`question_difficulties_irt.csv`)
- Translating all questions from Spanish to English to optimize performance with embedding models, which typically work best with English text
- Filtering out questions with fewer than 10 total responses to ensure reliable statistics
- Removing questions with extremely low difficulty values (`irt_difficulty < -6`) to exclude outliers

The final processed dataset contained 251,361 answers to 4,681 questions after filtering, with 42 columns including original and derived features. For modeling purposes, we implemented a robust three-way split at the question level rather than row level:

- **Training set**: 3,275 questions (70%) used for model training
- **Validation set**: 937 questions (20%) used for model tuning and performance assessment
- **Holdout test set**: 469 questions (10%) saved for final evaluation

The splits were carefully balanced to ensure similar distributions of question difficulty (IRT difficulty) and correctness rates across all three sets, enabling reliable assessment of model performance.

3.3 Data Exploration

Key variables are analyzed to understand their distributions and identify patterns in student behavior.

The dataset exhibits a consistent imbalance across all splits, with approximately 70% correct answers and 30% incorrect answers (specifically 30.5% for training, 28.7% for validation, 29.9% for test, and 30.1% overall), reflecting the general performance of students on the Zapien platform and confirming our data splits were properly balanced, which ensures that our model training and evaluation are performed on comparable data distributions.

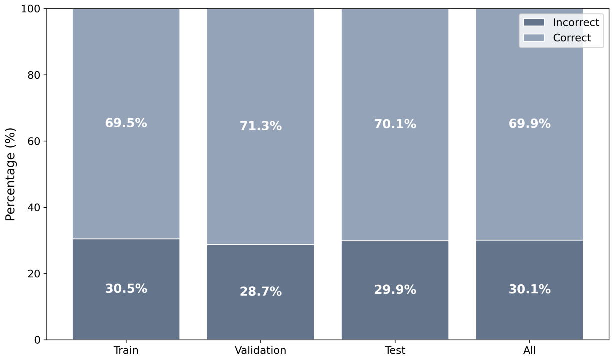

Figure 1: *Proportion of Correct and Incorrect Answers Across Data Splits*

The distribution of user ability levels in the Zapien platform shows a roughly normal distribution centered around -1, indicating that by the platform's internal calculations, the majority of students have skill levels in the negative range.

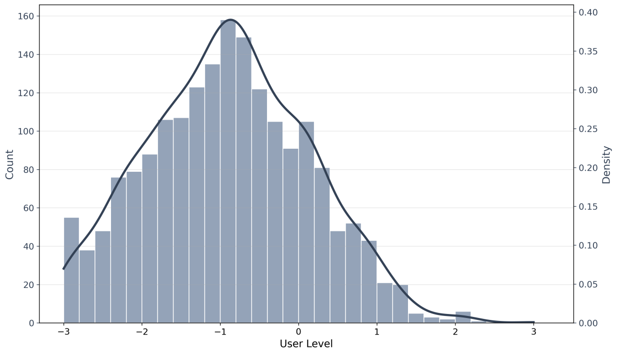

Figure 2: *Distribution of User Ability Levels*

The distribution of IRT difficulty parameters shows remarkable consistency across our three data splits (training, validation, and test), with overlapping density plots centered around -1 and most questions having parameters between -2 and 2. This balanced distribution, including a small spike of very easy questions around -5, confirms our splitting strategy's effectiveness and ensures that performance metrics reflect true model generalization rather than differences in underlying data distributions.

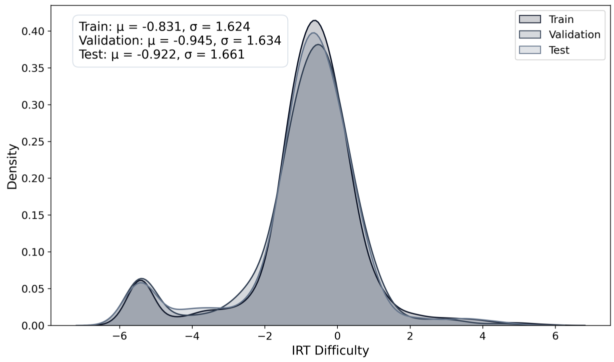

Figure 3: *Distribution of IRT Difficulty Parameters Across Data Splits*

4 Methods

This section outlines our methodological approach to predicting student performance on mathematics questions. We developed a pipeline that combines traditional machine learning techniques with features extracted using large language models (LLMs). Our approach integrates student-specific information, question characteristics, and interaction patterns to create a robust predictive framework.

We first describe our overall model architecture, followed by our feature engineering process which includes linguistic analysis, LLM-based feature extraction, and temporal interaction patterns. We then detail our model selection process and evaluation methodology.

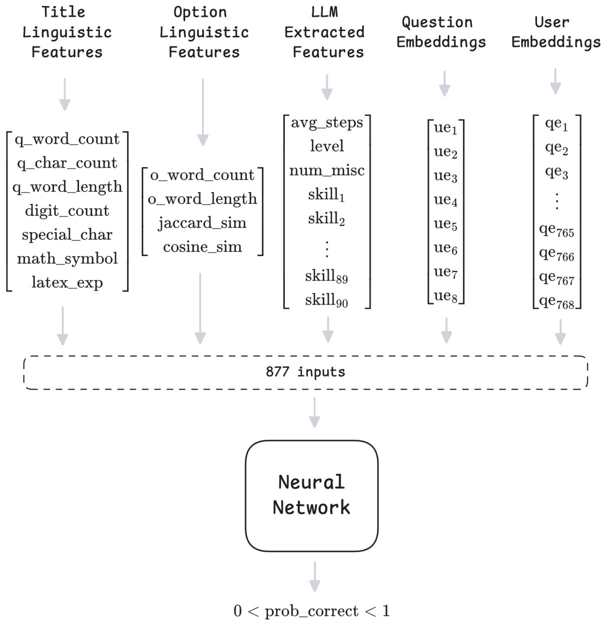

Figure 4: *Overview of the Model Pipeline*

4.1 Feature Engineering

4.1.1 Linguistic Features

To capture the textual complexity and mathematical content of the questions, we extracted a set of linguistic features from the question titles and answer options. These features aim to quantify various aspects of question complexity that might influence student performance.

Question Title Features We extracted the following features from the question titles:

- **q_word_count**: Number of words in the question title. This provides a simple measure of question length, with the average question containing 23 words.
- **q_char_count**: Total number of characters in the question title. The average question contained 118 characters, with the longest having 945 characters.
- **q_avg_word_length**: Average length of words in the title, calculated as the ratio of character count to word count. This metric helps assess the complexity of vocabulary used.
- **q_digit_count**: Number of numerical digits present in the question. This is particularly relevant for mathematics questions, which often contain numerical values.
- **q_special_char_count**: Count of special characters in the title, excluding common punctuation. Special characters often indicate more complex notation.
- **q_mathematical_symbols**: Count of mathematical symbols (e.g., +, -, ×, ÷, =, <, >) in the ques-

tion. Questions typically contained around 8 mathematical symbols on average.

- **q_latex_expressions**: Count of LaTeX expressions or mathematical formatting in the question, indicating the presence of complex mathematical notation.

Answer Option Features In addition to the question title, we analyzed the answer options to extract insights about their variability and length:

- **jaccard_similarity_std**: Standard deviation of Jaccard similarities between all pairs of options. This measures how similar or different the answer options are from each other.
- **avg_option_length**: Average character length of answer options, which averaged about 10 characters.
- **avg_option_word_count**: Average number of words in answer options, with most options containing between 1-3 words.

These linguistic features provided valuable signals for our predictive models. For example, the number of mathematical symbols and LaTeX expressions served as proxies for the mathematical complexity of questions, while features like word count and average word length helped quantify the linguistic complexity. The Jaccard similarity between answer options captured information about the discriminative power of the question, as very similar options might be more challenging for students to differentiate.

4.1.2 Question Embeddings

Semantic understanding of the questions is crucial for predicting student performance. To capture this, we generated embeddings of the complete question text along with its answer options using the ModernBERT model.

We formatted each question and its answer options into a structured format:

```
Question:     [question_title]
Correct Answer:     [correct_option]
Wrong Answer 1:     [wrong_option_1]
Wrong Answer 2:     [wrong_option_2]
Wrong Answer 3:     [wrong_option_3]
Wrong Answer 4:     [wrong_option_4]
```

The formatted text was processed through the ModernBERT model to generate 768-dimensional contextual embeddings. We applied mean pooling to combine token-level embeddings into sentence-level embeddings and normalized them to ensure consistent comparisons.

4.1.3 Options Embeddings

We also generated separate embeddings for each individual answer option using ModernBERT, creating 768-dimensional vectors for each of the five possible options (A through E). These embeddings captured the semantic meaning of each answer choice in isolation.

From these option embeddings, we calculated cosine similarity metrics between the correct answer and each incorrect option. For each question, we computed four similarity values (ranging from -1 to 1), creating features called `wrong_1_similarity` through `wrong_4_similarity`.

These similarity metrics provided our model with important information about the relationships between answer choices. While the question embeddings captured the overall problem context, these similarity features helped our models understand how answer options related to each other, offering insight into how specific answer choices might influence student responses.

4.1.4 Feature Extraction Using LLMs

In addition to traditional feature engineering and embedding techniques, we leveraged Large Language Models (LLMs) to extract pedagogically relevant features from questions that would be difficult to capture through conventional methods. We used Google's Gemini 2.0 Flash with Chain-of-Thought (CoT) prompting, requiring the model to articulate its reasoning process within `<thinking></thinking>` tags before providing the final output (see appendix). This transparency allowed us to evaluate the quality of feature extraction and ensure pedagogical validity.

Solution Step Count To quantify procedural complexity, we prompted the LLM to solve each mathematics question in the most detailed, pedagogically atomic manner possible. The LLM was instructed to break down the solution into discrete steps that would be appropriate for student instruction. For each question, we ran this extraction three separate times to account for variability. The average number of steps across these runs became a numerical feature (`avg_steps`). This feature served as a proxy for question complexity, based on the hypothesis that questions requiring more solution steps might be more difficult.

Mathematical Skills To identify the specific mathematical knowledge required to solve each question, we created a comprehensive taxonomy of 90 common mathematical skills (see appendix). The LLM was provided with the question, answer options, and a step-by-step solution. For each question, the model identified which skills from the taxonomy were utilized in the solution. We performed this extraction five times and retained only skills that appeared in at least three runs to ensure stability. These skills were converted to one-hot encoded features, creating binary indicators for each skill.

Cognitive Level To assess the level of thinking required by each question according to Bloom's Taxonomy, we provided the LLM with a detailed rubric describing the six levels of Bloom's Taxonomy. For each question, the model determined which cognitive level (1-6, see appendix) was primarily being assessed. This process was repeated three times with majority voting to determine the final cognitive level. The resulting ordinal feature (`cognitive_level`) captured the complexity of thinking required by each question, providing an important dimension of difficulty beyond purely mathematical complexity.

Misconception Count To understand the potential for student errors, we prompted the LLM to analyze possible misconceptions. The LLM was instructed to generate an exhaustive, atomic list of misconceptions students might have when approaching each question. This extraction was performed three times per question, and the average number of identified misconceptions became a numerical feature (`num_misconceptions`). This feature provided insight into questions with multiple potential pitfalls, helping us quantify another dimension of question complexity.

These LLM-extracted features complemented our traditional and embedding-based features by directly capturing pedagogical aspects of the questions. By using multiple runs and aggregation techniques (averaging or majority voting), we ensured the stability and reliability of these features.

4.1.5 User Embeddings

The last piece of the puzzle in our modeling approach was incorporating user information, since we had student answer data for each question. Accounting for individual student abilities was essential for accurate predictions, but presented significant technical challenges. We initially attempted to incorporate user IDs as one-hot encoded features in a LightGBM model, but this approach proved problematic. With 1,867 binary features for users compared to only 146 question-related features, the model essentially functioned as a user classifier rather than a question difficulty estimator. This structural imbalance resulted in predictions with minimal variation across questions for the same user (standard deviation only 0.026), compromising our ability to assess question characteristics.

To address this limitation, we implemented a neural network with a dedicated user embedding layer that mapped each user ID to an 8-dimensional vector space. This approach reduced the feature dimensionality while creating dense, continuous representations of user abilities. The embedding layer contained a learnable matrix of dimensions ($n\_users \times embedding\_dim$) that captured latent ability factors conventional one-hot encoding could not efficiently represent. This embedding-based approach successfully balanced user and question influences, enabling our model to make meaningfully different predictions for different questions presented to the same user, while ensuring user characteristics did not overshadow the question features we were primarily interested in analyzing.

4.2 Modeling Approach

After exploring several modeling approaches, we developed a neural network architecture specifically designed to balance the influence of user characteristics and question features. This addressed the primary limitation of our initial models, where user features dominated predictions.

Our model combines user embeddings (dimensions: $n\_users \times 8$), question numerical features, and question embeddings into a single concatenated feature vector. This combined representation is then passed through two dense layers with 64 and 32 units respectively, applying dropout rates of 0.3 and 0.2 to prevent overfitting. The network concludes with a sigmoid output layer that predicts the probability of a correct answer, effectively balancing the contributions of both user abilities and question characteristics.

Component	Implementation
L2 regularization	0.001 (all weight matrices)
Optimization	Adam optimizer
Initial Learning Rate	0.001
Loss Function	Binary cross-entropy
Early Stopping	10 epochs

Table 1: *Neural network implementation details.*

4.3 Prediction Task

While our neural network achieved strong classification performance, our primary research objective extended beyond merely predicting student responses. The ultimate goal was to use these predictions to estimate question difficulty parameters in a way that aligns with Item Response Theory (IRT), a foundational framework in educational assessment.

After training the neural network on 70% of the questions and validating on 20%, we used the model to generate predictions for the 10% completely unseen questions in the holdout test set. For each of the 469 holdout questions, we generated predictions for all 1,867 users. We first preserved the raw probability outputs from the model, representing each student's likelihood of answering correctly. We then applied a 0.5 threshold to these probabilities to convert them to binary predictions (0 for incorrect, 1 for correct). This conversion produced a correctness matrix where each cell contained either 0 or 1, representing whether a specific user would answer a specific question correctly.

We then used these correctness matrices as inputs to an Item Response Theory estimation process. We implemented a 2-Parameter Logistic (2PL) IRT model using PyTorch to estimate three key parameters: question difficulty (how challenging each question is), question discrimination (how well each question differentiates between ability levels), and user ability (the proficiency level of each user). These parameters are related through the IRT equation $P(\text{correct}) = \sigma(a(\theta - b))$, where σ is the logistic function. The parameter estimation employed a maximum likelihood approach, where parameters were initialized as random values with small variance and optimized using the Adam optimizer with a learning rate of 0.01 and weight decay of $1e - 5$. We used a learning rate scheduler that reduced the rate by half when progress plateaued, and the optimization process ran for up to 5,000 iterations with early stopping.

After estimating the parameters, we compared them to the ground truth difficulty parameters from the original dataset.

5 Results

5.1 Model Performance

Our neural network model achieved strong performance on the prediction task, effectively balancing the influence of user characteristics and question features. Classification metrics on the validation set demonstrated the model's ability to accurately predict student responses to unseen questions.

Metric	Value
Accuracy	78.15%
Precision	82.18%
Recall	88.12%
F1 Score	85.04%
AUC-ROC	0.8197

Table 2: *Classification performance metrics on the validation set.*

The classification metrics indicate that our model performs well at predicting individual student responses, with an accuracy of 78.15% on the validation set. The higher precision (82.18%) compared to accuracy suggests that when the model predicts a student will answer correctly, it is usually right. The recall of 88.12% indicates the model successfully identifies most instances where students would answer correctly. The F1 score of 85.04% represents a strong balance between precision and recall. The AUC-ROC value of 0.8197 further confirms the model's ability to distinguish between correct and incorrect responses across different threshold settings.

These results are particularly encouraging as they confirm that our model is effectively leveraging both question features and student characteristics. The strong classification performance indicates the model is capturing sufficient signal from student features to modify its predictions based on individual student abilities, while simultaneously accounting for inherent question characteristics. This balanced approach provides a solid foundation for the subsequent IRT parameter estimation, as accurate response predictions that account for both question difficulty and student proficiency are essential for deriving reliable difficulty parameters.

5.2 IRT Parameter Estimation

Beyond direct prediction performance, our primary research objective was to accurately estimate Item Response Theory (IRT) difficulty parameters for unseen questions. Our neural network approach demonstrated exceptional performance on this task.

When applying our model to the completely unseen questions in the holdout test set using predictions from all 1,867 users, we achieved alignment between estimated and true difficulty parameters:

Metric	Value
Pearson correlation	0.8549
Spearman rank correlation	0.9629
Mean Absolute Error (MAE)	0.6286
Root Mean Square Error (RMSE)	1.0760

Table 3: *IRT parameter estimation metrics on the holdout test set.*

The high correlation values (Pearson > 0.85, Spearman > 0.96) demonstrate that the difficulty estimations derived from our predicted student responses strongly align with the ground truth parameters.

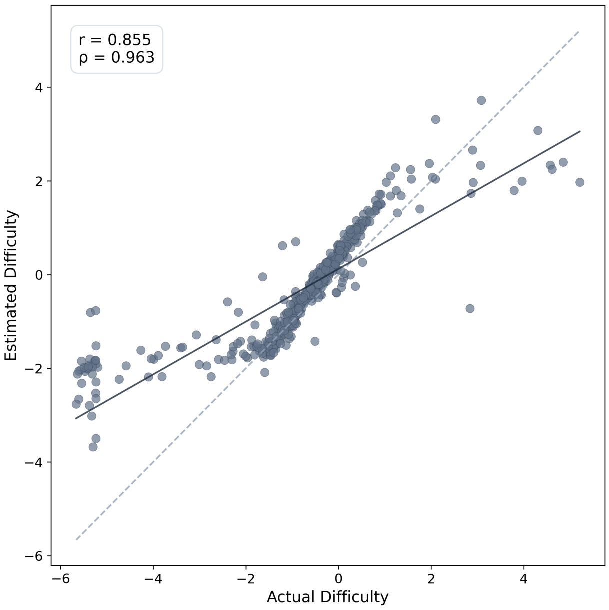

Figure 5: *Correlation between original and estimated IRT difficulty parameters showing strong linear relationship across the entire difficulty spectrum.*

As shown in Figures 5 and 6, our model captures both the relative ordering and the absolute scaling of question difficulties. The strong linear relationship in Figure 5 confirms the high correlation metrics, while Figure 6 demonstrates that the distributions of estimated and true difficulty parameters align remarkably well, preserving the overall shape and range found in the original dataset.

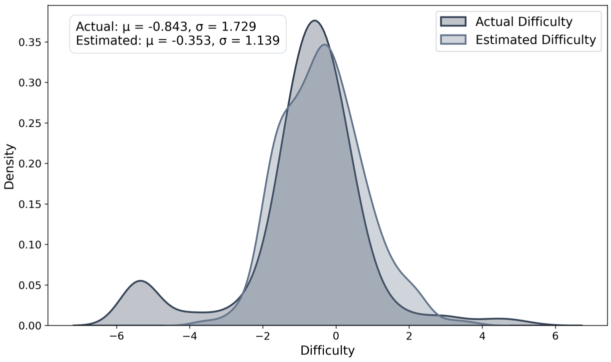

Figure 6: *Distribution comparison of original and estimated difficulty parameters demonstrating close alignment in both shape and range.*

This evaluation demonstrates that our neural network approach achieved its primary objective: accurately estimating question difficulty parameters for completely unseen questions, by predicting student responses to those questions.

6 Discussion

6.1 Conclusions

Our research demonstrates that Item Response Theory difficulty parameters can be accurately estimated without traditional student testing by combining linguistic features with pedagogical insights extracted using Large Language Models. This approach addresses a significant challenge in educational assessment: the need for efficient methods to determine question difficulty that don't rely on resource-intensive pre-testing with students.

The strong performance of our neural network model in predicting student responses (78.15% accuracy, 85.04% F1 score) confirms that our feature engineering approach effectively captures the factors that influence student performance on mathematics questions. By integrating traditional linguistic features, semantic embeddings, and LLM-extracted pedagogical insights, we created a representation of question characteristics that allowed our model to make accurate predictions.

Most importantly, when presented with completely unseen questions, our model generated response patterns so reliable that we could predict question difficulties with remarkable accuracy (Pearson correlation of 0.85). These high correlations demonstrate that our approach can effectively capture both the relative ordering and absolute scaling of question difficulties without requiring actual student testing.

6.2 Limitations

Despite the promising results, several limitations of our study should be acknowledged:

Generalizability : Our dataset originates from Chile, which may limit the generalizability of our findings to student populations elsewhere. Educational systems, curricula, and pedagogical approaches vary across countries and cultures, potentially affecting how question features relate to difficulty in different contexts.

LLM Stability : Large Language Models are inherently probabilistic. Even though we attempted to ensure stability by making multiple LLM calls and aggregating results, each new call could potentially yield different outputs. This variability introduces some uncertainty into the LLM-extracted features, which could affect the consistency of our predictions.

Model Interpretability : The neural network we employed is essentially a black box that performs well on the prediction task but offers limited insight into the relative importance of different features. This lack of interpretability makes it difficult to draw definitive conclusions about which specific question characteristics contribute most significantly to difficulty.

Model Sensitivity : Additionally, our results may vary depending on which LLM vendor we use (e.g., OpenAI, Anthropic, Google) and which inference parameters we select (e.g., temperature, top-p sampling). These choices can significantly impact the quality and consistency of the LLM-extracted features that feed into our model.

Skill Taxonomy : The mathematical skill taxonomy we used was created specifically for this study rather than drawing on a standardized, widely accepted framework. This limits comparability with other research and raises questions about the comprehensiveness and validity of our skill categorization.

6.3 Future Work

In future iterations of this research, we plan to address these limitations through several targeted improvements:

Dataset Expansion : We are currently in discussions to obtain the ASSISTments dataset to validate our approach on a different student population. This additional dataset would allow us to test the generalizability of our method across different educational contexts and student demographics.

Baseline Comparisons : We will define clear baselines to compare our approach against, including simpler models that rely solely on conventional features. Additionally, we plan to conduct ablation studies by systematically removing feature groups to assess their relative contribution to prediction accuracy and difficulty alignment. This will help determine which features are most critical for accurate difficulty estimation.

Methodological Variations : We will test variations of the model architecture, prompting strategies, and embedding models to evaluate the sensitivity of our results to these methodological choices. This exploration will help identify the most robust and effective configuration for difficulty prediction.

Pedagogical Feature Refinement : We plan to refine the pedagogical aspects of our LLM features, particularly the skill taxonomy and Bloom's taxonomy rubric used with the LLM. This refinement will involve aligning our framework with established educational standards and taxonomies to improve comparability with existing research.

Human Validation : We will compare LLM annotations with human expert annotations for a subset of questions to assess the agreement between automated and human judgments. This validation will help establish the credibility of LLM-extracted features and identify areas where LLM performance could be improved.

By addressing these directions in future research, we aim to strengthen the validity, reliability, and generalizability of our approach to difficulty estimation, potentially establishing a new standard for efficient assessment development in educational contexts.

References

[1] Lawrence Rudner. "Implementing the Graduate Management Admission Test Computerized Adaptive Test". In: Jan. 2010, pp. 151–165. ISBN: 978-0-387-85459-5. DOI: 10.1007/978-0-387-85461-8_8.

[2] James C. Impara and Barbara S. Plake. "Teachers' Ability to Estimate Item Difficulty: A Test of the Assumptions in the Angoff Standard Setting Method". In: *Journal of Educational Measurement* 35.1 (1998), pp. 69–81. DOI: https://doi.org/10.1111/j.1745-3984.1998.tb00528.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1745-3984.1998.tb00528.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1745-3984.1998.tb00528.x.

[3] Georg Rasch. "Probabilistic models for some intelligence and attainment tests". In: *Danish Institute for Educational Research* (1960).

[4] S. AlKhuzaey, F. Grasso, T. R. Payne, et al. "Text-based Question Difficulty Prediction: A Systematic Review of Automatic Approaches". In: *International Journal of Artificial Intelligence in Education* 34 (2024), pp. 862–914. DOI: 10.1007/s40593-023-00362-1.

[5] Inn-Chull Choi and Youngsun Moon. "Predicting the Difficulty of EFL Tests Based on Corpus Linguistic Features and Expert Judgment". In: *Language Assessment Quarterly* 17.1 (2020), pp. 18–42. DOI: 10.1080/15434303.2019.1674315. eprint: https://doi.org/10.1080/15434303.2019.1674315. URL: https://doi.org/10.1080/15434303.2019.1674315.

[6] Le An Ha et al. "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*. Ed. by Helen Yannakoudakis et al. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 11–20. DOI: 10.18653/v1/W19-4402. URL: https://aclanthology.org/W19-4402/.

[7] Luca Benedetto et al. "Introducing a framework to assess newly created questions with Natural Language Processing". In: 2020. arXiv: 2004.13530 [cs.CL]. URL: https://arxiv.org/abs/2004.13530.

[8] Hatice Cigdem Bulut, Okan Bulut, and Serkan Arikan. "Evaluating group differences in online reading comprehension: The impact of item properties". In: *International Journal of Testing* 23.1 (2023), pp. 10–33. DOI: 10.1080/15305058.2022.2044821. eprint: https://doi.org/10.1080/15305058.2022.2044821. URL: https://doi.org/10.1080/15305058.2022.2044821.

[9] Marian Hickendorff. "The Language Factor in Elementary Mathematics Assessments: Computational Skills and Applied Problem Solving in a Multidimensional IRT Framework". In: *Applied Measurement in Education* 26.4 (2013), pp. 253–278. DOI: 10.1080/08957347.2013.824451. eprint: https://doi.org/10.1080/08957347.2013.824451. URL: https://doi.org/10.1080/08957347.2013.824451.

[10] Luca Benedetto et al. *R2DE: a NLP approach to estimating IRT parameters of newly generated questions*. 2020. arXiv: 2001.07569 [cs.LG]. URL: https://arxiv.org/abs/2001.07569.

[11] Joshua F. Lawrence et al. "Reading Comprehension and Academic Vocabulary: Exploring Relations of Item Features and Reading Proficiency". In: *Reading Research Quarterly* 57.2 (2022), pp. 669–690. DOI: https://doi.org/10.1002/rrq.434. eprint: https://ila.onlinelibrary.wiley.com/doi/pdf/10.1002/rrq.434. URL: https://ila.onlinelibrary.wiley.com/doi/abs/10.1002/rrq.434.

[12] Ghader Kurdi, Bijan Parsia, and Uli Sattler. "An Experimental Evaluation of Automatically Generated Multiple Choice Questions from Ontologies". In: Feb. 2017, pp. 24–39. ISBN: 978-3-319-54626-1. DOI: 10.1007/978-3-319-54627-8_3.

[13] Le An Ha and Victoria Yaneva. "Automatic Distractor Suggestion for Multiple-Choice Tests Using Concept Embeddings and Information Retrieval". In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Ed. by Joel Tetreault et al. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 389–398. DOI: 10.18653/v1/W18-0548. URL: https://aclanthology.org/W18-0548/.

[14] Yukie Toyama. "What Makes Reading Difficult? An Investigation of the Contributions of Passage, Task, and Reader Characteristics on Comprehension Performance". In: *Reading Research Quarterly* 56.4 (2021), pp. 633–642. DOI: https://doi.org/10.1002/rrq.440. eprint: https://ila.onlinelibrary.wiley.com/doi/pdf/10.1002/rrq.440. URL: https://ila.onlinelibrary.wiley.com/doi/abs/10.1002/rrq.440.

[15] Zhenya Huang et al. "Question difficulty prediction for READING problems in standard tests". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 1352–1359. DOI: 10.5555/3298239.3298437.

[16] Zhaopeng Qiu, Xian Wu, and Wei Fan. "Question Difficulty Prediction for Multiple Choice Problems in Medical Exams". In: Nov. 2019, pp. 139–148. ISBN: 978-1-4503-6976-3. DOI: 10.1145/3357384.3358013.

[17] Kang Xue et al. "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*. Ed. by Jill Burstein et al. Seattle, WA, USA → Online: Association for Computational Linguistics, July 2020, pp. 193–197. DOI: 10.18653/v1/2020.bea-1.20. URL: https://aclanthology.org/2020.bea-1.20/.

[18] Luca Benedetto et al. "On the application of Transformers for estimating the difficulty of Multiple-Choice Questions from text". In: *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*. Ed. by Jill Burstein et al. Online: Association for Computational Linguistics, Apr. 2021, pp. 147–157. URL: https://aclanthology.org/2021.bea-1.16/.

[19] Sebastian Gombert et al. "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). Ed. by Ekaterina Kochmar et al. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 483–492. URL: https://aclanthology.org/2024.bea-1.40/.

[20] Radhika Kapoor et al. *Prediction of Item Difficulty for Reading Comprehension Items by Creation of Annotated Item Repository*. 2025. arXiv: 2502.20663 [cs.CL]. URL: https://arxiv.org/abs/2502.20663.

[21] Giovanni Aradelli. "Transformers for Question Difficulty Estimation from Text". MA thesis. Politecnico di Milano, Dec. 2020. URL: https://hdl.handle.net/10589/170785.

[22] Yixin Liu et al. "Improving Large Language Model Fine-tuning for Solving Math Problems". In: *ArXiv* abs/2310.10047 (2023). URL: https://api.semanticscholar.org/CorpusID:264146593.

[23] Aniket Didolkar et al. "Metacognitive Capabilities of LLMs: An Exploration in Mathematical Problem Solving". In: *ArXiv* abs/2405.12205 (2024). URL: https://api.semanticscholar.org/CorpusID:269921384.

[24] Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. "Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation". In: *Artificial Intelligence in Education*. Springer Nature Switzerland, 2024, pp. 165–179. ISBN: 9783031642999. DOI: 10.1007/978-3-031-64299-9_12. URL: http://dx.doi.org/10.1007/978-3-031-64299-9_12.

[25] Bryan Constantine Sadihin, Hector Rodriguez Rodriguez, and Matteo Jiahao Chen. "[Proposal-ML] Mining Misconception in Mathematics". In: *Submitted to Tsinghua University Course: Advanced Machine Learning*. under review. 2024. URL: https://openreview.net/forum?id=CE85qdNSlp.

Appendix

A Step by Step Example

The following is an example prompt used to generate step by step solution explanations for a mathematical problem:

Prompt

You are an expert in math pedagogy. Your task is to answer the following question deconstructing it to the most elemental steps. Don't skip any step. Don't assume anything about the reader, try to solve it in the most atomic and pedagogical way possible.

Here is the question:

Question: What is the decreasing order of the roots $4\sqrt{3}$, $3\sqrt{4}$, $2\sqrt{5}$, and $5\sqrt{2}$?
A) $2\sqrt{5} > 4\sqrt{3} > 5\sqrt{2} > 3\sqrt{4}$
B) $2\sqrt{5} > 3\sqrt{4} > 4\sqrt{3} > 5\sqrt{2}$
C) $5\sqrt{2} > 3\sqrt{4} > 4\sqrt{3} > 2\sqrt{5}$
D) $5\sqrt{2} > 4\sqrt{3} > 3\sqrt{4} > 2\sqrt{5}$
E) $2\sqrt{5} > 4\sqrt{3} > 5\sqrt{2} > 3\sqrt{4}$
Correct Answer: D

First, think about your task step by step inside <thinking></thinking> tags. Your thinking process should try to go over all the posible ways to solve the question and try to find the most atomic way to solve it.

Then, when you are ready to answer, after your </thinking> tag you should write your response using "Step" as the key and the step as the value. For example:

<thinking>
Thinking process... Think as long as you need to.
</thinking>

Step 1: ...
Step 2: ...
Step 3: ...
...

Important instructions:
- Don't skip any step.
- Don't assume anything about the reader, try to solve it in the most atomic and pedagogical way possible.
- Use as many steps as you need to.

And here is the LLM's response:

Thinking Process

<thinking>
Okay, let's break down this problem. We need to compare four numbers: $4\sqrt{3}$, $3\sqrt{4}$, $2\sqrt{5}$, and $5\sqrt{2}$ and arrange them in decreasing order. The main challenge is that the numbers are in a mixed form (integer multiplied by a square root). To compare them easily, we need to get rid of the square roots or the integers outside the square root.

The easiest way to compare these values is to bring the coefficients *inside* the square root.

Let's consider how to bring a number inside a square root. For example, consider $a\sqrt{b}$. We can rewrite this as $\sqrt{a^2 * b}$.

So, the first step is to convert all the numbers into the form of a single square root.

Then, we simply compare the numbers *inside* the square roots, because if $a > b$, then $\sqrt{a} > \sqrt{b}$.

Finally, we arrange the original numbers based on their corresponding squared values, in decreasing order.

Let's start the solution.
</thinking>

Response

Step 1: Convert $4\sqrt{3}$ into the form $\sqrt{x}$.
We have $4\sqrt{3} = \sqrt{4^2 * 3} = \sqrt{16 * 3} = \sqrt{48}$

Step 2: Convert $3\sqrt{4}$ into the form $\sqrt{x}$.
We have $3\sqrt{4} = \sqrt{3^2 * 4} = \sqrt{9 * 4} = \sqrt{36}$

Step 3: Convert $2\sqrt{5}$ into the form $\sqrt{x}$.
We have $2\sqrt{5} = \sqrt{2^2 * 5} = \sqrt{4 * 5} = \sqrt{20}$

Step 4: Convert $5\sqrt{2}$ into the form $\sqrt{x}$.
We have $5\sqrt{2} = \sqrt{5^2 * 2} = \sqrt{25 * 2} = \sqrt{50}$

Step 5: Now we have $\sqrt{48}$, $\sqrt{36}$, $\sqrt{20}$, and $\sqrt{50}$. We can easily compare the values inside the square roots: 48, 36, 20, and 50.

Step 6: Arrange the values inside the square roots in decreasing order:
50 > 48 > 36 > 20

Step 7: Replace the values inside the square

roots with their original forms:
$5\sqrt{2} > 4\sqrt{3} > 3\sqrt{4} > 2\sqrt{5}$

Step 8: Compare this result with the given options.
The correct option is D: $5\sqrt{2} > 4\sqrt{3} > 3\sqrt{4} > 2\sqrt{5}$.

B Mathematical Skill List

The following table presents the list of mathematical skills assessed in our dataset for our skill one-hot encoding.

Skill	Category
Addition	Arithmetic/Number Sense
Subtraction	Arithmetic/Number Sense
Multiplication	Arithmetic/Number Sense
Division	Arithmetic/Number Sense
Mental Math	Arithmetic/Number Sense
Estimation	Arithmetic/Number Sense
Number Properties	Arithmetic/Number Sense
Fractions	Arithmetic/Number Sense
Decimals	Arithmetic/Number Sense
Percentages	Arithmetic/Number Sense
Factors and Multiples	Arithmetic/Number Sense
Order of Operations	Arithmetic/Number Sense
Ratios	Ratios Proportions Rates
Proportions	Ratios Proportions Rates
Rates	Ratios Proportions Rates
Unit Conversions	Ratios Proportions Rates
Scale Factors	Ratios Proportions Rates
Variables and Expressions	Algebra
Linear Equations	Algebra
Systems of Equations	Algebra
Inequalities	Algebra
Polynomials	Algebra
Factoring	Algebra
Quadratic Equations	Algebra
Exponents and Radicals	Algebra
Rational Expressions	Algebra
Logarithms	Algebra
Function Notation	Functions and Graphing
Domain and Range	Functions and Graphing
Linear Functions	Functions and Graphing
Quadratic Functions	Functions and Graphing
Polynomial Functions	Functions and Graphing
Exponential Functions	Functions and Graphing
Logarithmic Functions	Functions and Graphing
Piecewise Functions	Functions and Graphing
Function Transformations	Functions and Graphing
Function Composition	Functions and Graphing
Angles	Geometry
Triangles	Geometry
Quadrilaterals	Geometry
Polygons	Geometry
Circles	Geometry
Area and Perimeter	Geometry
Volume and Surface Area	Geometry
Coordinate Geometry	Geometry
Transformations	Geometry
Pythagorean Theorem	Geometry
Similar Triangles	Geometry
Congruence	Geometry
Geometric Constructions	Geometry
Geometric Proofs	Geometry
Basic Trigonometric Ratios	Trigonometry
Inverse Trigonometric Functions	Trigonometry
Unit Circle	Trigonometry
Trigonometric Identities	Trigonometry
Law of Sines/Cosines	Trigonometry
Polar Coordinates	Trigonometry
Mean Median Mode	Statistics and Probability
Range and Standard Deviation	Statistics and Probability
Data Representation	Statistics and Probability
Basic Probability	Statistics and Probability
Compound Probability	Statistics and Probability
Conditional Probability	Statistics and Probability
Combinatorics	Statistics and Probability
Statistical Inference	Statistics and Probability
Regression Analysis	Statistics and Probability
Hypothesis Testing	Statistics and Probability
Limits	Calculus
Derivatives	Calculus
Integrals	Calculus
Applications of Calculus	Calculus
Word Problem Translation	Problem Solving
Multi-step Problem Solving	Problem Solving
Mathematical Modeling	Problem Solving
Logical Reasoning	Problem Solving
Pattern Recognition	Problem Solving
Complex Numbers	Advanced Topics
Matrices	Advanced Topics
Sequences and Series	Advanced Topics
Vector Geometry	Advanced Topics
Set Theory	Discrete Mathematics
Graph Theory	Discrete Mathematics
Financial Mathematics	Applied Mathematics
Technology in Mathematics	Tools and Technology
Mathematical Communication	Meta Skills
Mathematical Literacy	Meta Skills

C Bloom's Cognitive Level Table

Table 4 outlines Bloom's Taxonomy levels used to classify the cognitive demands of problems. Each level is an ordinal number in our feature encoding.

Cognitive Level	Description
Remember	Recall facts and basic concepts
Understand	Explain ideas or concepts, interpret, summarize
Apply	Use information in new situations, execute procedures
Analyze	Draw connections among ideas, break down into parts
Evaluate	Justify a stand or decision, verify, critique
Create	Produce new or original work, design, construct

Table 4: *Bloom's Taxonomy of cognitive levels applied to mathematical problem classification.*

D Example of Misconceptions List

Following the example from the appendix A, here is one of the outputs of the LLM when asked about common misconceptions for the question in the example.

1. Students may confuse 'decreasing order' with 'increasing order', leading them to reverse the correct order of the numbers.
2. Students might incorrectly apply the property $a\sqrt{b} = \sqrt{a^2 * b}$ by forgetting to square 'a' before multiplying by 'b', calculating it as $a\sqrt{b} = \sqrt{a * b}$.
3. Students might make arithmetic errors when squaring the number outside the square root (e.g., calculating 4^2 as 8 instead of 16).
4. Students might make arithmetic errors in the multiplication step within the square root (e.g., calculating $16 * 3$ as 45 instead of 48).
5. Students may believe that the magnitude of the number outside the square root is the primary determinant of the overall value, even if the value inside the square root is smaller (e.g., incorrectly assuming that $5\sqrt{2}$ is always greater than $2\sqrt{5}$ because 5 > 2).
6. After correctly comparing the values under the square roots, students might forget to convert them back to their original forms when selecting the final answer, leading them to choose an option with the numbers under the square roots ordered.
7. Students might make careless copying errors when transferring the correctly ordered expressions from their intermediate steps to the final answer choice.
8. Students may lack a strong number sense regarding the approximate values of square roots, making it difficult to estimate and compare the values without performing the conversion to a common form.
9. Students may struggle with the concept that irrational numbers can be ordered on a number line, treating them as inherently harder to compare than integers.
10. Students might try to apply the square root individually to each term, incorrectly stating that $a\sqrt{b} = \sqrt{a} * \sqrt{b}$.
11. Students might not understand what a root is and how it affects the order of the numbers.

This was performed 3 times and then averaged the number of misconceptions for each question.