FinAlign

Matteo Pietro Di Pilato 535298 - Lorenzo Cinquemani 535781 - Lorenzo Marchioni 535388
Emails: matteopietro.dipilato01@universitadipavia.it, lorenzo.cinquemani01@universitadipavia.it, lorenzo.marchioni01@universitadipavia.it

November 12, 2025

Abstract
This project addresses the challenge of Financial Question Answering (FiQA) by developing a hybrid retrieval system designed to bridge the vocabulary gap between non-expert user queries and technical financial documents. Our approach integrates traditional lexical retrieval (BM25) with dense retrieval (BGE-base) and explores advanced query expansion techniques, including "Hypothetical Document Embeddings" (HyDE) and technical keyword expansion using Large Language Models (LLMs). Key findings demonstrate that LLM-driven query rewriting significantly outperforms traditional baselines by effectively bridging the domain-knowledge gap for non-expert users. Furthermore, monoT5 reranker improved even more the performance, highlighting the importance of multi-stage pipelines in domain-specific retrieval.

1 Introduction and Motivation

In the financial domain, finding precise answers is often a challenge because traditional search engines rely on literal keyword matching. This approach frequently fails to bridge the gap between complex documentation and the way people actually ask questions. By leveraging the FiQA dataset, this project moves beyond simple word-matching to develop a system that understands true user intent.
Our system is designed primarily for non-experts who may lack deep financial literacy, and we believe that clear access to financial information helps individuals make better life decisions. To support this mission, our research investigates a central question:

> *Does the combination of query rewriting, prefix-based encoding, and document chunking improve the effectiveness of a hybrid (BM25 + dense) retrieval system?*

We hypothesize that enriching queries, explicitly guiding dense encoders with prefixes, and retrieving at the chunk level can mitigate semantic mismatch and improve retrieval quality.

2 Task and Dataset Description

2.1 Task Definition

The task is an opinion-based and explanatory retrieval problem, where the goal is to retrieve documents that help answer financial questions rather than returning short factual answers. Queries may ask for advice, explanations, or interpretations of financial concepts.

2.2 Dataset

Dataset: We utilized the FiQA (Financial Question Answering) dataset from the BEIR benchmark.

- Corpus: contains approximately 57,000 documents, including financial news headlines, microblog posts, and report snippets.
- Queries: a set of natural language questions (e.g., "What is the difference between a REIT and a real estate stock?").
- Relevance Judgments (Qrels): graded relevance annotations used for evaluation.

Challenges:

- Heterogeneity: The document collection mixes formal reports with informal, noisy social media text.
- Vocabulary Gap: Significant mismatch between user query terms and document content.
- Sparsity: Many queries have very few relevant documents, making recall a primary challenge.

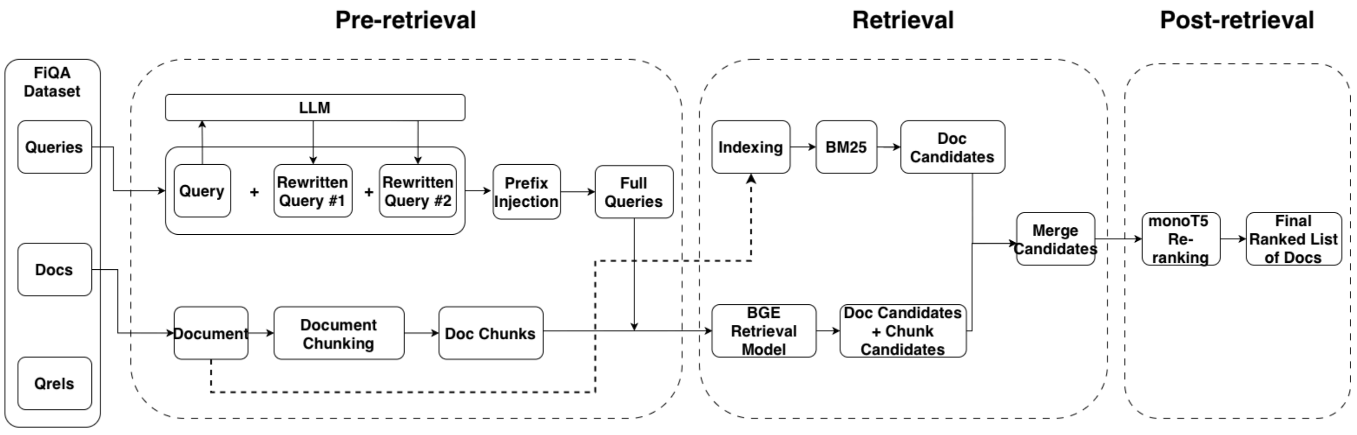
3 Methodology

3.1 Baseline Systems (Phase I)

We started with traditional models to set a benchmark:

- TF-IDF & BM25: We employed PyTerrier's standard implementation of TF-IDF and BM25. BM25 served as our primary strong baseline due to its robustness in exact term matching.
- BM25 + RM3: We included pseudo-relevance feedback (RM3) to test if statistical query expansion could address the vocabulary gap without external models.

3.2 Advanced Systems (Phase II)



3.2.1 Query Rewriting and HyDE-style Expansion

Financial queries in FiQA are often short, vague, and expressed in non-technical language. To reduce semantic mismatch, we apply LLM-based query rewriting to generate enriched query variants that introduce domain-specific terminology and make implicit financial concepts explicit.
In addition, we adopt a HyDE-style (Hypothetical Document Embeddings) approach, where an LLM generates a short hypothetical answer for each query. These hypothetical texts are not retrieved directly but are embedded and used to guide dense retrieval as semantic proxies of the user's information need, enabling effective matching even when lexical overlap is limited.
The original query, rewritten variants, and HyDE-generated representations are combined using score fusion to improve recall while maintaining robustness.

3.2.2 Hybrid BM25 + BGE Dense Retrieval with Query Prefixes

We implement a hybrid retrieval architecture combining BM25 and dense retrieval using a BGE bi-encoder. BM25 serves as a first-stage retriever to ensure lexical recall, while dense retrieval addresses vocabulary mismatch through semantic similarity.
For dense retrieval, we adopt query-only prefix-based encoding by prepending the instruction

"Represent this sentence for searching relevant financial passages:"

to all queries, while leaving documents unprefixed. This design explicitly guides the encoder to interpret queries as retrieval instructions without altering the document representation space.
BM25 and dense retrieval results are combined using both rank-based and score-based fusion strategies.

3.2.3 Document Chunking and Passage-Level Retrieval

FiQA documents vary in length and often contain multiple topics. To reduce semantic dilution in long texts, documents are split into overlapping fixed-length chunks prior to dense indexing.
Dense retrieval is performed at the chunk level, and chunk scores are aggregated to the document level using max pooling, allowing highly relevant passages to promote their parent documents in the final ranking.

3.2.4 Hybrid Retrieval and Score Fusion

To combine evidence from multiple retrieval signals, we evaluate two fusion strategies: Reciprocal Rank Fusion (RRF) and weighted linear score combination.
RRF merges ranked lists without score normalization, offering robustness when different retrievers perform well on different queries. In parallel, we apply linear fusion after min-max normalization to explicitly weight dense retrieval signals derived from rewritten queries and HyDE representations. Both strategies are evaluated to compare rank-based robustness with score-based control.

3.2.5 Neural Re-ranking with MonoT5

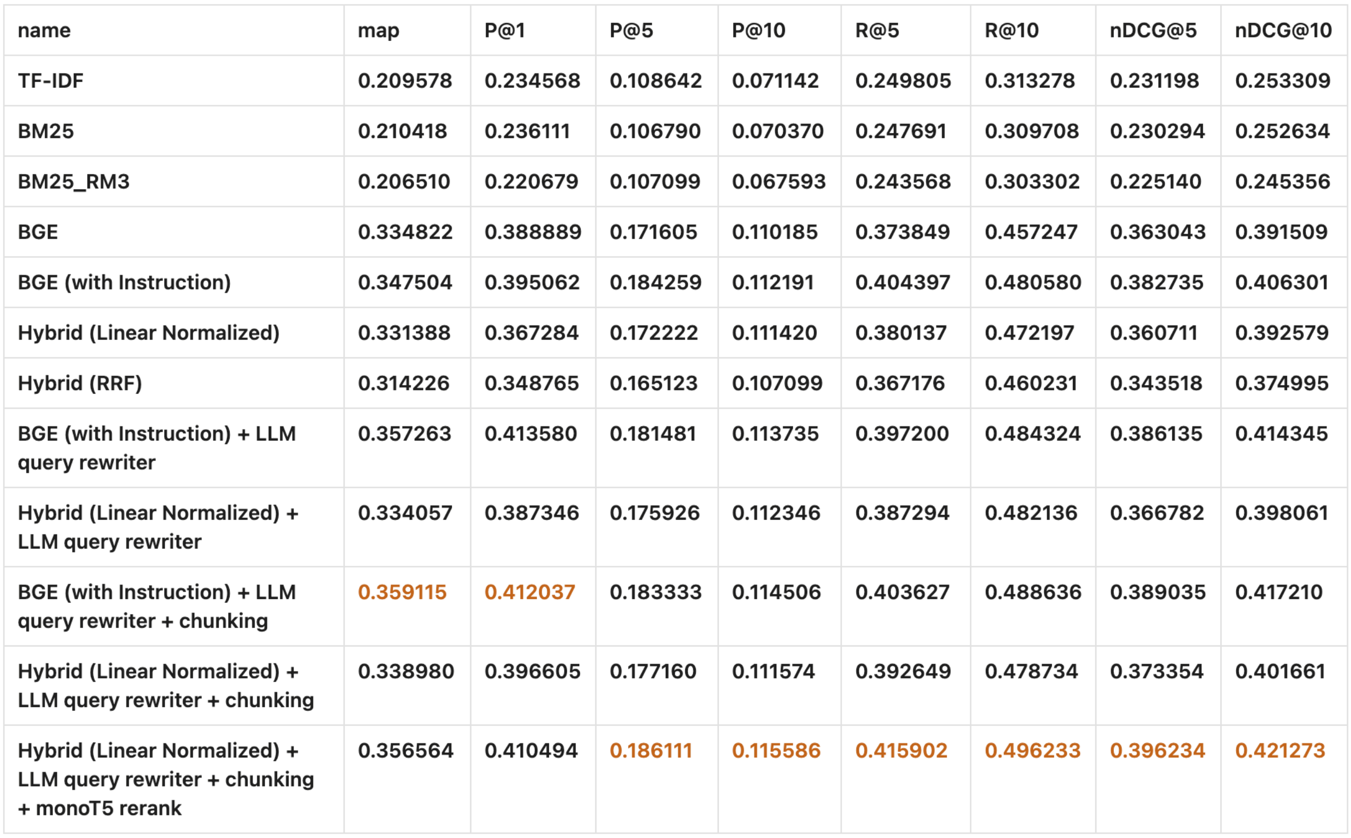
As a final stage, we apply a monoT5 cross-encoder to re-rank the top retrieved candidates. MonoT5 jointly encodes query-document pairs, enabling fine-grained relevance modeling beyond bi-encoder similarity. Re-ranking is applied only to a restricted candidate set due to computational constraints.

3.3 Implementation Details

The system is implemented in Python using established IR and NLP frameworks.
PyTerrier is used as the main experimental framework, with Terrier providing inverted indexing and BM25 retrieval.
Neural components rely on Hugging Face Transformers for LLM-based query rewriting, HyDE generation, and MonoT5 re-ranking. Dense retrieval is implemented using Sentence-Transformers with a BGE bi-encoder, while FAISS supports efficient vector similarity search.
All FiQA documents are indexed in full. A lexical index is built for BM25 using standard preprocessing, while a dense index encodes documents (or chunks) with BGE embeddings. To handle document length variability, chunking is applied before dense indexing, and chunk scores are aggregated at the document level using max pooling.
Queries are processed according to the experimental configuration: original queries for BM25, rewritten and HyDE-generated variants for dense retrieval. All dense queries are prefixed with the same retrieval instruction, while documents remain unprefixed. BM25 is used for first-stage retrieval, dense results are fused using RRF and linear combination, and MonoT5 is applied as a final re-ranking stage on a restricted candidate set.

4 Experiments and Results

The evaluation was conducted on a cleaned version of the **BEIR/FiQA test set**, consisting of **57,600 documents** and **648 unique queries** after filtering empty entries.



name	map	P@1	P@5	P@10	R@5	R@10	nDCG@5	nDCG@10
TF-IDF	0.209578	0.234568	0.108642	0.071142	0.249805	0.313278	0.231198	0.253309
BM25	0.210418	0.236111	0.106790	0.070370	0.247691	0.309708	0.230294	0.252634
BM25_RM3	0.206510	0.220679	0.107099	0.067593	0.243568	0.303302	0.225140	0.245356
BGE	0.334822	0.388889	0.171605	0.110185	0.373849	0.457247	0.363043	0.391509
BGE (with Instruction)	0.347504	0.395062	0.184259	0.112191	0.404397	0.480580	0.382735	0.406301
Hybrid (Linear Normalized)	0.331388	0.367284	0.172222	0.111420	0.380137	0.472197	0.360711	0.392579
Hybrid (RRF)	0.314226	0.348765	0.165123	0.107099	0.367176	0.460231	0.343518	0.374995
BGE (with Instruction) + LLM query rewriter	0.357263	0.413580	0.181481	0.113735	0.397200	0.484324	0.386135	0.414345
Hybrid (Linear Normalized) + LLM query rewriter	0.334057	0.387346	0.175926	0.112346	0.387294	0.482136	0.366782	0.398061
BGE (with Instruction) + LLM query rewriter + chunking	0.359115	0.412037	0.183333	0.114506	0.403627	0.488636	0.389035	0.417210
Hybrid (Linear Normalized) + LLM query rewriter + chunking	0.338980	0.396605	0.177160	0.111574	0.392649	0.478734	0.373354	0.401661
Hybrid (Linear Normalized) + LLM query rewriter + chunking + monoT5 rerank	0.356564	0.410494	0.186111	0.115586	0.415902	0.496233	0.396234	0.421273

4.1 Evaluation Setup

System performance was measured using standard Information Retrieval metrics, in particular we used:**MAP (Mean Average Precision),**, **P@1**, **P@5**, **P@10, R@5, R@10**, **nDCG@10, nDCG@5**. Relevance was determined by a set of **1,705 qrels** provided by the dataset.

All retrieval systems are evaluated on the same test collection using a fixed set of queries and relevance judgments. Relevance is assessed using the provided qrels, and only documents present in the final indexed corpus are considered, ensuring consistency across runs.

The following evaluation metrics are reported:

- **MAP (Mean Average Precision)**, to measure overall ranking quality across all relevant documents.
- **Precision@k (P@1, P@5, P@10)**, to evaluate early precision, which is particularly important for question answering scenarios.
- **Recall@k (R@5, R@10)**, to assess how many relevant documents are retrieved in the top ranks.
- **nDCG@k (nDCG@5, nDCG@10)**, to capture ranking quality while accounting for the position of relevant documents.

All systems are evaluated under identical conditions, and results are directly comparable since the same queries, relevance judgments, and evaluation protocol are used for every run.

4.2 Results

This section presents the results of our experiments, comparing various retrieval pipelines from simple baselines to advanced multi-stage systems. We focus on how each component contributes to improve the performances analyzing standard IR metrics.

Baseline Performance and Lexical Retrieval

TF-IDF and BM25 achieve very similar performance across all evaluation metrics, MAP ≈ 0.21 and nDCG@10 ≈ 0.25, indicating that improvements in lexical weighting alone are insufficient to address the main challenges of the FiQA task. Although BM25 incorporates term saturation and document length normalization, its advantage over TF-IDF is marginal.

Dense Retrieval and Semantic Matching

Dense retrieval with BGE outperforms lexical baselines over all metrics. This improvement highlights that FiQA is fundamentally a semantic retrieval task. Dense models effectively capture paraphrases, implicit financial concepts, and explanatory relevance, which are poorly handled by lexical matching. Financial terminology often appears in varied surface forms and dense embeddings are better suited to bridge these variations.

Effect of Instruction-Tuned Embeddings

Instruction-tuned BGE further improves performance over standard dense retrieval across all reported metrics. Explicitly framing queries as retrieval instructions reduces embedding ambiguity and aligns query representations more closely with retrieval intent.

Hybrid Retrieval and Fusion Behavior

In our tests, the hybrid approach combining BM25 and BGE did not perform as well as using the BGE model alone. While we expected the two signals to complement each other, the results showed that the dense retriever was already strong enough to capture most of the important information. Consequently, adding BM25 signals actually led to a slight decrease in overall performance compared to the results from instruction-tuned BGE.
The Reciprocal Rank Fusion (RRF) method also delivered disappointing results, with performances significantly below other hybrid methods.

Query Rewriting and Multi-Query Retrieval

Applying LLM-based query rewriting before dense retrieval produces moderate improvements, while combining multiple rewritten queries leads to clearer gains. The best-performing rewriting setup leverages the original query together with rewritten and hypothetical variants, improving both MAP and early precision.
These results suggest that rewriting is most effective when treated as query diversification rather than substitution. In FiQA, where queries are often vague or underspecified, multiple semantic views help cover different interpretations of user intent without excessively harming precision.

Document Chunking

Document chunking is a key design choice in our system, as it changes the retrieval unit from entire documents to smaller, semantically focused passages. This is particularly important in FiQA, where relevant information is often localized within short explanatory spans rather than distributed across full documents.
The impact of chunking is clear. The BGE-Chunked-LLM-Linear pipeline consistently improves MAP and nDCG@10 compared to its non-chunked counterpart, indicating that chunk-level retrieval reduces semantic dilution in dense embeddings.
These results suggest that chunking is not merely a preprocessing detail but a meaningful retrieval strategy, especially when combined with semantic modeling.

Advanced Pipelines and Re-Ranking

The strongest overall results are achieved by advanced multi-stage pipelines combining dense retrieval, LLM-based query rewriting, document chunking, and re-ranking. In particular, pipelines incorporating monoT5 re-ranking achieve the highest nDCG@10 and recall values.

These results confirm that FiQA benefits from late-stage relevance modeling, where cross-encoder-style re-ranking captures nuanced financial reasoning that bi-encoders alone cannot model. Chunking further amplifies this effect by exposing localized relevance signals to the re-ranker.

5. Discussion and Conclusions

This project provided valuable insights into the challenges and design trade-offs involved in financial information retrieval, particularly for opinion-based question answering over heterogeneous data such as FiQA.

What We Learned

A key lesson is that combining multiple retrieval models is non-trivial and requires careful understanding of each component's assumptions and limitations. We observed that dense retrievers such as BGE benefit significantly from document chunking, as they are more effective when operating on shorter, semantically focused text segments. Chunking mitigates the issue of long documents exceeding token limits and reduces semantic dilution, leading to more accurate similarity matching.

Query rewriting proved more challenging than expected. While LLM-based rewriting can improve recall in some cases, many generated rewrites were either too generic or insufficiently aligned with dense retrievers. This highlighted that rewriting quality is critical. In practice, small instruction-tuned models produced limited gains, while larger models were computationally impractical in constrained environments.
Another important finding concerns instruction-based query prefixing. Properly prefixing queries for BGE substantially improved dense retrieval performance, confirming that understanding a model's training paradigm and best practices is often as important as the choice of model itself.

Finally, we learned that computational constraints matter. Running complex multi-stage pipelines involving dense retrieval, query rewriting, and re-ranking on limited hardware (e.g., Colab with T4 GPUs) imposed significant runtime and memory limitations, directly influencing which approaches were feasible.

Main Difficulties

The primary challenges encountered were related to computational cost, model integration, and data heterogeneity. Many experiments were expensive to run, making extensive ablation studies difficult. Integrating multiple models, especially LLM-based query rewriting, introduced practical issues related to input formats, latency, and pipeline coordination.

Future Work

Several promising directions emerge from this work. First, using larger or better fine-tuned models for both dense retrieval and query rewriting could yield further improvements. Second, systematic hyperparameter tuning, particularly for linear fusion weights between original, rewritten, and HyDE-based queries, could significantly improve performance beyond the configurations explored here.

Another interesting direction is difficulty-aware re-ranking, where different re-ranking strategies are applied depending on query ambiguity or confidence signals from earlier stages. Finally, extending the retrieval system into a retrieval-augmented generation (RAG) pipeline could improve usability for non-expert users by synthesizing retrieved information into concise, natural-language answers.

Takeaway

Overall, this project demonstrates that effective financial information retrieval requires more than strong individual models: it depends on careful pipeline design, semantic alignment, and practical engineering choices. Hybrid retrieval, document chunking, and instruction-aware dense encoders provide clear benefits, but only when combined thoughtfully. These findings reinforce the importance of system-level reasoning in building robust retrieval pipelines for real-world financial question answering.

AI Tools and Assistance Declaration (Required)

AI tools such as ChatGPT, Claude, Gemini, and GitHub Copilot were used only for support during coding and debugging, and for minor assistance in improving the clarity of the final report.
All design choices, implementations, experiments, and results were entirely developed by the authors.