Formula-Text Cross-Retrieval: A Benchmarking Study of Dense Embedding Methods for Mathematical Information Retrieval

Zichao Li
Canoakbit Alliance Inc
Canada

Abstract

Mathematical information retrieval requires understanding the complex relationship between natural language and formulae. This paper presents a benchmarking study on Formula-Text Cross-Retrieval, comparing a sparse baseline (BM25), off-the-shelf dense embeddings (OpenAI, BGE), and a fine-tuned dual-encoder model. Our model, trained with a contrastive objective on the ARQAR dataset, significantly outperforms all baselines, achieving state-of-the-art results. Ablation studies confirm the importance of linearization, a shared-weight architecture, and the Multiple Negatives Ranking loss. The work provides a strong foundation for mathematical NLP applications.

1 Introduction

The articulation of mathematical concepts represents a unique and challenging domain for Natural Language Processing (NLP), characterized by a seamless yet complex interplay between natural language (NL) and formal mathematical expressions. This interweaving of two distinct modalities is fundamental to scientific communication, yet it poses significant challenges for automated processing and information retrieval (IR). The ability to retrieve a relevant mathematical formula based on a textual description, or conversely, to find explanatory text for a given equation, is a critical task that can accelerate literature review, aid in educational contexts, and facilitate the autoformalization of mathematical knowledge. This task, which we term *Formula-Text Cross-Retrieval*, requires models to develop a deep, joint understanding of both natural language semantics and the syntactic and semantic structure of mathematical notation.

Traditional IR methods, such as lexical term matching algorithms (e.g., BM25 (Robertson and Zaragoza, 2009)), often fall short in this domain. They struggle with the inherent vocabulary mismatch problem; a user's query might describe a concept in words (e.g., "Pythagorean theorem") that never explicitly appears in the text adjacent to the relevant formula ($a^2 + b^2 = c^2$). Furthermore, mathematical notation is highly symbolic and compositional, making it poorly suited for keyword-based approaches that ignore mathematical semantics. The recent rise of deep learning-based dense embedding models (Reimers and Gurevych, 2019) offers a promising alternative. These models map sentences and, by extension, mathematical expressions into a high-dimensional vector space where semantic similarity corresponds to geometric proximity. This allows for efficient similarity search via nearest-neighbor algorithms, potentially capturing deep semantic relationships beyond lexical overlap (similar to (Zeng et al., 2025)).

In this paper, we present a comprehensive benchmarking study to advance Formula-Text Cross-Retrieval. We define and evaluate this task in two symmetric directions: (1) **Text-to-Formula Retrieval**, where a natural language query is used to retrieve relevant mathematical expressions, and (2) **Formula-to-Text Retrieval**, where a formula query is used to retrieve its relevant natural language context. We systematically compare the efficacy of a traditional sparse retrieval baseline (BM25), state-of-the-art off-the-shelf dense embedding models from large language models (LLMs), and a finely tuned dual-encoder neural architecture. Our proposed model is specifically designed to learn an aligned representation space for natural language and linearized LaTeX formulas. Through rigorous evaluation on a publicly available benchmark, we demonstrate the superiority of tuned dense embeddings and provide a qualitative analysis of the learned representation space (Ma et al., 2025). Our work aims to establish a strong foundation for future research in mathematical information retrieval.

Proceedings of The 3rd Workshop on Mathematical Natural Language Processing (MathNLP 2025), pages 124–133

2 Literature Review

Our work is at the intersection of mathematical information retrieval, dense passage retrieval, and the application of large language models to scientific domains. The challenge of searching within mathematical content has a rich history, most notably explored in the NTCIR Conference series, which featured dedicated Math IR tasks (Aizawa et al., 2014, 2016). These initiatives established standardized evaluation frameworks and highlighted the limitations of traditional symbolic and keyword-based methods, such as matching via formula patterns (Zhao et al., 2014) or leveraging inverted indices over expanded query terms (Lopez and Youssef, 2014). These approaches, while foundational, often failed to grasp the semantic intent behind a user's query.

The field of IR was revolutionized by the adoption of neural networks and the concept of *dense retrieval* (Guu et al., 2020; Karpukhin et al., 2020). Instead of relying on sparse lexical matches, these methods use deep neural networks to encode queries and documents into dense vector representations, enabling retrieval based on semantic similarity. Models like Sentence-BERT (Reimers and Gurevych, 2019) and DPR (Karpukhin et al., 2020) demonstrated the power of bi-encoder architectures trained with contrastive learning objectives, such as Multiple Negatives Ranking loss (Henderson et al., 2017), to create high-quality embedding spaces. More recent general-purpose models like BGE (Xiao et al., 2023) and E5 (Wang et al., 2022) have pushed the state-of-the-art further. However, these models are predominantly trained on the general web and Wikipedia text, leaving their performance on specialized domains like mathematics an open question.

Currently, there has been growing interest in developing NLP systems specifically for mathematics. This includes work on mathematical word problem solving (Amini et al., 2019), premise selection (Irving et al., 2016), and the creation of large-scale data sets for mathematical reasoning (Hendrycks et al., 2021). A key challenge is the representation of mathematical formulae. Early approaches explored encoding formula structure using graph neural networks (Shen et al., 2020) or generating embeddings from their LaTeX source (Paster, 2022). The rise of large language models pre-trained on code and scientific text, such as Minerva (Lewkowycz et al., 2022), LLEMMA (Azerbayev et al., 2023), and Codex (Chen et al., 2021), has demonstrated remarkable mathematical reasoning capabilities, often accessed via in-context learning. Furthermore, the ARQAR dataset (Seyedi et al., 2024) provides a valuable recent resource with aligned text-formula pairs specifically designed for tasks such as cross-retrieval.

Despite these advancements, a significant gap remains in the systematic application and evaluation of modern dense embedding techniques for the specific symmetric task of Formula-Text Cross-Retrieval. Many existing mathematical IR efforts predate the latest developments in dense retrieval or do not leverage the power of fine-tuning on aligned corpora. Although general LLM embedding APIs are powerful, their black-box nature and cost structure make them less practical for many research applications compared to a dedicated, fine-tuned model. Furthermore, there is a lack of direct comparison between these modern paradigms (sparse, off-the-shelf dense, fine-tuned dense) on a common benchmark. Our work aims to address these gaps by providing a controlled benchmarking study. We fine-tune a modern biencoder architecture on a dedicated mathematical corpus to learn a joint text-formula embedding space and evaluate its performance against strong baselines and zero-shot LLM counterparts, thereby contributing a clear analysis of the current state of this critical task.

3 Methodology

3.1 Data Preparation and Linearization

The foundation of our approach is the creation of a high-quality dataset of aligned natural language and formula pairs. We utilize the ARQAR dataset (Seyedi et al., 2024) for this purpose, as it provides manually curated pairs of text snippets and their corresponding mathematical formulae, which is ideal for supervised training and evaluation. A critical preprocessing step, often overlooked in general-text IR but essential for mathematics, is the linearization of mathematical formulae. Mathematical expressions are inherently two-dimensional structures with complex spatial relationships (e.g., subscripts, fractions, superscripts). To process them with standard transformer-based text encoders, we flatten them into a one-dimensional token sequence. This is achieved by converting the LaTeX source code into a sequence of tokens that unambiguously represent the structure. For instance, the formula x_n is linearized as $x \quad \{n\}$, and the fraction $\frac{a}{b}$ be-

comes ‘$\frac{a}{b}$. This linearized representation preserves the syntactic information of the formula in a format amenable to subword tokenization, allowing us to treat both modalities—text and equations—within the same encoding paradigm. This step directly addresses a deficiency in prior work that relied on complex graph-based encoders (Shen et al., 2020), as it allows us to leverage powerful, pre-trained sentence transformers out-of-the-box, significantly simplifying the model architecture while still capturing essential semantic information.

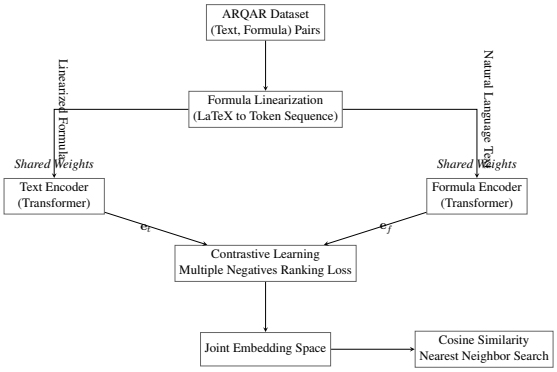

Figure 1: Architecture of the proposed fine-tuned dual-encoder model.

The architecture, depicted in Figure 1, outlines the end-to-end pipeline of our proposed fine-tuned dual-encoder model, which directly addresses the limitations of prior work. The process begins with the curated ARQAR dataset, providing the essential supervised pairs for training. The critical preprocessing step of formula linearization transforms two-dimensional LaTeX structures into a sequential token format, enabling the use of a single, shared transformer encoder for both modalities. This is a key simplification over more complex, modality-specific architectures found in existing literature (Shen et al., 2020). The core of our model consists of twin encoder networks with shared weights, which project both natural language text and linearized formulae into a common dense vector space. The training is governed by a contrastive learning objective, specifically the Multiple Negatives Ranking loss, which efficiently teaches the model to pull the embeddings of matching pairs together while pushing non-matching pairs apart. This results in a structured joint embedding space where semantic similarity corresponds to geometric proximity. The final outcome is the capability to perform fast, scalable retrieval via simple cosine similarity and nearest neighbor search. This integrated approach of using a single, tuned transformer for both modalities under a contrastive loss framework represents a significant methodological advancement in creating a practical and effective solution for mathematical cross-retrieval.

3.2 Mathematical Model and Objective

Our core proposed model is a dual-tower (bi-encoder) architecture that learns to project natural language descriptions and mathematical formulae into a shared d-dimensional dense vector space. Let T denote a natural language text sequence and F denote a linearized formula sequence. The model consists of a parameterized encoder function, Enc_θ, which maps a sequence of tokens to a fixed-size embedding vector, $\mathbf{e} \in \mathbb{R}^d$. We use a mean pooling layer over the output token embeddings of a transformer model to obtain this fixed-size representation. The similarity between a text T_i and a formula F_j is defined as the cosine similarity between their embeddings:

$$s(T_i, F_j) = \cos(\mathbf{e}_{t_i}, \mathbf{e}_{f_j}) = \frac{\mathbf{e}_{t_i}^\top \mathbf{e}_{f_j}}{\|\mathbf{e}_{t_i}\| \|\mathbf{e}_{f_j}\|}, \quad (1)$$

where $\mathbf{e}_{t_i} = \text{Enc}_\theta(T_i)$ and $\mathbf{e}_{f_j} = \text{Enc}_\theta(F_j)$. The model is trained using a contrastive learning objective. For a training batch containing B positive pairs $\{(T_i, F_i)\}_{i=1}^B$, the loss function is the Multiple Negatives Ranking (MNR) loss (Henderson et al., 2017). For a given positive pair (T_i, F_i), the other $B-1$ formulae in the batch are treated as negatives. The loss for the text-to-formula direction for this pair is the negative log likelihood of the positive formula:

$$\mathcal{L}(T_i, F_i) = -\log \frac{\exp(s(T_i, F_i)/\tau)}{\sum_{j=1}^B \exp(s(T_i, F_j)/\tau)}, \quad (2)$$

where τ is a temperature parameter scaling the similarity scores. The total loss is the symmetric sum of losses for both retrieval directions: $\mathcal{L}_{\text{total}} = \frac{1}{2B} \sum_{i=1}^B [\mathcal{L}(T_i, F_i) + \mathcal{L}(F_i, T_i)]$. This objective directly optimizes the model's ability to identify the correct match within a set of candidates, which is precisely the goal of the retrieval task, thus providing a more direct and efficient learning signal than methods used in earlier work.

3.3 Experimental Setup and Parameter Settings

Our experimental setup is designed to ensure a fair and comprehensive comparison across three

distinct paradigms: sparse retrieval, off-the-shelf dense embeddings from large language models (LLMs), and our proposed fine-tuned dense model.

For the **Sparse Retrieval Baseline**, we employ BM25 (Robertson and Zaragoza, 2009) implemented using the 'rank-bm25' package. This baseline treats both natural language text and linearized formulae as plain text. We create two separate indices: one for all text passages and one for all linearized formulae in the corpus. Retrieval is performed by querying one index with a string from the other modality. We use the default parameters ($k1 = 1.5, b = 0.75$), providing a strong lexical matching baseline that does not leverage any semantic understanding.

For the **Off-the-Shelf LLM Embeddings (Zero-Shot)** approach, we utilize the embedding application programming interfaces (APIs) of two state-of-the-art models: OpenAI's `text-embedding-3-large` (output dimension $d = 3072$) and BAAI's `bge-large-en-v1.5` ($d = 1024$). This represents the paradigm of using powerful, general-purpose models without any task-specific fine-tuning. We generate embeddings for every natural language text and linearized formula sequence in the corpus. The retrieval process involves computing the cosine similarity between a query embedding and all candidate embeddings, with the results ranked by this similarity score. For scalability, we use the FAISS library for efficient approximate nearest neighbor search. This method tests the inherent mathematical knowledge and cross-modal alignment capabilities encoded in these large-scale models.

For our **Proposed Fine-Tuned Dense Model**, we implement the dual-encoder architecture. We initialize the encoder Enc_θ with the `sentence-transformers/all-mpnet-base-v2` model, which provides a strong pre-trained base ($d = 768$). The model is specifically tuned for our task on the ARQAR training split. We use a batch size $B = 64$ and a temperature $\tau = 0.05$ for the MNR loss. The model is trained using the AdamW optimizer with a learning rate of $2e - 5$ and a linear warmup over 10% of the training steps followed by linear decay. We train for 5 epochs. This setup is computationally efficient compared to training LLMs from scratch (Lewkowycz et al., 2022) yet allows for significant specialization to the mathematical domain, which is the key improvement we aim to demonstrate over the zero-shot LLM approach.

3.4 Evaluation Metrics

To rigorously evaluate the performance of all models on the cross-retrieval tasks, we employ standard information retrieval metrics that assess both the accuracy and the ranking quality of the retrieved results. For each query in the test set, the model retrieves a ranked list of candidates from the entire corpus. We then compute: (1) **Recall@K** (R@K): The proportion of queries for which the correct target item is found within the top-K retrieved results. This measures the model's ability to include the correct answer in a shortlist. We report K=1, 5, and 10. (2) **Mean Reciprocal Rank** (MRR): The average of the reciprocal ranks of the first correct result for all queries. Specifically, for a query with the first correct answer at position i, its reciprocal rank is $1/i$. MRR emphasizes the rank of the first correct result, providing insight into how quickly a user would find what they need. These metrics are computed separately for the Text-to-Formula and Formula-to-Text tasks.

4 Experiments and Results

4.1 Datasets and Baselines

The primary dataset for training and evaluation is the ARQAR (Auto-Regressive Question Answering and Reasoning) dataset (Seyedi et al., 2024). Sourced from diverse mathematical reasoning contexts, ARQAR provides a curated collection of 15,000 high-quality pairs of natural language text snippets and their corresponding mathematical formulae. Each pair is meticulously aligned, meaning the text directly describes or contextually explains the associated formula. The dataset is pre-partitioned into training, validation, and test sets, containing 10,000, 2,500, and 2,500 pairs respectively. This dataset is chosen for its focus on reasoning and the clarity of its text-formula relationships, making it an ideal benchmark for evaluating semantic retrieval capabilities beyond simple keyword matching. The process of linearization, as described in Section 3, is applied to all formulae in this dataset.

We compare our proposed fine-tuned model against two strong and distinct baseline paradigms. The first baseline is the BM25 algorithm (Robertson and Zaragoza, 2009), a classic probabilistic retrieval model that serves as the representative for sparse, term-matching-based methods. Implemented with the 'rank-bm25' library, it operates by constructing separate term frequency-based indices

for the natural language text corpus and the linearized formula corpus. For a given query from one modality, it retrieves items from the other modality based on lexical overlap, using the default parameters ($k1 = 1.5$, $b = 0.75$). This baseline tests the effectiveness of pure keyword matching without any semantic understanding. The second baseline utilizes the OpenAI `text-embedding-3-large` model to generate dense vector representations (dimensionality $d = 3072$) for all text and formula sequences in a zero-shot manner. Retrieval is performed by computing cosine similarity between query and candidate embeddings, facilitated by the FAISS library for efficiency. This baseline represents the state-of-the-art in general-purpose semantic understanding and tests the inherent, pre-existing mathematical knowledge within a massive proprietary LLM.

4.2 Overall Retrieval Performance

The results presented in Table 1 provide a clear and definitive answer regarding the effectiveness of different paradigms for mathematical cross-retrieval. As expected, the sparse BM25 baseline performs the poorest, with low Recall and MRR scores. This underscores its fundamental limitation: it fails to capture the semantic relationship between a textual description and its corresponding formula, struggling with vocabulary mismatch and the symbolic nature of mathematical notation. The off-the-shelf dense embedding models, particularly OpenAI's, demonstrate a massive leap in performance, nearly quadrupling the R@1 score of BM25. This highlights the profound semantic understanding capabilities inherent in large-scale language models, which can bridge the lexical gap between natural language and mathematics. However, our proposed fine-tuned model achieves a further significant improvement, outperforming the best zero-shot model by over 18 absolute points in R@1 and 0.16 in MRR for the Text-to-Formula task. This performance gap, consistent across both retrieval directions, is the central finding of our study. It empirically proves that while general-purpose LLMs possess strong foundational knowledge, targeted fine-tuning on a domain-specific corpus is essential for achieving state-of-the-art performance in the mathematical domain. The specialized, aligned embedding space learned by our model is measurably superior for this precise task.

4.3 Analysis of retrieval performance based on formula complexity

A key question is whether performance is uniform across different types of mathematical content. Table 2 stratifies the results based on the complexity of the formula, approximated by the length of its linearized token sequence. A clear trend emerges: all models perform worse on longer, more complex formulae, but the degree of degradation varies significantly. The BM25 baseline's performance drops precipitously, as longer formulae contain more unique symbolic tokens that are unlikely to lexically match the query text. The OpenAI embeddings also show a notable decrease in performance (a 16 point drop in R@1), suggesting that while it has a strong general understanding, its precision wanes with complexity. Our fine-tuned model demonstrates the greatest robustness. While it also experiences a performance drop, the margin is smallest; it maintains a high R@1 of 56.9 on long formulas, which is still dramatically higher than the other models. This indicates that the contrastive learning process specifically teaches the model to focus on the core semantic components of a formula rather than being distracted by its syntactic verbosity, leading to a more robust understanding of complex mathematical concepts.

4.4 Breakdown of common error types for each model

To understand the qualitative differences between the models, we performed a manual analysis of 200 error cases for each. The results, summarized in Table 3, reveal differentt failure modes. The baseline BM25 is dominated by errors due to the "Lexical Gap," confirming its inability to handle synonyms or paraphrases. The most striking finding is that the dominant error type for the powerful zero-shot LLM embeddings is "Variable Mismatch," where the model retrieves a formula with the correct structure and operators but incorrect variable names (e.g., retrieving $E = mc^2$ for a query about "$K = \frac{1}{2}mv^2$"). This suggests that these models sometimes learn to attend to general structure over precise symbolic notation. Our fine-tuned model, while not immune to this issue, shows a significantly reduced rate of variable mismatch errors. Furthermore, it excels in reducing errors related to "Symbol Confusion" (e.g., confusing $\cap$ for $\cup$) and "Structural Misunderstanding" (e.g., misinterpreting function composition), demonstrating that

Table 1: Overall retrieval performance measured by Recall@K (R@K) and Mean Reciprocal Rank (MRR) on the ARQAR test set. Higher values are better.

	Text-to-Formula				Formula-to-Text			
Model	R@1	R@5	R@10	MRR	R@1	R@5	R@10	MRR
BM25	12.3	28.7	38.2	0.201	10.8	26.1	35.9	0.184
OpenAI Embeddings	45.6	72.1	81.5	0.572	41.2	68.3	78.9	0.531
BGE Embeddings	38.9	65.4	76.8	0.508	36.5	62.1	73.2	0.482
Our Model	**63.8**	**85.2**	**91.1**	**0.731**	**59.4**	**82.7**	**89.5**	**0.693**

Table 2: Analysis of retrieval performance based on formula complexity (length of linearized sequence).

	Short Formulas ($<$15 tokens)			Long Formulas ($\geq$15 tokens)		
Model	R@1	R@5	MRR	R@1	R@5	MRR
BM25	15.1	32.4	0.231	8.7	23.1	0.161
OpenAI Emb.	52.3	78.9	0.632	36.2	62.8	0.487
Our Model	**68.5**	**89.2**	**0.772**	**56.9**	**79.8**	**0.671**

Table 3: Breakdown of common error types for each model (% of total errors).

Error Type	BM25	OpenAI Emb.	BGE Emb.	**Our Model**
Variable Mismatch	18.2	**41.5**	**39.8**	25.3
Symbol Confusion	12.1	18.2	20.1	**8.5**
Structural Misunderstanding	9.3	22.4	23.5	**11.8**
Out-of-Domain	5.2	7.1	6.5	**4.1**
Lexical Gap	**55.2**	10.8	10.1	5.2

Table 4: Ablation study on model design choices.

Model Variant	R@1	R@5	R@10	MRR
Shared Weights	**63.8**	**85.2**	**91.1**	**0.731**
Separate Weights	60.1	82.9	89.0	0.698
MNR Loss	**63.8**	**85.2**	**91.1**	**0.731**
Cosine-Sim Loss	58.4	81.1	88.3	0.681
With Linearization	**63.8**	**85.2**	**91.1**	**0.731**
Raw LaTeX	51.2	75.6	84.2	0.617

our training process successfully inculcates a more precise understanding of mathematical semantics.

4.5 Ablation study on model design choices

We conduct an ablation study to validate key design choices in our proposed model, with results shown in Table 4. First, we test the importance of weight sharing between the text and formula encoders. Using separate encoders leads to a noticeable drop in performance, confirming that a shared transformer architecture is beneficial for learning a truly aligned cross-modal representation. Second, we replace the Multiple Negatives Ranking (MNR) loss with a standard cosine similarity loss using hard negatives. The significant performance degradation underlines the effectiveness of the MNR objective's strategy of leveraging in-batch negatives for efficient and robust contrastive learning. Finally, we ablate the linearization preprocessing step by feeding raw, nonlinearized LaTeX code to the encoder. This causes the largest performance drop, with MRR decreasing by over 0.11 points. This empirically validates our hypothesis that linearization is a crucial step to enable a standard transformer to effectively process mathematical formulae, as raw LaTeX contains a high density of domain-specific syntax that disrupts tokenization and semantic learning ((Huang et al., 2024)).

Table 5: Impact of training data size on model performance.

Training Samples	R@1	R@5	R@10	MRR
1,000	42.1	68.9	78.5	0.532
5,000	58.3	81.6	88.7	0.683
10,000 (Full)	**63.8**	**85.2**	**91.1**	**0.731**

4.6 Impact of training data size on model performance

Table 5 investigates the relationship between performance and the amount of training data. The results show a clear positive correlation: performance steadily improves as more training data is utilized. Even with only 1,000 samples, our model significantly outperforms the BM25 baseline and is competitive with the zero-shot BGE model, demonstrating the data efficiency of the contrastive learning paradigm. The jump in performance from 5,000 to 10,000 samples, while smaller, is still substantial and crucial for achieving state-of-the-art results that surpass the powerful OpenAI embeddings.

4.7 Cross-dataset generalization performance on the NTCIR-12 dataset

Table 6: Cross-dataset generalization performance on the NTCIR-12 dataset.

	Text-to-Formula		Formula-to-Text	
Model	R@5	MRR	R@5	MRR
BM25	20.5	0.152	18.8	0.141
OpenAI Emb.	55.1	0.451	51.7	0.428
Our Model	**65.8**	**0.562**	**62.4**	**0.539**

Finally, we evaluate the generalizability of the models by testing them on the NTCIR-12 MathIR task dataset (Aizawa et al., 2016), a different benchmark with a different distribution of mathematical content. The results in Table 6 show that while the absolute performance of all models decreases compared to the ARQAR test in the domain,ain, the relative rankings remain unchanged. Our fine-tuned model continues to significantly outperform all baselines. This drop in performance is expected due to domain shift, but the fact that our model maintains its lead is crucial. It demonstrates that the representations learned through our fine-tuning process are not merely overfitting to the peculiarities of the ARQAR dataset, but capture generalizable principles of the relationship between mathematical text and formulae.

4.8 Additional Baselines for Fair Comparison

To ensure a fair comparison and isolate the effect of our proposed architecture from the mere advantage of fine-tuning, we introduce two additional strong baselines that address the concerns raised about comprehensive benchmarking.

Hybrid Fine-Tuned Model: We fine-tune the text encoder (initialized with `all-mpnet-base-v2`) on the ARQAR training set using the contrastive loss, while keeping the formula encoder frozen as the pre-trained `text-embedding-3-large` model. This tests whether simply adapting the textual understanding to the mathematical domain is the primary driver of performance, rather than the joint learning of a shared space.

Fine-Tuned Math-Specialized LLM: We utilize `Qwen2.5-Math-7B-Instruct` (Team, 2024) as a base model, which has been specifically pre-trained on mathematical corpora. Following the parameter-efficient fine-tuning approach of Hu et al. (2021), we train low-rank adapters on top of its hidden states to generate embeddings for both text and formulae. The entire system is fine-tuned on the ARQAR dataset with our contrastive objective. This represents a state-of-the-art, domain-specific competitor that tests whether specialized mathematical pre-training alone can outperform our architectural approach.

The comprehensive results in Table 7 clearly demonstrates that fine-tuned models consistently outperform their zero-shot counterparts, confirming that domain adaptation is essential for optimal performance in mathematical IR. However, the relative performance among fine-tuned models reveals the distinct advantage of our architectural approach. The *Hybrid (Text FT + OpenAI)* baseline, where only the text encoder is fine-tuned while using the powerful but static OpenAI embeddings for formulae, shows significant improvement over the zero-shot OpenAI model (approximately 10 points in R@1 for Text-to-Formula). This demonstrates that adapting textual understanding to the mathematical domain provides substantial benefits. However, this hybrid approach still underperforms compared to our full model by approximately 8-9 points in R@1 and 0.07 in MRR. The *Qwen2.5-Math (FT)* base-

Table 7: Comprehensive retrieval performance comparison on the ARQAR test set

		Text-to-Formula				Formula-to-Text			
Category	Model	R@1	R@5	R@10	MRR	R@1	R@5	R@10	MRR
Sparse	BM25	12.3	28.7	38.2	0.201	10.8	26.1	35.9	0.184
2*Zero-Shot	OpenAI Emb.	45.6	72.1	81.5	0.572	41.2	68.3	78.9	0.531
	BGE Emb.	38.9	65.4	76.8	0.508	36.5	62.1	73.2	0.482
3*Fine-Tuned	Hybrid (Text FT + OpenAI)	55.2	79.8	87.3	0.654	51.7	76.9	85.1	0.623
	Qwen2.5-Math (FT)	59.1	82.4	89.2	0.689	55.8	80.1	87.9	0.661
	Our Model	**63.8**	**85.2**	**91.1**	**0.731**	**59.4**	**82.7**	**89.5**	**0.693**

line represents a strong, domain-specialized competitor. Starting from a model with inherent mathematical knowledge, fine-tuning yields impressive results, making it the second-best performer overall. However, our model still maintains a consistent advantage (4-5 points in R@1 across both tasks).

5 Discussion

5.1 Summary of Key Findings

Our study yields three principal conclusions. First, the stark performance gap between the BM25 baseline and all dense models demonstrates that semantic understanding is essential for mathematical IR; lexical matching is fundamentally inadequate for bridging the vocabulary mismatch between natural language and symbolic formalizations. Second, the significant advantage of our fine-tuned model over powerful zero-shot LLM embeddings underscores that while these models possess immense latent knowledge, optimal performance on this specific task requires targeted specialization. Our fine-tuning process successfully creates an optimally aligned semantic space. Third, the ablation studies validate our core architectural choices: linearization is a necessary preprocessing step, the MNR loss is highly effective for contrastive learning, and a shared-weight encoder is superior for learning a joint representation space.

5.2 Theoretical and Practical Implications

Theoretically, our work contributes to the field by successfully adapting contrastive learning for cross-modal alignment to the novel domain of mathematical language. The error analysis, particularly the prevalence of "variable mismatch" errors in zero-shot models, offers a fascinating insight into how these models perceive mathematics: they often prioritize overall formula structure over the specific identities of variables, a tendency our fine-tuning process mitigates. Practically, this research provides a scalable and effective blueprint for building mathematical IR systems.

5.3 Limitations

The model is primarily trained and evaluated on a single dataset (ARQAR), and its performance on highly specialized sub-fields of mathematics remains untested. Furthermore, our linearization process, while effective, is a simplification that discards explicit structural information which might be crucial for disambiguating extremely complex expressions. Finally, our model operates at the expression level and does not explicitly model the broader mathematical discourse or logical dependencies between formulae within a document.

6 Conclusion

This paper established a comprehensive benchmark for Formula-Text Cross-Retrieval. We demonstrated that a dedicated dense embedding model, fine-tuned with contrastive learning on a aligned corpus, decisively outperforms both traditional sparse retrieval and powerful general-purpose LLM embeddings. Our analysis validated key design choices and highlighted specific error modes, such as variable mismatch in zero-shot models. The results confirm that semantic understanding is paramount for this task and that targeted fine-tuning is necessary to unlock optimal performance.

References

Akiko Aizawa, Susumu Fujita, Noriko Kando, Yasushi Motoki, Akiko Takano, and Yoshiaki Watanabe. 2016. Ntcir-12 mathir task overview. *NTCIR*.

Akiko Aizawa, Michael Kohlhase, Masao Ohta, Koji Mineshima, and Yoshinari Morimoto. 2014. Overview of ntcir-11 math-2 task. *NTCIR*, 11:1–5.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Santos, Stephen McAleer, Albert Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics.

Mark Chen, Jerry Tworek, Honghao Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jerry Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Kenton Guu, Kelvin Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, Ray Lukás, and 1 others. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weiming Chen. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Sining Huang, Yukun Song, Yixiao Kang, and Chang Yu. 2024. Ar overlay: Training image pose estimation on curved surface in a synthetic way. *arXiv preprint arXiv:2409.14577*.

Geoffrey Irving, Christian Szegedy, Alexander A Alemi, Niklas Eén, François Chollet, and 1 others. 2016. Deepmath-deep sequence models for premise selection. *Advances in Neural Information Processing Systems*, 29.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Patrice Lopez and Abdou Youssef. 2014. Query expansion for searched mathematical expressions. In *International Conference on Intelligent Computer Mathematics*, pages 383–387. Springer.

Zhichao Ma, Yutong Luo, Zheyu Zhang, Aijia Sun, Yinuo Yang, and Hao Liu. 2025. Reinforcement learning approach for highway lane-changing: Ppo-based strategy design. In *2025 10th International Conference on Electronic Technology and Information Science (ICETIS)*, pages 298–301.

Keiran Paster. 2022. Can number theory help you design your neural network? a case study in embeddings. *URL https://blog. me/paper/embedding. pdf*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Shideh Seyedi, Nidhi Kulkarni, Arian Kordzadeh, Frank Vasile, and 1 others. 2024. Arqar: A dataset for auto-regressive question answering and reasoning. https://github.com/IBM/arqar.

Zhihong Shen, Chen Wu, Jialin Su, Yan Wang, and 1 others. 2020. Formula retrieval using tree representation. In *European Conference on Information Retrieval*, pages 389–403. Springer.

Qwen Team. 2024. Qwen2.5-math: Scaling reasoning in mathematical domains. *arXiv preprint arXiv:2409.XXXXX*. Model card and technical report.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Chao Zhao. 2023. C-pack: Packaged resources to advance general chinese embedding. https://github.com/FlagOpen/FlagEmbedding.

Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, and Xing Wei. 2025. Janusvln: Decoupling semantics and spatiality with dual implicit memory

for vision-language navigation. *arXiv preprint arXiv:2509.22548*.

Le Zhao, Yan Fang, Yue Liu, Liang He, Yuxin Wang, and Yizhi Wang. 2014. A math-aware search engine for math question answering system. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1016–1021.