Reasoning-Enhanced Retrieval for Misconception Prediction: A RAG-Inspired Approach with LLMs

Divya Chaudhary[*], **Chang Xue** [†] , **Shaorui Sun**[†]
Northeastern University
Seattle, Washington, USA
d.chaudhary@northeastern.edu

Abstract

Educational Data Mining (EDM) is a growing field that leverages data-driven methods to improve learning and teaching processes. Among its applications, diagnostic questions have emerged as a valuable tool for identifying common student misconceptions. These questions feature a correct answer and distractors, each aligned with specific misunderstandings. In this study, we propose a two-stage retrieval framework inspired by Retrieval-Augmented Generation (RAG) techniques to predict and rank misconceptions associated with incorrect answers in mathematical multiple-choice questions. Our approach leverages semantic retrieval to identify candidate misconceptions and employs large language models (LLMs) to reason about and refine the ranking of these misconceptions. By combining retrieval with LLM-based reasoning, our method improves both the accuracy and the interpretability of the prediction of misconceptions, offering a scalable solution for educational data mining. The experimental results demonstrate the effectiveness of our approach, outperforming traditional retrieval methods in predicting student misconceptions. Beyond its educational context, our method advances AI-enabled scientific workflows by framing misconception detection as a multi-stage process where LLMs assist in generating hypotheses, evaluating candidate explanations, and interpreting human-produced knowledge representations.

1 Introduction

Diagnosis of student cognitive misconceptions is a fundamental challenge in mathematics education. Misconceptions often stem from systematic misunderstandings of mathematical concepts, which pose significant barriers to effective learning. Identifying these misconceptions accurately and efficiently is crucial to providing personalized feedback and improving educational outcomes. However, traditional diagnostic methods, which are based on predefined error patterns or rigid criteria, struggle to adapt to various problem solving scenarios(Baker and Inventado, 2014; Khosravi et al., 2022).

Recent advances in natural language processing (NLP) and information retrieval (IR) have introduced powerful tools for tackling complex educational tasks. Transformer-based models such as BERT(Reimers and Gurevych, 2019) and GPT(Brown et al., 2020) have significantly advanced semantic understanding and retrieval, enabling insights into large, diverse datasets(Lewis et al., 2021; Devlin et al., 2019). Despite these breakthroughs, applying such models to diagnose misconceptions in mathematics presents unique challenges. Diagnosis of errors involves not only understanding mathematical content, but also reasoning about the cognitive processes that lead to incorrect answers, an area where current models often fail(Liu et al., 2023; Nye et al., 2021).

This study aims to design a framework for effectively identifying and ranking misconceptions related to incorrect answers in educational assessments in emerging space of LLM-assisted scientific workflows. Achieving this requires the development of a robust ranking mechanism that leverages the semantic and conceptual affinity between misconceptions and incorrect answers, while simultaneously addressing several critical challenges:

- **Complex reasoning demands**: Current large language models (LLMs) excel at solving mathematical problems but often lack the ability to engage in diagnostic reasoning. Identifying misconceptions requires counterfactual reasoning, understanding the flawed thought processes that lead to incorrect answers, which remains an underexplored limitation in existing models(Liu et al., 2023; Nye et al., 2021).

[*]Corresponding Author
[†]Both Chang Xue and Shaorui Sun contributed equally to this research.

Proceedings of The First Workshop on Human–LLM Collaboration for Ethical and Responsible Science Production (SciProdLLM), pages 38–51
December 23, 2025 ©2025 Association for Computational Linguistics

- **Subtle distinctions in misconceptions**: Misconceptions in mathematics often exhibit nuanced differences, requiring high precision to distinguish between closely related conceptual or computational errors. These distinctions are crucial for a meaningful diagnosis and personalized feedback(King et al., 2024).
- **Generalization to novel misconceptions**: Beyond identifying known misconceptions, models must demonstrate the flexibility to generalize their predictions to previously unseen cases, a critical capability for scaling to diverse educational settings(King et al., 2024).

Although our primary application lies in mathematics education, our framework directly contributes to responsible human-LLM scientific workflows. Misconception detection is structurally similar to the scientific quality-control tasks: identifying flawed reasoning, detecting inconsistencies, interpreting human-generated text, and evaluating conceptual validity. Our two-stage retrieval + LLM reasoning pipeline functions as an automated scientific workflow that (1) preprocesses data, (2) retrieves hypotheses (candidate misconceptions), (3) conducts automated experimentation via reranking, and (4) performs LLM-based inference to evaluate and refine the retrieved knowledge. Thus, our system is an instance of an LLM-assisted scientific pipeline aimed at improving the accuracy, reliability, and interpretability of human knowledge representations.

To address these challenges, we propose a novel two-stage framework inspired by Retrieval-Augmented Generation (RAG)(Levonian et al., 2023). Our approach combines semantic retrieval to identify candidate misconceptions with large language model (LLM)-based reasoning to refine and rank these misconceptions. By integrating retrieval with reasoning, our framework improves both diagnostic accuracy and interpretability, offering a scalable solution for educational data mining. This study makes the following key contributions:

- **Framework Innovation**: We introduce a two-stage pipeline that integrates semantic retrieval and LLM reasoning to diagnose and rank misconceptions in mathematical multiple-choice questions.
- **Enhanced Reasoning and Discrimination**: Our method addresses the limitations of counterfactual reasoning and provides fine-grained differentiation among closely related misconceptions, tackling critical challenges in this domain.
- **Empirical Validation**: Extensive experiments on a real-world dataset demonstrate significant improvements in prediction accuracy and generalization compared to baseline methods(King et al., 2024).

2 Related Work

2.1 Advances in Deep Learning for NLP and IR

Deep learning has significantly advanced natural language processing (NLP) and information retrieval (IR). Transformer-based models, notably BERT and Sentence-BERT, have enhanced semantic search and contextual understanding(Reimers and Gurevych, 2019). Pre-trained models like GPT have excelled in tasks such as text generation and knowledge-intensive retrieval(Devlin et al., 2019; Brown et al., 2020). These models have been widely adopted for text ranking tasks, improving the precision and relevance of search results(Lin et al., 2021; Guo et al., 2019). Sentence-BERT, for instance, provides high-quality sentence embeddings for semantic similarity tasks.

Recent studies have further explored these developments. Min et al. (2021) surveyed the use of large pre-trained language models in NLP tasks, discussing approaches like pre-training, fine-tuning, prompting, and text generation(Min et al., 2021). Torfi et al. (2020) provided a comprehensive overview of deep learning advancements in NLP, highlighting the impact of models like BERT and GPT on various applications(Torfi et al., 2021). Chernyavskiy et al. (2021) examined the limitations of transformer-based models, emphasizing the need for models to handle certain information types effectively(Chernyavskiy et al., 2021). Omar et al. (2022) discussed the robustness of NLP techniques, addressing challenges such as adversarial attacks and the importance of developing models capable of handling real-world complexities(Omar et al., 2022). Hagos and Rawat (2024) explored the current state of generative AI and large language models, discussing their applications and emerging challenges(Hagos et al., 2024).

These studies underscore the transformative impact of deep learning on NLP and IR, providing essential insights and tools that pave the way for

future innovations in addressing complex reasoning.

Despite these advancements, challenges remain, including the need for counterfactual reasoning to understand flawed cognitive processes behind incorrect answers and addressing nuanced distinctions between similar misconceptions, which require higher semantic precision.

2.2 AI Diagnosis of Math Misconceptions

In mathematics education, traditional methods for diagnosing misconceptions often rely on structured scoring criteria or predefined error categories(Baker and Inventado, 2014). While effective in controlled settings, these methods struggle to adapt to the diverse responses seen in problem-solving scenarios(Khosravi et al., 2022).

Recent research has highlighted the potential of Large Language Models (LLMs) in addressing these limitations(Liu et al., 2023). Similarly, studies like (Nye et al., 2021) highlight the importance of intermediate reasoning steps in explaining student behavior.

Natural language processing (NLP) methods have been utilized to detect patterns in students' textual responses, uncovering common misconceptions that may not be evident through traditional analysis(Michalenko et al., 2017). These advancements facilitate the development of personalized educational tools that can adapt to individual learning needs. Additionally, comprehensive surveys of EDM and LA highlight the integration of various data mining techniques to enhance personalized education, emphasizing the importance of cognitive diagnosis and knowledge tracing in understanding student learning behaviors(Xiong et al., 2024).

Some efforts have been made to use AI to assist in mathematics education, including leveraging large language models (LLMs) to generate high-quality distractors for multiple choice mathematical questions(Fernandez et al., 2024) and utilizing LLMs to solve mathematical problems(Era et al., 2025).

The evolution of EDM underscores the critical role of technology in transforming educational practices to meet the diverse needs of learners(Lin et al., 2024).

Recent work in the LLM in Science Production community highlights LLMs as meta-scientific tools that support idea generation, hypothesis exploration, error detection, multimodal content generation, and workflow automation. In this framing, LLMs do not merely solve tasks, but analyze, critique, and evaluate human-generated content. Our work contributes directly to this line of research by treating student free-text explanations and incorrect answers as scientific artifacts that require structured evaluation. The proposed retrieval + LLM reasoning pipeline mirrors scientific fact-checking workflows, where a system must retrieve plausible hypotheses (candidate misconceptions), evaluate them, and assign evidence-based relevance scores. We position misconception detection as a scientific knowledge-validation problem, aligned with research on LLM-supported scientific production, quality control, and responsible AI-generated analysis.

3 Preliminaries

In this section, we first introduce the research problem. Then, we describe the characteristics and challenges associated with mathematical misconceptions. Finally, we discuss the technical foundations and evaluation metrics that provide the basis for our proposed methodology.

3.1 Formal Problem Definition

Let $\mathcal{Q}$ denote a mathematical multiple-choice question (MCQ) with a stem $\mathcal{S}$ and answer options $O = \{o_1, \ldots, o_n\}$, where one option is correct and others are distractors. Each distractor $o_i \in O_{\text{incorrect}}$ is associated with a set of predefined misconceptions $\mathcal{M} = \{m_1, \ldots, m_k\}$.

The goal is to design a system that retrieves and ranks the most relevant misconceptions $\mathcal{M}^* \subseteq \mathcal{M}$ for each o_i, such that:

$$\mathcal{M}^* = \arg \max_{\mathcal{M}' \subseteq \mathcal{M}} P(\mathcal{M}' \mid \mathcal{Q}, O_{\text{incorrect}}), \quad (1)$$

where P measures the likelihood of misconceptions explaining the incorrect answers in Table 1. Consider the query derived from the student task question - (0.9 ÷ 0.3 = ?). The retriever converts this query into a dense embedding:

$$v_q = \text{BGE}([Subject = Decimals; \; Construct = Divide\,two\,decimals; \; Question; \; StudentAnswer]) \quad (2)$$

A misconception such as Students assume the quotient must have the same number of decimal places as the operands is encoded as:

$$v_m = \text{BGE}([MisconceptionText]) \quad (3)$$

The retrieval stage computes cosine similarity between v_q, and all v_m, returning the most semantically relevant misconception hypotheses.

3.2 Key Challenges and Problem Characteristics

Mapping mathematical misconceptions from distractor options is a multi-faceted problem, distinguished by the following theoretical and practical challenges:

1. *Semantic Misalignment Between MCQs and Misconceptions*: Mathematical misconceptions in $\mathcal{M}$ are often described in semi-structured formats (e.g., natural language, equations, or diagrams), while MCQs are composed of diverse textual and mathematical components. This mismatch complicates direct similarity computation and demands robust representation learning.

2. *Contextual Relationships of Distractors*: Unlike traditional IR tasks, where documents are evaluated independently, misconceptions in $\mathcal{M}$ exhibit structured relationships. While each incorrect option o_i is primarily associated with a specific misconception, semantically similar misconceptions ($m_i \simeq m_j$) may lead to overlapping error patterns in $\mathcal{O}_{\text{incorrect}}$. Effectively capturing these relationships requires a framework that can distinguish nuanced variations between related misconceptions while maintaining their conceptual boundaries.

3. *Balancing Precision, Recall, and Efficiency*: High recall is essential to ensure relevant misconceptions are included in $\mathcal{M}_{\text{candidate}}$, while precision is critical for $\mathcal{M}_{\text{ref}}$. Furthermore, $\mathcal{M}^*$ must exhibit efficiency to avoid unnecessary computational overhead in generating explanations for distractors. Achieving this balance necessitates novel re-ranking and optimization techniques.

4. *Theoretical Underpinning of Misconception Spaces*: Misconceptions $\mathcal{M}$ can be viewed as residing in a latent conceptual space where distances correspond to semantic and contextual similarities. Understanding this space's geometry, such as clusters or subspaces representing specific misconception categories, is pivotal for retrieval and reasoning.

4 Methodology

4.1 Framework Overview

To solve the problem defined in Section 3.1, we propose a two-stage retrieval framework that combines dense semantic search with LLM-based reasoning. The pipeline operates in five phases (Figure 1):

1. **Initial Semantic Retrieval**: Encode $\mathcal{Q}$ and retrieve top-100 misconceptions $\mathcal{M}_{\text{candidate}}$ via cosine similarity.

2. **First-Stage Re-ranking**: Refine $\mathcal{M}_{\text{candidate}}$ to top-50 using contextual relevance scores.

3. **LLM Reasoning**: Analyze $O_{\text{incorrect}}$ to infer potential misconceptions $\mathcal{M}_{\text{LLM}}$ through structured prompting.

4. **Final Ranking**: Fuse $\mathcal{M}_{\text{LLM}}$ with the pre-retrieved top-50 candidates, then re-rank them to produce $\mathcal{M}^*$ (top-25).

This design addresses the challenges in Section 3.2: initial retrieval ensures high recall, while LLM reasoning injects diagnostic insights to resolve ambiguous cases (e.g., distinguishing $|x|$ vs. $\sqrt{x^2}$ misconceptions).

4.2 Semantic Retrieval Stage

Model Architecture. We adopt the `BAAI/bge-large-en-v1.5` model, fine-tuned on the Eedi misconception dataset. The model converts questions and misconceptions into 1024-dimensional vectors via the following encoding process:

$$v_q = BGE(Subject; Construct; QuestionText; Answers) \quad (4)$$

Similarity Computation Cosine similarity identifies top candidates:

$$\text{sim}(v_q, v_m) = \frac{v_q \cdot v_m}{\|v_q\|\|v_m\|} \quad (5)$$

We retain the top-100 misconceptions ($\mathcal{M}_{\text{candidate}}$) to balance recall and computational cost. This threshold was validated through grid search on recall@K (see Appendix F.)

Fine-tuning Protocol. The BGE model was optimized with MultipleNegativeRankingLoss, where each training batch contains one positive misconception and 15 hard negatives extracted from incorrect answers. Hyperparameters include:

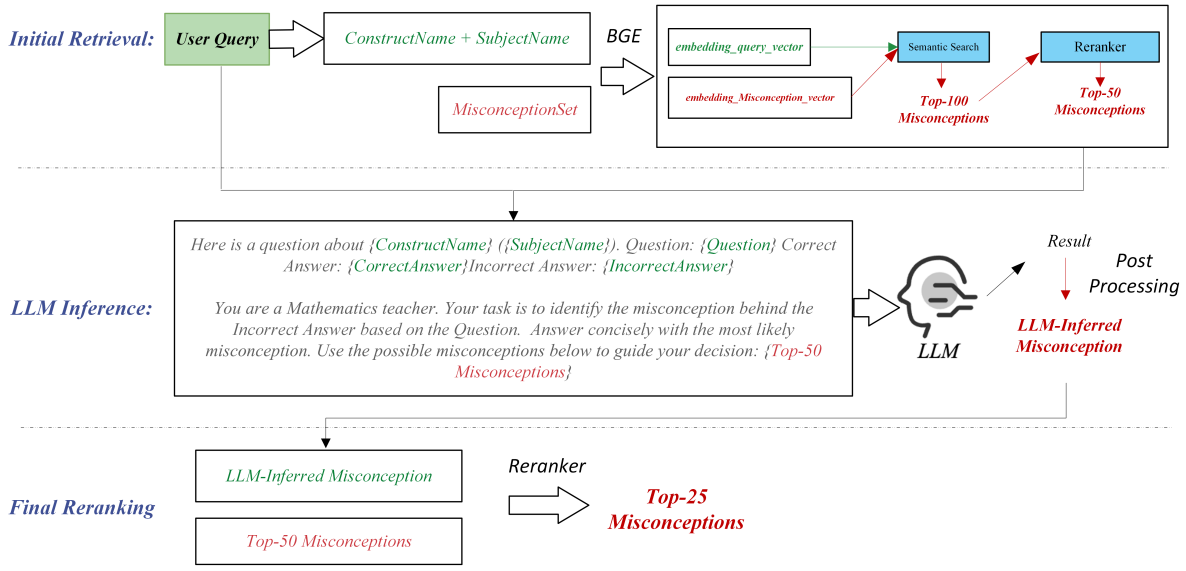

Figure 1: Framework for Two-Stage Retrieval and LLM-Based Inference

- Learning rate: 2×10^{-5} (AdamW optimizer)
- Batch size: 8 (gradient accumulation over 16 steps)
- Training epochs: 2 (early stopping on validation MRR@10)

4.3 Re-ranking Stage

Model Architecture. We utilize a fine-tuned `BAAI/bge-reranker- large` model, which has been adapted on the Eedi misconception dataset. The BAAI/bge-reranker-large model uses a cross-encoder approach, the objective is to assign a higher score to relevant misconceptions than irrelevant ones. We define the relevance score as:

$$S(q, m) = \mathbf{W}_r \cdot h_{[\text{CLS}]}^{(q,m)} \tag{6}$$

$[CLS]$ is a special token used in Transformer-based models, to represent the entire input sequence.
$h_{[CLS]}^{(q,m)}$ is the contextual embedding output from the $[CLS]$ token after encoding both the question q and the misconception m.
W_r is a learned weight matrix that transforms the $[CLS]$ token representation into a scalar relevance score.

After the Semantic Retrieval Stage, which returns the top 100 misconceptions ($\mathcal{M}_{\text{candidate}}$), the Reranker Stage further refines the candidates by selecting the top 50 misconceptions ($\mathcal{M}_{\text{ref}}$) based on relevance scores.

Fine-tuning Protocol. The model was optimized with MarginRankingLoss, where each training batch contains one positive misconception and 15 hard negatives mined from incorrect answers. Hyperparameters include:

- Learning rate: 2×10^{-5} (AdamW optimizer)
- Batch size: 8 (gradient accumulation over 16 steps)
- Training epochs: 3 (early stopping based on validation performance)

4.4 LLM Reasoning Stage

Model Selection. We employ the `Qwen-2.5-32B-Instruct` model.

Prompt Engineering. The LLM receives structured prompts to constrain outputs:

Post-Processing. Algorithm 1 filters LLM outputs:

1. Match the generated text with the predefined $\mathcal{M}$ via the Levenshtein distance (≤ 2).
2. Remove nonmathematical terms (e.g., "calculation error").
3. Deduplicate synonyms (e.g., "confuses area/perimeter" vs. "perimeter/area confusion").

Beyond its educational application, our method contributes to the broader SciProdLLM agenda by modeling misconception detection as a scientific workflow, where LLMs assist in hypothesis generation, automated evaluation, and interpretation of human-produced knowledge artifacts.

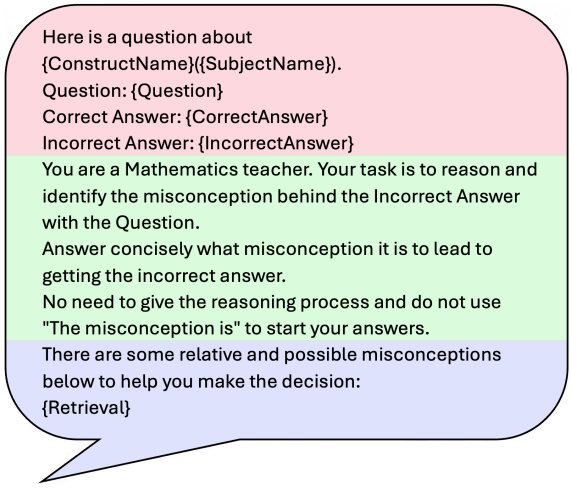

Figure 2: Prompt for LLM Reasoning

Algorithm 1 LLM Output Post-Processing

Require: Raw LLM output t, predefined misconceptions $\mathcal{M}$

```
Extract keywords from t using POS tagging
for each m_i ∈ M do
    if Levenshtein(m_i, t) < 2 OR keyword match > 80% then
        return m_i
    end if
end for
return ∅ (discard if no match)
```

5 Experiment

5.1 Experimental Setup

5.1.1 Dataset

Our experiments are conducted on the *Eedi - Mining Misconceptions in Mathematics* dataset (King et al., 2024), a comprehensive collection of multiple-choice questions designed to evaluate students' mathematical understanding. Each question includes potential misconceptions linked to incorrect answers. The dataset has several key features and characteristics described in the following.

Data Description. The dataset encompasses 1,857 unique questions spanning various elementary mathematics concepts, accompanied by 2,587 misconceptions. Each question in the dataset is meticulously structured with essential information: a unique identifier (QuestionId), the question text (QuestionText) describing the mathematical problem, and associated knowledge components (SubjectName and ConstructName) that specify the mathematical concepts being tested. Each question includes four answer choices (labeled A through D) with one marked as correct. Table 1 illustrates a representative example from the dataset, showcasing how mathematical concepts, questions, answers, and their associated misconceptions are structured.

Data Split. We employ a 5-fold cross-validation scheme to ensure evaluation stability. The data is evenly divided into five parts, with four parts used for training and one part for validation in each iteration. We calculate metrics such as MAP@25 five times and use the mean values for final performance evaluation.

For the Kaggle competition submission, the final model is trained using the entire training set and evaluated on the hidden test set with MAP@25 as the official ranking metric.

Data Augmentation. To enhance the model's generalization capability, we leverage ChatGPT to generate additional training samples. The augmentation process consists of two primary stages: data generation and quality assurance.

For quality assurance, we implement a rigorous validation protocol. Three expert annotators independently evaluate the generated samples, filtering out duplicates and logically inconsistent entries. Through this careful verification process, we retain 9,200 high-quality augmented samples that maintain consistency with the original dataset structure.

The validated augmented data are then integrated with the original training set and utilized in our 5-fold cross-validation experiments. This combined dataset enables a more comprehensive evaluation of our approach while maintaining data quality standards. For detailed prompt engineering processes and annotation guidelines, please refer to Appendix B.

5.1.2 Baselines

To comprehensively evaluate our proposed method, we compare it with several baseline approaches, which can be categorized into traditional retrieval methods and deep learning-based retrieval models.

Traditional retrieval methods. Including **BM25**, a sparse retrieval model that ranks documents according to term frequency, inverse document frequency (IDF) and normalization of document length. BM25 applies a smoothing mechanism to mitigate the influence of overly high or low term frequencies. Similarly, **TF-IDF** is a weighting scheme that measures the importance of a term within a document relative to a collection of doc-

Feature	Content
Subject	Multiplying and Dividing with Decimals
Construct	Divide two decimals with the same number of decimal places
QuestionText	$0.9 \div 0.3 =$
Answer	0.3
Misconception	When dividing decimals with the same number of decimal places as each other, assumes the answer also has the same number of decimal places

Table 1: An example in Eedi.

uments. Both methods rely on lexical matches, which makes them effective for exact keyword matching. However, they struggle with semantic understanding, especially in complex contexts such as mathematical reasoning, where concepts and relations may not be explicitly expressed through surface-level terms.

Deep learning-based retrieval models. Including **Sentence-BERT**, a semantic retrieval model based on the BERT architecture. Sentence-BERT uses a dual-encoder architecture to encode sentences into embeddings, enabling efficient similarity comparisons with metrics like cosine similarity. This approach improves inference speed for tasks like sentence similarity and retrieval. By capturing contextual and semantic information, Sentence-BERT produces high-quality embeddings, making it suitable for semantic similarity tasks, including educational data mining. However, fine-tuning may be needed for optimal performance in specialized domains like mathematical reasoning.

All baseline methods are trained using 5-fold cross-validation under consistent evaluation metrics to ensure reliable comparison. This approach allows us to evaluate the performance of our proposed method against a variety of established techniques, highlighting its strengths and areas for improvement.

5.1.3 Implementation Details

Our method is implemented with careful consideration of the computing environment, training configuration, and optimization strategies to ensure efficient model performance and scalability. The computing environment is specified in detail in the Appendix.

5.2 Main Results

In this subsection, we assess the performance of our proposed method using a variety of evaluation metrics, including MAP@25 (both on Kaggle and locally), Recall@25, and Precision@5. We compare our method with several baselines, encompassing traditional retrieval techniques such as BM25 and TF-IDF, deep learning models like Sentence-BERT and BGE-Retriever, and a combined approach Retriever + Reranker.

5.2.1 Overall Performance Comparison

Table 2 presents the comparative results of different methods averaged over multiple runs. Overall, our proposed method consistently outperforms all baseline methods across all evaluation metrics. Specifically, it achieves superior performance in both ranking quality and retrieval comprehensiveness, as indicated by the MAP@25 and Recall@25 metrics, respectively.

Method	MAP@25 (Kaggle)	MAP@25 (Local)	Recall@25
BM25	0.152	0.175	0.678
TF-IDF	0.128	0.138	0.692
Sentence-BERT	0.203	0.224	0.750
BGE-Retriever	0.232	0.271	0.896
Retriever + Reranker	0.301	0.304	0.911
Our Method	**0.496**	**0.523**	**0.939**

Table 2: Overall Performance Comparison

Mean Average Precision at 25 (MAP@25). The MAP@25 metric provides insight into how well each method ranks relevant documents higher than irrelevant ones. As shown in Table 2, our proposed method achieves a MAP@25 of 0.496 on Kaggle and 0.523 locally, significantly outperforming all other methods. This indicates that our method is highly effective in capturing complex semantic relationships and ranking relevant documents accurately. Traditional methods like BM25 and TF-IDF show considerably lower performance, with MAP@25 values of 0.152 and 0.128 on Kaggle, respectively. Deep learning models such as Sentence-BERT and BGE-Retriever also demonstrate improved performance but still fall short compared to our method.

Recall at 25 (Recall@25). Recall@25 measures the proportion of relevant documents retrieved

within the top 25 results. Our method achieves the highest Recall@25 of 0.939, indicating its exceptional ability in comprehensively retrieving relevant documents. This suggests that our method can effectively cover a large set of relevant documents while maintaining high precision. In contrast, traditional methods like BM25 and TF-IDF achieve Recall@25 values of 0.678 and 0.692, respectively, which are relatively lower. Deep learning models like Sentence-BERT and BGE-Retriever also show significant improvements but do not match the performance of our method.

Comparative Insights and Discussion. From the detailed analysis of the evaluation metrics, it is evident that our proposed method consistently outperforms all baseline methods across all metrics. The significant improvements over traditional and deep learning-based methods highlight the advantages of our method's design and optimization strategies. Specifically:

- Our method achieves the highest MAP@25, indicating superior ranking quality of relevant documents.
- The highest Recall@25 value achieved by our method suggests comprehensive retrieval of relevant documents.

These results demonstrate the robustness and effectiveness of our proposed method in information retrieval tasks. Future work could explore further enhancements and applications of our method in diverse retrieval scenarios.

In conclusion, this comprehensive comparison provides valuable insights into the strengths and limitations of different retrieval methods, and highlights the significant advancements achieved by our proposed method.

5.3 A Case Study of Misconception Mining

To demonstrate the effectiveness of our framework , we conduct a case study using a challenging mathematical problem involving the equation of a parabola. This problem is designed to elicit multiple nuanced misconceptions related to algebraic reasoning and geometric interpretation.

This question is particularly complex because:

1. It requires students to understand the standard form of a parabola equation ($y = a(x-h)^2 + k$) and how to compute the coefficient a.
2. It tests their ability to correctly interpret the relationship between the vertex, given points, and the quadratic coefficient.
3. It elicits multiple closely related misconceptions that are subtle but critical for accurate diagnosis.

Feature	**Content**
Construct	Parabola Equation and Vertex Form
Subject	Quadratic Functions and Equations
Question	What is the equation of the parabola with its vertex at (2, -3) and passing through the point (4, 5)?
Wrong Answer	y = (x-2)^2 $-$ 3

Table 3: A Case Study

Traditional retrieval methods fail to accurately identify misconceptions, often retrieving irrelevant results. For example, the result retrieved by TF-IDF is *"Students ignored the importance of squaring in their calculations."* This issue arises because TF-IDF relies solely on surface-level word matching and cannot capture the underlying mathematical logic of the problem. The result retrieved by BM25 is *"Students confused the direction of a parabola's opening."* While it appears related to parabolas, it does not accurately reflect the core misconception behind the distractor.

In contrast, our framework retrieves the result: *"Students failed to correctly understand the role of a in the quadratic equation and mistakenly assumed that a is always equal to 1."* This improvement is due to our use of the BGE model, which encodes both the problem and misconceptions as dense vectors, capturing deeper semantic relationships. In the initial retrieval stage, relevant misconception candidates related to option are identified. The re-ranking stage further optimizes ranking by prioritizing misconceptions that are contextually relevant. Finally, the LLM reasoning module generates explanatory reasoning, clearly identifying the specific source of the student's misconception.

6 Discussion

This study introduces a novel framework for leveraging initial retrieval results to guide large language models (LLMs) in generating clues, followed by refined retrieval to enhance overall performance. While the framework demonstrates promising po-

tential, several limitations and areas for improvement remain to be addressed.

First, the evaluation currently relies solely on MAP@25, which provides a partial view of the framework's performance. Future work could incorporate additional metrics, such as NDCG or Precision@k, to offer a more comprehensive assessment of retrieval effectiveness across diverse scenarios.

Second, the framework treats all semantic components equally during encoding, which may not align with the hierarchical nature of certain tasks, such as solving mathematical problems. For example, in such contexts, the `question` might be more critical than the `subjectName` or `constructName`. To address this, future efforts could introduce a hierarchical semantic representation model, decomposing problems into dimensions such as `subject`, `construct`, and `text`. Leveraging attention mechanisms to capture interactions across these dimensions and dynamically adjusting their importance via learnable weights may further improve performance.

The limited dataset size remains a key challenge for achieving robust performance. Potential solutions include data augmentation such as generating synthetic samples using large language models or GANs and leveraging transfer learning from related tasks to reduce data scarcity. These limitations point to future research directions, including evaluating the framework on larger, more diverse datasets, incorporating domain specific knowledge, and developing interactive retrieval systems to improve user experience.

7 Conclusion

In this study, we propose a Retriever+ Reranker+LLM Reasoning framework to advance misconception retrieval in mathematics education. Our method integrates semantic retrieval, large language model based reasoning, and targeted data augmentation to enhance both accuracy and interpretability. By incorporating ChatGPT generated augmentation and Hard Negative mining, the framework achieves substantial performance gains, outperforming traditional retrieval baselines on the Eedi Kaggle benchmark in MAP@25. Experimental results demonstrate that Hard Negative mining strengthens model discrimination by introducing challenging negative examples that help the retriever and reranker differentiate subtle misconception patterns. Data augmentation further broadens the training distribution, enabling improved generalization across mathematical constructs and question formats. Finally, LLM driven reasoning provides more structured and explainable misconception ranking, aligning retrieved misconceptions more closely with authentic student thinking.

Although our primary domain is education, the pipeline represents a generalizable scientific workflow: retrieve hypotheses, evaluate them through automated ranking, and refine them using LLM-based reasoning. Such workflows are increasingly used in scientific production for validating experimental results, detecting flawed reasoning in manuscripts, and improving the rigor of LLM-assisted scientific analysis. Our findings demonstrate that structured retrieval-and-reasoning systems can serve as responsible LLM collaborators that improve the quality, interpretability, and trustworthiness of human-generated scientific content.

This work opens several directions for improvement. Enhancing retriever performance through stronger dense retrievers, hybrid retrieval pipelines, or adaptive mechanisms that respond to question complexity could further strengthen reasoning accuracy. Refining LLM prompting strategies may improve interpretability and alignment with pedagogical expectations. Integrating structured knowledge graphs offers another promising pathway, capturing hierarchical relationships among mathematical concepts and misconceptions to support richer reasoning. Finally, testing the framework beyond mathematics, including in physics or chemistry, would help assess its broader applicability across educational domains. Overall, our results highlight the potential of combining retrieval with LLM based reasoning to improve diagnostic and interpretive capabilities in educational AI. Addressing the outlined challenges will help refine the framework further and support scalable, reliable deployment across diverse learning contexts.

Ethical Considerations: Since our work evaluates and interprets human-generated reasoning, it intersects with responsible science production. We highlight risks such as LLM hallucination, incorrect conceptual inference, and bias amplification. Our structured prompts, post-processing constraints, and multi-stage retrieval strategy serve as safeguards, aligning with the workshop's emphasis on ethical and responsible LLM-enabled scientific workflows.

References

Ryan Shaun Baker and Paul Salvador Inventado. 2014. *Educational Data Mining and Learning Analytics*, pages 61–75. Springer New York, New York, NY.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2021. Transformers: "the end of history" for nlp? *Preprint*, arXiv:2105.00813.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Chenxi Dong, Yimin Yuan, Kan Chen, Shupei Cheng, and Chujie Wen. 2025. How to build an ai tutor that can adapt to any course using knowledge graph-enhanced retrieval-augmented generation (kg-rag). *Preprint*, arXiv:2311.17696.

Jalisha Jashim Era, Bidyarthi Paul, Tahmid Sattar Aothoi, Mirazur Rahman Zim, and Faisal Muhammad Shah. 2025. Empowering bengali education with ai: Solving bengali math word problems through transformer models. *Preprint*, arXiv:2501.02599.

Nigel Fernandez, Alexander Scarlatos, Wanyong Feng, Simon Woodhead, and Andrew Lan. 2024. Divert: Distractor generation with variational errors represented as text for math multiple-choice questions. *Preprint*, arXiv:2406.19356.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2019. A deep look into neural ranking models for information retrieval. *Preprint*, arXiv:1903.06902.

Desta Haileselassie Hagos, Rick Battle, and Danda B. Rawat. 2024. Recent advances in generative ai and large language models: Current status, challenges, and perspectives. *Preprint*, arXiv:2407.14962.

Hassan Khosravi, Simon Buckingham Shum, Guanliang Chen, Cristina Conati, Yi-Shan Tsai, Judy Kay, Simon Knight, Roberto Martinez-Maldonado, Shazia Sadiq, and Dragan Gašević. 2022. Explainable artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 3:100074.

Jules King, L Burleigh, Simon Woodhead, Panagiota Kon, Perpetual Baffour, Scott Crossley, Walter Reade, and Maggie Demkin. 2024. Eedi - mining misconceptions in mathematics. https://kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics. Kaggle.

Youngjin Lee. 2024. Developing a computer-based tutor utilizing generative artificial intelligence (gai) and retrieval-augmented generation (rag). *Education and Information Technologies*.

Zachary Levonian, Chenglu Li, Wangda Zhu, Anoushka Gade, Owen Henkel, Millie-Ellen Postle, and Wanli Xing. 2023. Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference. *Preprint*, arXiv:2310.03184.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Preprint*, arXiv:2010.06467.

Yuanguo Lin, Hong Chen, Wei Xia, Fan Lin, Zongyue Wang, and Yong Liu. 2024. A comprehensive survey on deep learning techniques in educational data mining. *Preprint*, arXiv:2309.04761.

Chang Liu, Loc Hoang, Andrew Stolman, and Bo Wu. 2024. Hita: A rag-based educational platform that centers educators in the instructional loop. In *Artificial Intelligence in Education*, pages 405–412, Cham. Springer Nature Switzerland.

Naiming Liu, Shashank Sonkar, Zichao Wang, Simon Woodhead, and Richard G. Baraniuk. 2023. Novice learner and expert tutor: Evaluating math reasoning abilities of large language models with misconceptions. *Preprint*, arXiv:2310.02439.

Joshua J. Michalenko, Andrew S. Lan, and Richard G. Baraniuk. 2017. Data-mining textual responses to uncover misconception patterns. *Preprint*, arXiv:1703.08544.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *Preprint*, arXiv:2111.01243.

Horia Modran, Ioana Corina Bogdan, Doru Ursuțiu, Cornel Samoila, and Paul Livius Modran. 2024. Llm intelligent agent tutoring in higher education courses using a rag approach. *Preprints*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *Preprint*, arXiv:2112.00114.

Marwan Omar, Soohyeon Choi, DaeHun Nyang, and David Mohaisen. 2022. Robust natural language processing: Recent advances, challenges, and future directions. *Preprint*, arXiv:2201.00768.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Maung Thway, Jose Recatala-Gomez, Fun Siong Lim, Kedar Hippalgaonkar, and Leonard W. T. Ng. 2024. Battling botpoop using genai for higher education: A study of a retrieval augmented generation chatbots impact on learning. *Preprint*, arXiv:2406.07796.

Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A. Fox. 2021. Natural language processing advancements by deep learning: A survey. *Preprint*, arXiv:2003.01200.

Zhang Xiong, Haoxuan Li, Zhuang Liu, Zhuofan Chen, Hao Zhou, Wenge Rong, and Yuanxin Ouyang. 2024. A review of data mining in personalized education: Current trends and future prospects. *Frontiers of Digital Education*, 1(1):26–50.

Tianshi Zheng, Weihan Li, Jiaxin Bai, Weiqi Wang, and Yangqiu Song. 2024. Assessing the robustness of retrieval-augmented generation systems in k-12 educational question answering with knowledge discrepancies. *Preprint*, arXiv:2412.08985.

A Related Work

A.1 RAG in Education

Retrieval-Augmented Generation (RAG) is an emerging framework that combines retrieval and generation models to enhance the performance of large language models (LLMs) in tasks requiring deep semantic understanding and external knowledge integration. First introduced by Lewis et al. (2020), RAG bridges the gap between generative capabilities and knowledge-intensive problem-solving by integrating relevant external information into LLM-driven workflows(Lewis et al., 2021).

In educational contexts, RAG has proven effective for developing intelligent tutoring systems that personalize learning experiences and reduce errors in AI-generated responses. For instance, Lee (2024) designed a RAG-based statistics tutor to assist students with quantitative analysis, specifically addressing challenges like hallucination in LLMs(Lee, 2024). Similarly, Dong (2023) introduced a low-code framework for building AI tutors that leverage RAG technology to deliver accurate, context-aware feedback tailored to individual learners(Dong et al., 2025). Furthermore, Modran et al. (2024) proposed a chatbot tutoring system that combines RAG with custom LLMs, creating a platform for precise, contextually relevant, and personalized learning assistance(Modran et al., 2024).

Further research has demonstrated the broader adaptability of RAG-based educational tools. Thway et al. (2024) introduced "Professor Leodar," a chatbot leveraging RAG to deliver personalized guidance while minimizing misinformation(Thway et al., 2024). Zheng et al. (2024) assessed the robustness of RAG systems in K–12 education, focusing on discrepancies between textbook content and AI-generated responses(Zheng et al., 2024). Their work underscores the importance of aligning such systems with authoritative sources to ensure reliability.

Specific to the field of mathematics education, RAG has contributed significantly by enhancing the effectiveness and adaptability of AI-driven tools, enabling more precise and context-aware learning support. Levonian et al. (2023) investigated its application in math question-answering systems, highlighting the trade-offs between maintaining grounded, fact-based responses and aligning with user preferences(Levonian et al., 2023). Their findings emphasize the importance of designing systems that support both educational accuracy and user engagement.

The versatility of RAG extends beyond tutoring systems to broader educational platforms. Liu et al. (2024) highlighted how RAG enables personalized and adaptive learning content, empowering educators to deliver more effective and responsive instruction(Liu et al., 2024).

These studies collectively illustrate the framework's potential to transform educational technology, particularly in tasks demanding reasoning, contextual understanding, and scalability.

B Data Augmentation Details

To enhance the model's generalization, we employ a structured data augmentation approach consisting of three key phases: prompt engineering, data generation, and quality control. This ensures that the generated samples align with real-world student misconceptions while maintaining consistency with the original dataset.

B.1 Prompt Engineering

The generation process begins with designing effective prompts that guide ChatGPT to produce meaningful synthetic data. Each prompt is carefully structured to capture common student misconcep-

tions while ensuring diversity in reasoning patterns. The prompt includes a mathematical construct, a multiple-choice question with distractors, and an explanation of the reasoning behind each incorrect response. This ensures that the generated samples are pedagogically relevant and aligned with the original dataset.

A typical prompt template used for augmentation is as follows:

> **Prompt Template:**
>
> *Given the following multiple-choice mathematics question:*
>
> **ConstructName:** {Most granular level of knowledge related to question}
>
> **SubjectName:** {More general context than the construct}
>
> **Question:** {Mathematical Question Text}
>
> **Answer Choices:** {A, B, C, D}
>
> **Correct Answer:** {Correct Option}
>
> **Known Misconceptions:** {List of Misconceptions}
>
> *Generate a new incorrect response that a student might select based on a misunderstanding. Provide a brief explanation of the thought process that led to the incorrect answer.*

These modifications allow for the introduction of nuanced misconception patterns, ensuring a broad representation of potential student errors.

B.2 Data Generation

With the prompt engineering phase established, the data generation process involves extracting questions from the original dataset and synthesizing new misconception-based distractors. The model is instructed to generate plausible incorrect answers while preserving the logical and linguistic consistency of the problem format.

During this phase, ChatGPT is guided to produce errors that mimic actual student mistakes, drawing on existing misconception patterns. The generated distractors introduce variations in reasoning, helping the retrieval model distinguish between subtle conceptual misunderstandings. This approach ensures that the synthetic data remains both realistic and educationally relevant.

B.3 Quality Control and Dataset Integration

To ensure high data quality, we implement a rigorous validation process involving three expert annotators. Each generated sample undergoes independent review, where annotators evaluate its correctness, coherence, and consistency with known misconception patterns. This process filters out duplicate entries and logically inconsistent samples, ensuring that only well-structured data is retained.

The validation process includes duplicate detection, where cosine similarity is applied to identify and remove redundant samples, and logical consistency checks, where annotators verify that each misconception reflects plausible student reasoning. Through this careful quality assurance protocol, we curate a final set of 9,200 high-quality augmented samples that are then integrated into the training pipeline.

The augmented dataset is combined with the original training set and incorporated into 5-fold cross-validation experiments. This integration enables a more comprehensive evaluation of the retrieval system while maintaining consistency with the original dataset structure. By incorporating augmented data, the model is better equipped to retrieve and rank misconceptions accurately, leading to improved generalization and retrieval robustness.

B.4 Example Output

Question: What is the equation of this circle? (The circle goes through the points (4,0), (0, -4), (-4,0), and (0,4).)

Answer Choices:

- **A.** $x^2 + y^2 = 16$ (Correct Answer)
- **B.** $x^2 + y^2 = 8$ (Incorrect - Misconception: Confusing radius r with diameter, using $r = 4$ but mistakenly squaring half the radius instead of the full radius.)
- **C.** $x^2 + y^2 = 4$ (Incorrect - Misconception: Misinterpreting the radius as the distance to one axis point, rather than the full extent of the circle.)
- **D.** $x^2 + y^2 = 32$ (Incorrect - Misconception: Mistakenly doubling the radius before squaring it, treating $r^2 = 2 \times 4^2 = 32$ instead of $4^2 = 16$.)

C Implementation Details

Computing Environment. The experiments are conducted on a machine equipped with an NVIDIA A100 GPU, utilizing CUDA and PyTorch frameworks. This setup provides the necessary infrastructure to support both the training and inference processes for the retrieval and re-ranking models.

Training Details. The retriever is based on the pre-trained `BAAI/ bge-large-en-v1.5` model, which is fine-tuned on the misconception dataset using a multiple negative ranking loss function. The training samples consist of one positive misconception and three hard negatives. Key hyperparameters include a learning rate of 2×10^{-5}, a batch size of 8 (with gradient accumulation over 16 steps), and a training duration of 2 epochs, with early stopping based on validation performance.

Similarly, the reranker utilizes the `BAAI/bge-reranker-large` model and is optimized with a margin ranking loss. The training process for the reranker also incorporates hard negative sampling. Hyperparameter settings are consistent with those of the retriever, including a learning rate of 2×10^{-5}, a batch size of 8, and early stopping after 3 epochs.

For the reasoning stage, we employ the `Qwen-2.5-32B-Instruct` model. Structured prompts are designed to minimize hallucinations and ensure that the model generates relevant outputs. To enhance efficiency, inference is accelerated using vLLM, reducing both latency and resource consumption.

These implementation choices are crucial in achieving robust performance, enabling the integration of semantic retrieval and large language model-based reasoning within the two-stage retrieval framework.

D Evaluation Metrics

To comprehensively assess the effectiveness of our proposed method, we employ a range of evaluation metrics that capture different aspects of retrieval and ranking performance. These metrics allow us to rigorously measure both the ranking quality and the retrieval coverage of our system.

Official Kaggle Evaluation. The primary metric used for official evaluation on Kaggle is Mean Average Precision at 25 (MAP@25). MAP@25 is a ranking-based metric that calculates the mean of average precision scores across queries, focusing on the top-25 retrieved misconceptions. It effectively quantifies the system's ability to prioritize relevant misconceptions within the highest-ranked results, making it a robust indicator of ranking performance.

Local Evaluation Metrics. In addition to MAP@25, we also conduct local evaluations using Recall@K ($K = 25$). These metrics allow us to assess the system's ability to retrieve relevant misconceptions at different levels of granularity. Specifically, Recall@K measures the proportion of relevant misconceptions covered within the top-K retrieved results, while Precision@K evaluates the proportion of relevant misconceptions among the top-K retrieved results. These metrics are particularly useful for understanding the trade-offs between recall and precision at different stages of the retrieval process.

E Ablation Study

To further understand the contributions of different components in our proposed method, we conduct an ablation study. The results are summarized in Table 4 and visually represented in Figure 3. We evaluat four variants of our model: (1) Only Retriever, (2) Retriever + Reranker, (3) Retriever + Reranker + LLM Reasoning, and (4) Retriever + Reranker + LLM Reasoning + Data Augmentation. Each variant was assessed using the MAP@25 metric on both Kaggle and local datasets.

Variant	MAP@25 (Kaggle)	MAP@25 (Local)
Only Retriever	0.315	0.313
Retriever + Reranker	0.409	0.412
+ LLM Reasoning	0.468	0.471
+ Data Augmentation	0.496	0.523

Table 4: Ablation Study Results

Impact of the Reranker. The first two rows in Table 4 and the corresponding bars in Figure 3 compare the performance of the "Only Retriever" variant with the "Retriever + Reranker" variant. The addition of the reranking component significantly improves the MAP@25 scores from 0.315 to 0.409 on Kaggle and from 0.313 to 0.412 locally. This indicates that the reranker effectively refines the initial retrieval results by reordering the documents based on their relevance to the query. The reranker's ability to capture more nuanced relation-

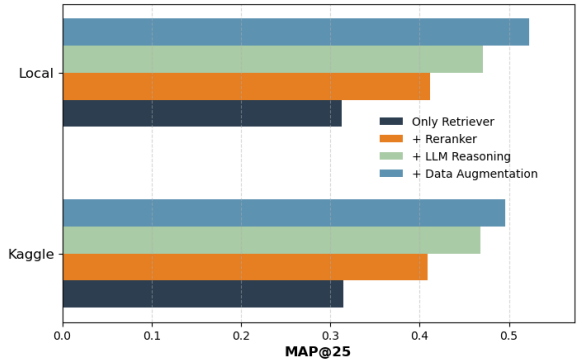

Figure 3: Ablation Study Results on different retrievers

ships between queries and documents contributes to this improvement.

Contribution of LLM Reasoning. The third row in Table 4 and the corresponding bar in Figure 3 show the results when LLM reasoning is added to the "Retriever + Reranker" variant. The MAP@25 scores further increase to 0.468 on Kaggle and 0.471 locally. This demonstrates the significant impact of incorporating large language model (LLM) reasoning in our framework. LLMs can provide deeper semantic understanding and context-aware reasoning, which enhances the model's ability to accurately rank relevant documents. The improved performance suggests that LLM reasoning complements the retriever and reranker effectively, leading to better overall retrieval quality.

Effect of Data Augmentation. The final row in Table 4 and the corresponding bar in Figure 3 evaluate the impact of data augmentation on the variant "Retriever + Reranker + LLM Reasoning". The addition of data augmentation techniques leads to a substantial improvement in MAP@25 scores, reaching 0.496 on Kaggle and 0.523 locally. Data augmentation helps the model generalize better by exposing it to a wider variety of training examples. This increased diversity in the training data enables the model to learn more robust representations and make more accurate predictions. The results confirm that data augmentation is a crucial component in enhancing the performance of our proposed method.

Summary and Insights. The ablation study provides valuable information on the contributions of each component in our proposed method. Sequential addition of the reranker, LLM reasoning, and data augmentation consistently improve performance in both evaluation settings. These findings highlight the importance of each design choice and validate the effectiveness of our comprehensive approach. In summary:

- The reranker significantly refines the initial retrieval results.
- LLM reasoning improves the model's semantic understanding and context-aware ranking.
- Data augmentation improves the model's generalization and robustness.

These results underscore the necessity of integrating these components for achieving superior retrieval performance. Future work could explore additional enhancements and optimizations to further improve the model's capabilities.

F Choice of k in Top-k Candidate Selection

Selecting an appropriate value for k in top-k candidate selection is crucial for balancing retrieval effectiveness and computational efficiency. Increasing k improves recall, as it allows for retrieving a broader set of candidates, but it also increases computational overhead. Conversely, choosing a smaller k can make the retrieval process more efficient while potentially missing relevant misconceptions.

To determine an optimal k, we conduct empirical experiments evaluating retrieval performance across different values of k, using **MAP@25** and **Recall@25** as the primary evaluation metrics. While a higher k increases recall, we observe diminishing returns in terms of MAP@25, suggesting that retrieving too many candidates introduces unnecessary noise. Additionally, increasing k significantly raises computational cost, which can be a bottleneck for real-time applications.

Table 5 presents the impact of different k values on retrieval effectiveness and computational efficiency. Based on these results, we select $k = 100$ as the optimal setting, balancing retrieval performance and efficiency.

k Value	MAP@25 (Kaggle)	Recall@k	Computational Cost (ms)/Question
50	0.466	0.871	4743
100	0.496	0.939	10572
200	0.505	0.972	23693
300	0.509	0.989	43580

Table 5: Comparison of Different k Values in Top-k Selection