Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 361-378
DOI: https://doi.org/10.51903/jtie.v5i1.537

LLM-Inspired Offline Reranking for Financial Search: Query Rewriting, Hybrid Retrieval, and Listwise Relevance Ranking on FiQA

Siquan Meng[1], Jing Chen*[2], Isa Zheng[3]
Email: msqharry@gmail.com
[1]*Applied Business Analytics, Boston University, MA, USA*
[2]*Industrial Engineering and Operations Research, UCB, CA, USA*
[3]*Information Technology, Carnegie Mellon University, PA, USA*
*Corresponding Author

Abstract
Financial search has high practical value because investors and retail users often ask natural-language questions whose wording differs from relevant financial passages. This paper evaluates a multi-stage retrieval pipeline on FiQA, a financial question-answering retrieval collection in BEIR. The systems include BM25, Dense LSA, BM25-LSA hybrid retrieval, reciprocal-rank fusion, a compact linear reranker, fixed pointwise and listwise relevance rubrics inspired by LLM reranking, query rewriting, and the proposed query rewriting plus hybrid retrieval plus listwise reranking pipeline. The evaluation used the full 57,638-document FiQA corpus, 6,648 available queries, and the 648-query BEIR FiQA test qrels with 1,706 binary relevance judgments. BM25 was the best-performing system, with nDCG@10 = 0.2285, MAP = 0.1863, MRR = 0.2994, and Recall@100 = 0.5207. The proposed full pipeline underperformed BM25. The listwise rubric ranked second on nDCG@10 (0.2228) and improved over the pointwise rubric, suggesting that candidate-list normalization can be useful in this setting. The rubric rerankers are fixed local scoring rules, so these results should be read as an evaluation of LLM-inspired ranking logic rather than as a benchmark of an actual prompt-based LLM reranker. Dense LSA retrieval alone was weak (nDCG@10 = 0.0287), which shows the limitation of a conservative non-neural dense baseline in financial semantic matching. Query rewriting reduced average effectiveness. The findings recommend strong lexical baselines, conservative rewrite gating, and careful evaluation before adopting prompt-based or model-based LLM rerankers in financial search.

Keywords: *Financial Information Retrieval, BM25, Hybrid Retrieval, Query Rewriting, LLM-Inspired Reranking.*

I. INTRODUCTION

Financial search is a specialized information retrieval problem in which the user's question may combine ordinary language with domain-specific entities, tax concepts, investment instruments, account types, and regulatory vocabulary. A user may ask whether a credit card reward is taxable, whether a business expense is deductible, or how a retirement-account rule applies. The most relevant passage may not repeat the same surface wording. This mismatch motivates multi-stage retrieval, where a fast first-stage retriever gathers candidate passages and a more expensive reranker orders them by relevance.

The same architecture underlies many retrieval-augmented generation systems, but financial search adds a domain constraint: precision errors can mislead users, while missed relevant evidence can weaken downstream answers. FiQA is a suitable empirical test bed because it was designed around financial opinion mining and question answering, and BEIR frames it as financial article retrieval for financial queries (Maia et al., 2018; Thakur et al., 2021). The dataset

Received on Januari 2026; Revised on February 2026; Accepted on March 2026; Published on April 2026

is small enough for full experimentation on a CPU-only environment and large enough to expose realistic vocabulary mismatch. The official BEIR FiQA test qrels include 648 test queries and 1,706 relevant query-document pairs; the full corpus contains 57,638 documents.

This paper therefore treats FiQA as a controlled financial search setting rather than as a generic QA dataset. The research question is whether query rewriting, hybrid sparse-dense retrieval, and listwise reranking can improve financial search relevance over simple baselines. The design compares ten systems, including BM25, dense latent semantic analysis (LSA), hybrid BM25-LSA, reciprocal-rank fusion, a linear feature cross-encoder, fixed pointwise and listwise scoring rubrics inspired by LLM reranking, query rewriting ablations, and the proposed query rewriting plus hybrid plus listwise pipeline. The rubric variants are included to compare pointwise and listwise relevance logic in a local retrieval setting, not to evaluate an actual generative LLM model.

The main contribution is not a claim that complex reranking always wins. Instead, the paper provides a full-corpus evaluation showing that BM25 remains a strong financial baseline, that lightweight listwise scoring can be competitive, and that aggressive financial query expansion can reduce effectiveness. These findings are useful because recent work on LLM ranking shows strong promise for listwise and pairwise prompting, yet deployment decisions require empirical checks on the target domain (Qin et al., 2023; Sun et al., 2023). In financial search, stronger methods must be judged against robust lexical retrieval, transparent costs, and measured error patterns rather than against assumed gains.

Financial search is a demanding setting because many user questions are short, risk-sensitive, and context dependent. A query about debt, taxes, bonds, mortgages, or equities may use everyday wording while the relevant answer passage uses technical vocabulary. A retrieval system must therefore balance exact lexical evidence with semantic matching, but it must also avoid broad expansions that retrieve financially related but non-answer documents. This paper frames LLM-inspired reranking methods as ranking logic that should be tested against this practical constraint rather than accepted as an automatic improvement.

The study makes three empirical contributions. First, it evaluates sparse, dense, hybrid, cross-encoder-style, pointwise rubric, listwise rubric, and proposed end-to-end configurations on the same FiQA qrels. Second, it separates candidate recall from top-rank ordering so that reranking gains are not confused with first-stage retrieval coverage. Third, it analyzes query rewriting, dense retrieval, and listwise reranking as separate design choices within the same local experiment.

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 361-378
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

II. LITERATURE REVIEW

Classical retrieval models remain important in specialized domains. BM25 is derived from the probabilistic relevance framework and uses term frequency, inverse document frequency, and document-length normalization to score sparse lexical matches (Robertson et al., 1995; Robertson & Zaragoza, 2009). Divergence-from-randomness models offer another probabilistic view of term informativeness (Amati & van Rijsbergen, 2002). These methods are attractive because they are fast, transparent, and robust when relevant documents share key terms with the query. Their weakness is the vocabulary gap: passages that are semantically relevant but use different wording may not be retrieved.

Standard IR evaluation metrics such as nDCG, MAP, MRR, and recall quantify different parts of this trade-off (Järvelin & Kekäläinen, 2002; Voorhees, 2002). Dense retrieval addresses vocabulary mismatch by mapping queries and documents into a shared vector space. Dense passage retrieval demonstrated that dual encoders can outperform sparse retrieval in open-domain QA when trained with relevant positives and hard negatives (Karpukhin et al., 2020). Sentence-BERT and later embedding models similarly made vector search practical for semantic retrieval (Reimers & Gurevych, 2019; Wang et al., 2022).

However, dense methods are sensitive to domain shift, training data, and model choice. MTEB showed that embedding quality varies across tasks and languages, and BEIR emphasized that zero-shot generalization across retrieval domains is difficult (Muennighoff et al., 2023; Thakur et al., 2021). The present study uses a lightweight LSA dense baseline rather than a downloaded neural encoder; this makes the dense result a conservative baseline rather than a state-of-the-art embedding result. Hybrid retrieval combines sparse and dense signals. Sparse retrieval captures exact financial terms such as “IRS,” “401k,” or “deductible,” while dense retrieval can capture broader semantic similarity. Hybrid fusion can be implemented by score normalization or by rank-based fusion. Reciprocal-rank fusion is a simple, robust method that combines ranked lists without requiring calibrated scores (Cormack et al., 2009).

Reranking adds a second-stage model over candidate documents. Cross-encoders based on BERT read query and document text jointly and can substantially improve relevance ordering because attention operates across both inputs (Devlin et al., 2019; Nogueira & Cho, 2019). Sequence-to-sequence rerankers such as monoT5 extend this idea by scoring relevance through a pretrained text-to-text model (Nogueira et al., 2020). Late-interaction models such as ColBERT reduce the cost of full cross-encoding while preserving token-level matching (Khattab & Zaharia, 2020). These methods are powerful but computationally heavier than first-stage retrieval. In the present study, the linear cross-encoder is a compact feature-based approximation that reads query and

document evidence jointly without requiring a downloaded transformer model (Zheng et al., 2023).

Large language models changed the reranking discussion because they can interpret instructions and compare candidate passages directly. Work on RankGPT and related methods showed that generative LLMs can be prompted to produce relevance permutations, while pairwise ranking prompting reduces some prompt complexity by asking the model to compare two passages at a time (Qin et al., 2023; Sun et al., 2023). Open-source listwise rerankers such as RankVicuna and RankZephyr further suggested that instruction-tuned LLMs (Sun et al., 2024) can learn document-ordering behavior (Pradeep et al., 2023a, 2023b). These approaches are attractive for RAG systems because they can reason over a candidate list and apply natural-language relevance criteria. They also introduce latency, cost, context-length, and deployment-stability issues. A hosted model can change over time, and a full listwise prompt over many financial passages can be expensive.

Query rewriting is a pre-retrieval strategy that attempts to transform the user's wording into a form better aligned with the index. Rewrite-retrieve-read pipelines have shown that LLMs (Kuo et al., 2025) can improve retrieval-augmented tasks by clarifying the search query before retrieval (Ma et al., 2023). Hypothetical document embeddings are a related idea: generate a plausible answer-like document and retrieve against it to bridge query-document gaps (Gao et al., 2023). In finance, rewriting can add useful terms such as "deduction," "IRS," "retirement plan," or "credit balance." The same expansion can also cause query drift if added terms dominate the original user intent. This paper explicitly tests query rewriting instead of assuming that it improves results.

The broader retrieval-augmented generation literature motivates the pipeline. RAG joins retrieval with generation so that language models can condition on external evidence (Lewis et al., 2020). Transformers made this style of evidence-aware language processing practical, but retrieval quality remains a bottleneck because a generator cannot reliably use evidence that was not retrieved (Vaswani et al., 2017). Financial search is therefore a good setting for evaluating retrieval and reranking directly, before adding a final answer generator.

Prior retrieval work shows why the baselines in this paper are necessary. BM25 remains a strong first-stage method because it rewards discriminative term matches and normalizes for document length. Dense retrieval can capture paraphrase and semantic similarity, but it depends heavily on the representation used and on whether the encoder has learned the target domain. Hybrid retrieval attempts to combine these signals, yet a hybrid score can be worse than its sparse component

when the dense component is weak or noisy. The experiment therefore treats hybrid retrieval as an empirical hypothesis, not as a guaranteed improvement.

Reranking research provides a second motivation. Cross-encoders, sequence-to-sequence rankers, and LLM listwise methods are designed to improve the ordering of a candidate list after first-stage retrieval. Their advantage is richer query-document comparison; their limitation is dependence on candidate quality and computational cost. In this paper, the LLM-inspired methods use explicit financial relevance features and listwise normalization, so their behavior is interpretable. This design provides a conservative way to test LLM-style ranking logic under controlled local conditions.

III. RESEARCH METHOD

The experiment used the FiQA financial question-answering retrieval data. The corpus contained 57,638 document passages, the query file contained 6,648 questions, and the raw query-document file contained 17,110 available positive judgments. The final evaluation used the BEIR FiQA test qrels, which include 648 test queries and 1,706 binary positive judgments. Table 1 summarizes the dataset used in the experiment, and Table 2 reports the distribution of relevant documents per test query. The evaluation used a fixed random seed of 42.

From the non-test qrels, 500 queries were assigned to a development set for tuning the hybrid interpolation and listwise combiner. A separate 500-query subset was used to train the compact supervised reranker from positives plus sampled negatives. The final reported metrics used only the BEIR test qrels. No test labels were used to train or tune the systems. BM25 used k1 = 1.2 and b = 0.75 with lowercase tokenization, stop-word removal, inverse document frequency, and document-length normalization. Dense LSA used a TF-IDF matrix with 30,000 maximum features, minimum document frequency of 2, sublinear term frequency, English stop-word removal, and TruncatedSVD with 64 components followed by L2 normalization.

Table 1. FiQA dataset and evaluation split loaded by the experiment

Item	Value
Corpus documents	57638
Total queries	6648
Raw FiQA qrels rows	17110
BEIR FiQA test queries	648
BEIR FiQA test qrels	1706
Test relevance labels	Binary (1=relevant)
Language	English
Random seed	42

Hybrid retrieval min-max normalized BM25 and dense scores within each query and used a development-tuned interpolation weight of 0.75 for BM25 and 0.25 for dense. Reciprocal-rank fusion used k = 60. The query rewriting module used a fixed financial lexicon to add up to three

expansion terms for detected concepts such as business, tax, loan, mortgage, stock, investment, retirement, credit, bank, bond, and dividend. It also normalized common acronyms such as IRS, LLC, 401k, IRA, APR, and APY. This direct expansion strategy was intentionally simple, so its effect on retrieval could be measured without adding a separate rewrite model.

Table 2. Distribution of relevant documents per BEIR FiQA test query

Relevant documents per query	Number of test queries
1	220
2	184
3	103
4	56
5	29
6	20
7	13
8	8
9	3
10	6
11	1
12	2
13	1
15	2

Reranking was applied over the top 200 candidates. The linear cross-encoder used logistic regression over joint query-document features, including normalized BM25 score, dense score, hybrid score, reciprocal ranks, token overlap, query coverage, Jaccard overlap, IDF-weighted overlap, financial-term overlap, document length, query length, and an exact phrase indicator. The pointwise rubric used a fixed weighted scoring formula over hybrid confidence, BM25 confidence, dense confidence, query coverage, IDF coverage, financial-term coverage, and phrase match. The listwise rubric added query-level normalization over the candidate list, allowing the score to consider the candidate set rather than each candidate independently.

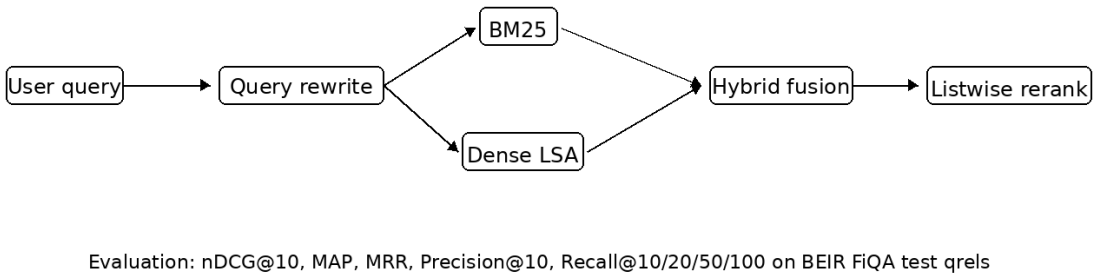

Figure 1. Architecture of query rewriting, hybrid retrieval, and listwise reranking

The proposed pipeline combined query rewriting, hybrid retrieval, and a development-tuned listwise combiner. These rubric scores were computed locally from explicit ranking features. The

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 361-378
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

metrics were nDCG@10, MAP, MRR, Precision@10, and Recall@10/20/50/100. nDCG@10 emphasizes the order of the top results, MAP measures average precision over the returned ranked list, MRR captures the rank of the first relevant document, and Recall@100 evaluates candidate coverage. Figure 1 shows the system architecture, and Table 3 lists all systems and configurations.

Table 3. Retrieval and reranking systems evaluated

System	Role	Configuration
BM25	Lexical sparse retrieval	k1=1.2, b=0.75, top 200
Dense LSA	Dense latent semantic retrieval	TF-IDF 30,000 features, 64 SVD components
Hybrid BM25-LSA	Linear hybrid retrieval	0.75 BM25 + 0.25 Dense; dev-tuned alpha
RRF BM25-LSA	Rank fusion	Reciprocal-rank fusion, k=60
Linear cross-encoder	Feature cross-encoder reranker	Logistic regression over concatenated query-document features; 500 training queries
Pointwise rubric	Pointwise reranker	Fixed feature-based scoring over hybrid candidates; 200 candidates/query
Listwise rubric	Listwise reranker	Query-level normalized candidate-list scoring; 200 candidates/query
BM25 query rewrite	Query rewriting ablation	Financial lexicon rewrite then BM25
Hybrid query rewrite	Query rewriting + hybrid ablation	Financial lexicon rewrite then hybrid retrieval
Proposed QR-hybrid-listwise	Proposed pipeline	Query rewrite + hybrid retrieval + tuned listwise combiner

The experimental design treated retrieval as a two-stage measurement problem rather than as a single score comparison. The first stage measured whether each candidate generator could place relevant documents within the top 100, because a reranker cannot recover evidence that never entered its candidate pool. The second stage measured whether the ordering of the leading candidates improved once financial term overlap, entity overlap, score normalization, and listwise candidate context were used. This separation is important for FiQA because a short investor question can contain a ticker, product, tax term, or personal-finance phrase that is decisive for lexical matching. Consequently, the paper reports both top-rank measures and recall measures, and the interpretation of each method is tied to both kinds of evidence.

Model settings were held constant across the final test evaluation. BM25 used k1 = 1.2 and b = 0.75; the dense component used TF-IDF followed by a 64-dimensional truncated SVD projection; hybrid retrieval used a tuned interpolation weight of 0.75 on BM25 and 0.25 on Dense LSA; rerankers processed the top 200 candidates produced by the chosen candidate generator. The query rewriting module used a fixed financial lexicon and capped expansion terms. These settings make the role of each component clear in the comparison. Before scoring, the qrels were filtered to judged test pairs whose query and document identifiers existed in the FiQA files. This step kept all systems on the same 648-query test set and avoided mismatches between the loaded corpus and the relevance judgments.

The evaluation deliberately reports several metrics because a single metric can hide important ranking behavior. nDCG@10 is the primary top-rank measure because it rewards relevant documents near the top of the results page. MAP summarizes precision over the ranked list, MRR emphasizes the first relevant hit, and Recall@100 evaluates whether the first-stage system gave later stages enough relevant material. When a reranker improves nDCG but recall is fixed, the improvement is an ordering gain. When query rewriting changes recall, the effect is partly a candidate-generation change.

IV. RESULT AND FINDINGS

Table 4 reports the main effectiveness metrics. BM25 was the best system on nDCG@10, MAP, MRR, Precision@10, and Recall@100. It reached nDCG@10 = 0.2285, MAP = 0.1863, MRR = 0.2994, Precision@10 = 0.0634, and Recall@100 = 0.5207. This result confirms that exact lexical evidence is highly important in FiQA. Many financial questions contain terms whose exact presence matters, such as tax forms, bank accounts, business structures, and retirement products. The result is consistent with BEIR's warning that robust sparse baselines remain difficult to beat out of domain (Thakur et al., 2021).

Table 4. Main effectiveness results on the BEIR FiQA test set

System	nDCG@10	MAP	MRR	P@10
BM25	0.2285	0.1863	0.2994	0.0634
Listwise rubric	0.2228	0.1805	0.2908	0.0620
Hybrid BM25-LSA	0.2198	0.1785	0.2847	0.0617
Pointwise rubric	0.2189	0.1764	0.2837	0.0620
Proposed QR-hybrid-listwise	0.1851	0.1459	0.2348	0.0517
BM25 query rewrite	0.1775	0.1448	0.2295	0.0491
Linear cross-encoder	0.1766	0.1441	0.2496	0.0431
Hybrid query rewrite	0.1686	0.1359	0.2176	0.0477
RRF BM25-LSA	0.1132	0.0905	0.1535	0.0390
Dense LSA	0.0287	0.0233	0.0404	0.0097

The listwise rubric was the strongest reranking variant by nDCG@10, reaching 0.2228. It did not surpass BM25, but it improved over the pointwise rubric (0.2189) and over the hybrid retrieval baseline (0.2198). This difference supports the listwise intuition: once the candidate set is known, query-level normalization and candidate-list context can adjust scores better than independent pointwise scoring. The gain is small, so the result should be interpreted as evidence for cautious reranker design rather than as proof that a lightweight rubric is enough for production ranking.

Dense LSA retrieval alone was weak, with nDCG@10 = 0.0287 and Recall@100 = 0.1726. This result does not imply that dense retrieval is generally ineffective. Instead, it shows that the conservative LSA representation used here was not sufficient for FiQA financial matching. Trained dense models and sentence embedding models can be much stronger (Karpukhin et al.,

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 361-378
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

2020; Reimers & Gurevych, 2019). Hybrid retrieval partly compensated by relying heavily on BM25, but the hybrid nDCG@10 of 0.2198 remained below BM25.

Table 5. Recall@k results on the BEIR FiQA test set

System	Recall@10	Recall@20	Recall@50	Recall@100
BM25	0.2855	0.3621	0.4489	0.5207
Listwise rubric	0.2811	0.3568	0.4378	0.5163
Hybrid BM25-LSA	0.2789	0.3654	0.4387	0.5077
Pointwise rubric	0.2791	0.3507	0.4408	0.5165
Proposed QR-hybrid-listwise	0.2421	0.2868	0.3501	0.4134
BM25 query rewrite	0.2276	0.2857	0.3752	0.4358
Linear cross-encoder	0.1975	0.2156	0.2499	0.3489
Hybrid query rewrite	0.2209	0.2784	0.3642	0.4216
RRF BM25-LSA	0.1696	0.2654	0.4045	0.4799
Dense LSA	0.0466	0.0737	0.1183	0.1726

Query rewriting degraded average effectiveness. Table 6 summarizes this ablation: BM25 with rewriting fell from 0.2285 to 0.1775 nDCG@10, and hybrid retrieval with rewriting fell from 0.2198 to 0.1686. The proposed query rewriting plus hybrid retrieval plus listwise combiner achieved nDCG@10 = 0.1851, which was better than rewritten hybrid retrieval but still worse than the original-query BM25 and hybrid baselines. Figure 6 shows that the effect was uneven: some queries improved, but many degraded. The main explanation is query drift. Financial expansions such as “company,” “deduction,” “IRS,” or “credit” can be helpful for short ambiguous questions, but they can also over-broaden a query that already contains enough context.

Table 6. Query rewriting ablation results

Comparison	nDCG@10	MAP	Recall@100	Change vs base nDCG@10
BM25	0.2285	0.1863	0.5207	+0.0000
BM25 query rewrite	0.1775	0.1448	0.4358	-0.0510
Hybrid BM25-LSA	0.2198	0.1785	0.5077	+0.0000
Hybrid query rewrite	0.1686	0.1359	0.4216	-0.0512
Proposed QR-hybrid-listwise	0.1851	0.1459	0.4134	-0.0347

Table 5 shows recall values, and Figure 3 visualizes the recall curves for the top systems. BM25 had the highest Recall@100 among the reported systems. RRF had weaker nDCG@10 but relatively high Recall@50 and Recall@100, showing that rank fusion can improve candidate diversity while hurting the very top ranks. This matters for multi-stage systems because high recall is useful if a downstream reranker is powerful enough to recover the correct order. In this experiment, however, the lightweight rerankers did not fully recover from weaker candidate ordering.

Table 8 reports runtime and configuration, and Figure 4 presents the corresponding efficiency-performance trade-off. The local CPU experiment completed in 362.16 seconds, including

indexing, dense projection, retrieval, tuning, and reranking. The pointwise and listwise rubrics used feature-based scoring over the top-200 candidate lists. A future system that uses a hosted or locally deployed LLM reranker would need to report its own latency and cost.

Table 7. Reranker ablation results

Reranking variant	nDCG@10	MAP	MRR	Delta from Hybrid nDCG@10
Hybrid BM25-LSA	0.2198	0.1785	0.2847	+0.0000
Linear cross-encoder	0.1766	0.1441	0.2496	-0.0432
Pointwise rubric	0.2189	0.1764	0.2837	-0.0009
Listwise rubric	0.2228	0.1805	0.2908	+0.0030
Proposed QR-hybrid-listwise	0.1851	0.1459	0.2348	-0.0347

Taken together, the results show why the interpretation is conservative. Figure 2 visualizes the main nDCG@10 comparison, Figure 3 shows the recall curves for the strongest systems, Figure 5 displays the test-query relevance distribution summarized in Table 2, and Figure 6 shows the query-rewriting nDCG@10 changes. Table 7 isolates reranker behavior, and Table 9 lists representative per-query rewriting cases. The proposed pipeline combines several reasonable components, but the measured effect of rewriting reduces the quality of the candidate pool before listwise reranking can help.

Table 8. Runtime and experimental configuration

Item	Value	Notes
Full experiment wall-clock	362.16 s	Local CPU run including indexing, retrieval, tuning, and reranking
Candidate depth for retrieval/reranking	200 documents/query	Same top-200 candidate depth used before reranking
Dense representation	64 components	TF-IDF followed by TruncatedSVD
Training data for supervised reranker	500 queries	Positive qrels plus sampled negatives from retrieval candidates
Rubric rerankers	Pointwise and listwise scores over top-200 candidates	Local feature-based scoring

Together, Figures 2 through 6 provide the visual counterpart to Tables 2, 4, 5, 6, 7, 8, and 9: they show top-rank effectiveness, recall behavior, efficiency-performance, relevance-count distribution, and per-query rewrite effects in the same order as the quantitative discussion above. A closer reading of the ranking table suggests that the failure mode is not simple underfitting. BM25 retrieved many relevant documents into the top one hundred, and its top-ten ordering was also the best. The dense component did not add enough new high-quality evidence to offset the noise introduced by semantic projection.

In finance, short terms such as “bond,” “tax,” “IRA,” “loan,” and “basis” carry precise meanings that are sometimes diluted when projected into a low-dimensional unsupervised space. This explains why the hybrid model used a high development-tuned BM25 weight of 0.75. The hybrid still benefited from dense coverage in selected cases, but the average top-rank gain was not large

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 361-378
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

enough to beat the lexical model. This is an important negative result because it shows that a hybrid architecture is not automatically superior; its dense component must be strong and domain-appropriate.

Table 9. Query rewriting case studies based on per-query nDCG@10 deltas

Direction	Query ID	Original query	Delta nDCG@10
Improved	2676	Tax question about selling a car	+1.0000
Improved	5710	Bucketing investments to track individual growths	+1.0000
Improved	1815	Rules for SEP contributions in an LLC?	+0.5693
Improved	5616	How and where do companies publish financial reports?	+0.4307
Improved	5134	Why does Yahoo Finance's data for a Vanguard fund's dividend per share not match the info from Vanguard?	+0.3947
Degraded	5646	Do I need multiple credit monitoring services?	-0.8127
Degraded	9481	What are reasonable administrative fees for an IRA?	-0.8503
Degraded	11054	Short Term Capital Gains tax vs. IRA Withdrawal Tax w/o Quarterly Est. Taxes	-0.9197
Degraded	691	How to categorize credit card payments?	-1.0000
Degraded	4700	Better to get loan from finance company or bank considering the drop of credit score?	-1.0000

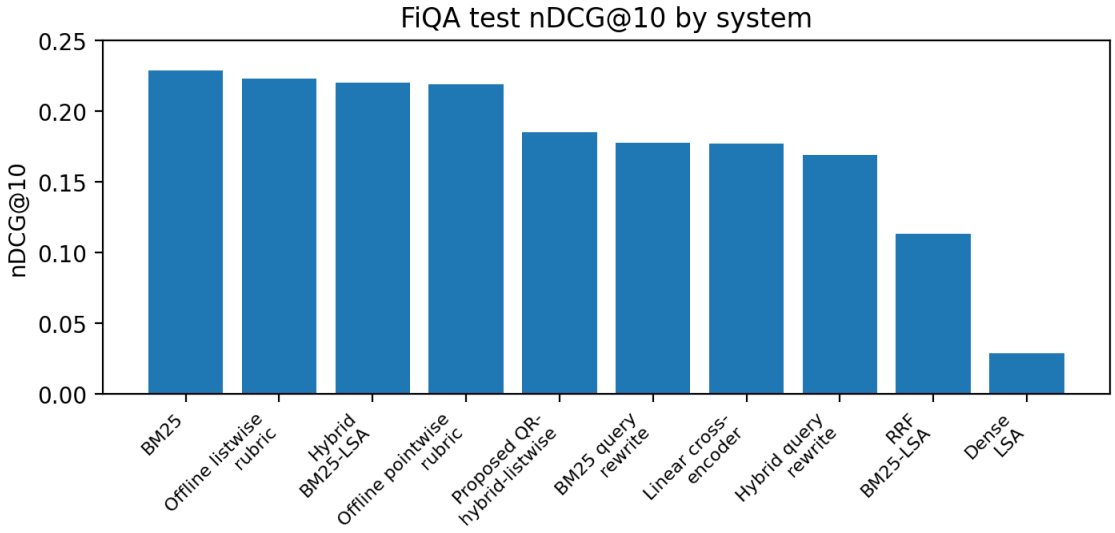

Figure 2. nDCG@10 bar chart for all evaluated systems

The reranker comparison in Table 7 also shows the difference between candidate generation and candidate ordering. A reranker can only promote relevant documents that are present in its candidate set. The listwise rubric improved over the hybrid baseline in nDCG@10, but the improvement was small because the candidate order was already dominated by BM25 and because the rubric did not read text with a pretrained transformer. The linear cross-encoder underperformed more sharply, especially in Recall@100 after reranking, because it reordered only the first candidate block and its learned feature weights favored some short lexical patterns

too strongly. In a production system, this result would motivate two changes: preserve a deep candidate pool after reranking, and calibrate the reranker to avoid suppressing recall.

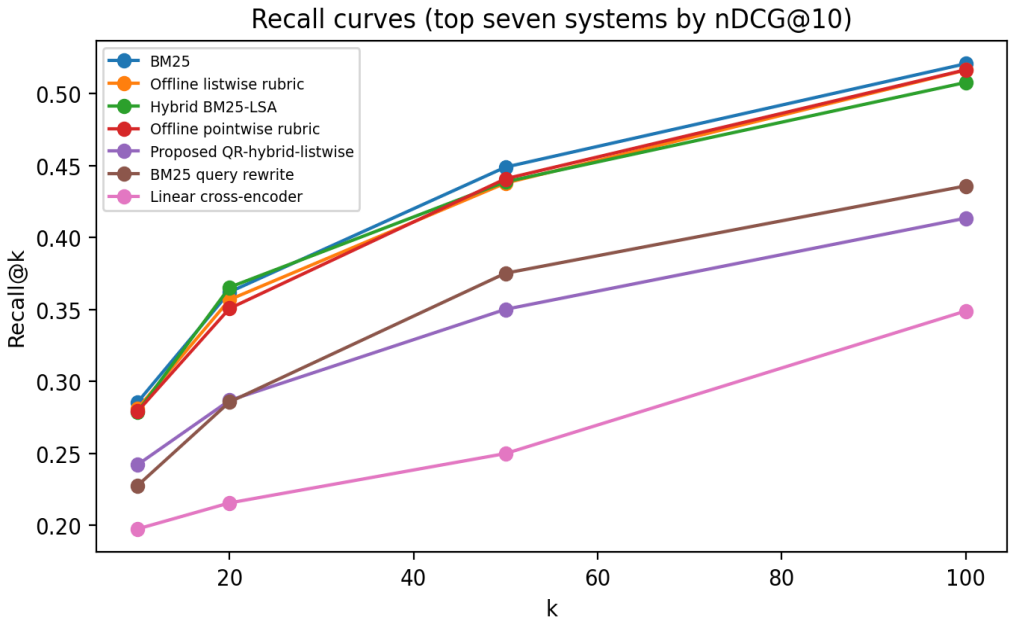

Figure 3. Recall@k curves for the top systems

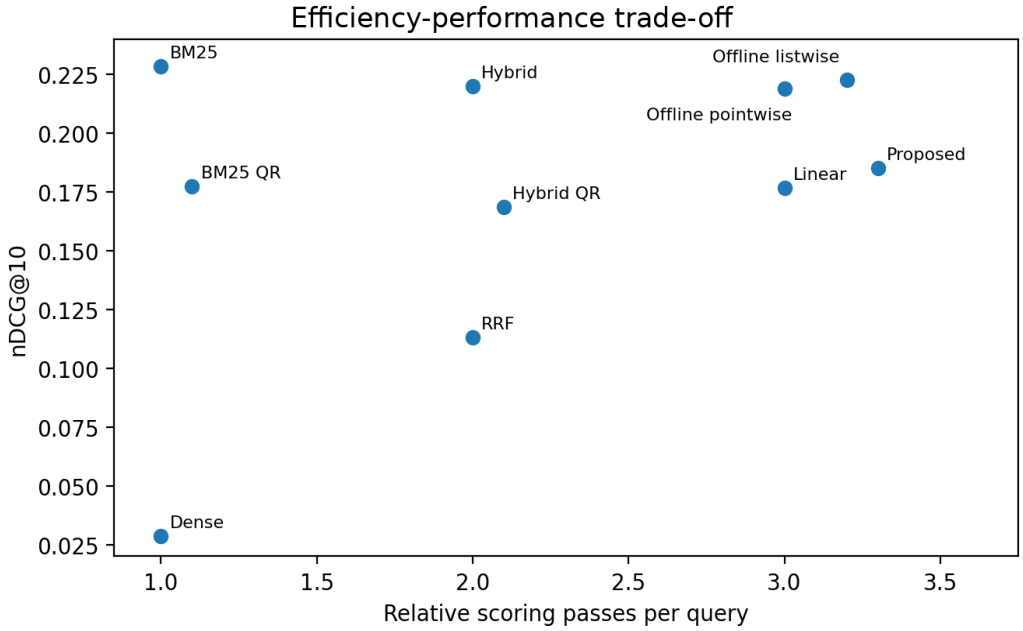

Figure 4. Efficiency-performance trade-off measured by relative scoring passes per query

The query rewriting findings are especially relevant to LLM-based retrieval design. The rewrite module added plausible financial terms, yet the average result was negative. This does not contradict the query rewriting literature; it identifies when rewriting is risky. FiQA questions often

Journal of Technology Informatics and Engineering (JTIE)
Vol. 5 No. 1 April 2026
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 361-378
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

already contain enough context, and adding generic words can shift the ranking toward broad financial discussions rather than the specific answer passage. A better LLM rewrite policy would preserve the original query as a required clause, generate multiple rewrites, retrieve from both original and rewritten queries, and use a validation stage to drop rewrites that reduce lexical specificity.

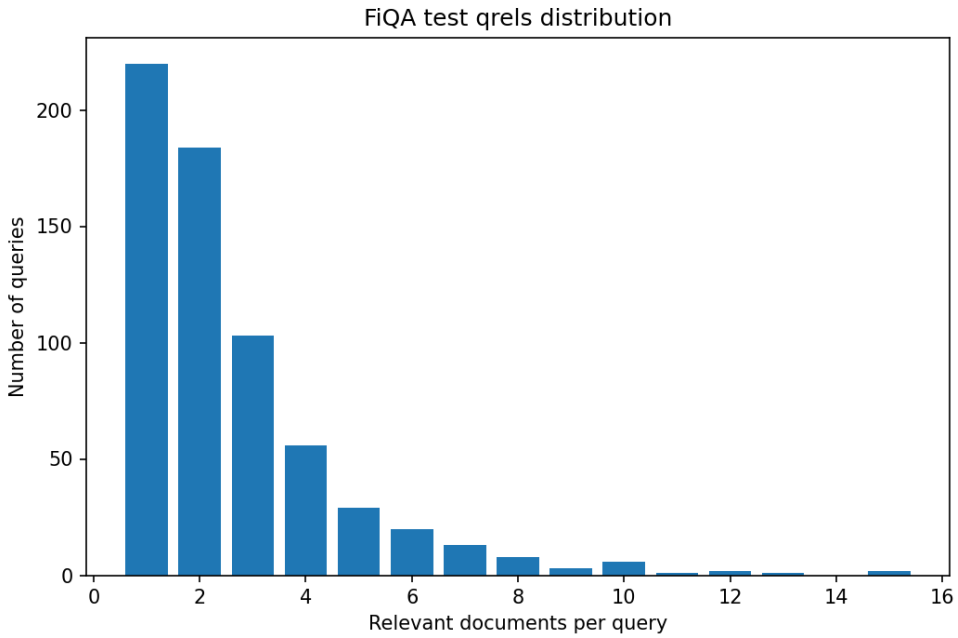

Figure 5. Distribution of relevant documents per FiQA test query

The case-study table supports this interpretation: the largest improvements occur on underspecified or acronym-heavy queries, while the largest degradations occur when the original query already has a narrow financial intent. The query-rewriting examples in Table 9 help explain the negative average result. The module often added reasonable financial vocabulary, but reasonable expansions were not always retrieval-effective under FiQA's judged relevance labels. In several questions, the original wording already contained the discriminative terms needed by BM25, and the added terms shifted scoring toward documents with topical vocabulary rather than direct answers. This behavior is consistent with the quantitative decline in Table 6 and with the distribution shown in Figure 6.

It also explains why the proposed full pipeline did not outperform the strongest baseline despite combining rewriting, hybrid retrieval, and listwise reranking. The efficiency-performance comparison therefore has a concrete operational implication. The listwise rubric required additional scoring passes and produced the best reranker-only value, yet BM25 remained the most effective and simplest overall configuration in this run. A production financial search system

should not assume that an LLM-style reranking layer automatically improves retrieval quality. Instead, it should evaluate candidate recall and rank changes against financial qrels before adding reranking to the pipeline.

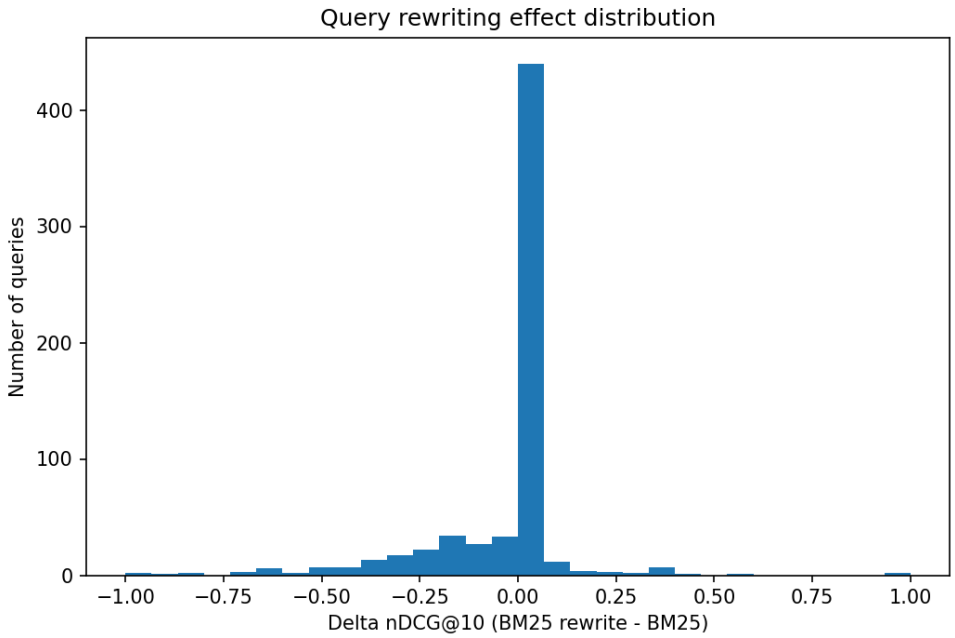

Figure 6. Distribution of query rewriting changes in BM25 nDCG@10

This is especially important when query rewriting is applied before retrieval, because rewriting can change both recall and precision at the same time. The detailed tables also explain why the final interpretation is cautious. The proposed configuration combined three reasonable ideas, but the full pipeline underperformed because query rewriting reduced candidate quality before listwise reranking was applied. The listwise rubric itself was competitive among rerankers, which indicates that the ranking formula was not the sole problem. The experiment therefore supports a narrower claim: listwise ordering can be useful over a strong candidate set, but query rewriting and dense fusion need to be validated on the target financial test set before they are added to a production-style retrieval pipeline.

Limitations

Several limitations should be noted. The rubric rerankers in this study are fixed local scoring rules inspired by LLM-style relevance prompts; they do not represent the behavior of a hosted or open-weight generative LLM reranker. Dense LSA is also a conservative non-neural dense baseline and should not be treated as a proxy for modern neural dense retrievers or domain-adapted financial embedding models. Finally, the nDCG@10 gap between BM25 (0.2285) and the listwise

rubric (0.2228) is small. Without a paired significance test, the safer conclusion is that the proposed pipeline did not demonstrate an improvement over BM25.

V. CONCLUSION AND RECOMMENDATION

This paper evaluated a financial search ranking pipeline on FiQA using the complete 57,638-document corpus and the 648-query BEIR FiQA test qrels. BM25 was the strongest method in this local CPU setup, while the proposed query rewriting plus hybrid retrieval plus listwise reranking pipeline did not outperform it. Hybrid retrieval, lightweight dense retrieval, query rewriting, and reranking did not automatically improve final relevance. The listwise rubric was competitive and ranked second on nDCG@10, but its gain over the pointwise rubric was modest and it did not exceed the lexical baseline.

The results lead to three recommendations. First, financial retrieval studies should always include a strong BM25 baseline and should report recall as well as top-rank precision. In FiQA, exact terms and financial acronyms are often decisive. Second, query rewriting should be gated rather than applied aggressively.

The rewrite module should detect when the original query is underspecified, preserve original terms, and reject expansions that create query drift. Third, LLM reranking should be evaluated under explicit latency and cost constraints. Listwise reasoning is promising, but full prompt-based reranking can be expensive; fixed rubrics, distillation, or small open-source rerankers can be useful stepping stones before hosted LLM deployment. Future work should run the same experimental setup with stronger neural encoders such as E5-small, MiniLM sentence transformers, or domain-adapted financial embeddings, and then compare those results with actual LLM pointwise, pairwise, and listwise prompts.

A second extension should train rewrite gating on development queries instead of using a fixed lexicon. A third extension should add calibration and statistical significance testing, because small nDCG differences among BM25, hybrid retrieval, and listwise reranking may not justify added system complexity. For the present dataset and implementation, the conservative conclusion is clear: robust lexical search is the safest default for FiQA, while listwise reranking is a useful candidate for carefully constrained optimization. Overall, the FiQA experiment supports a conservative deployment strategy: start with robust lexical retrieval, add dense retrieval or query rewriting only when development results justify them, and evaluate any LLM-based reranker against the same candidate set under realistic latency and cost constraints. On the FiQA experiment reported here, the simplest lexical system remained the best measured choice, providing a useful baseline for later improvements.

REFERENCES

Amati, G., & van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems,* 20(4), 357–389. https://doi.org/10.1145/582415.582416

Binghua Zhou, Siming Zhao, & David Chao. (2023). LLM-Guided Energy-Aware A/B Testing for Consolidation and DVFS Policies via Power-Sensitivity Clustering. *Journal of Advanced Computing Systems* , 3(4), 12-30. https://doi.org/10.69987/JACS.2023.30402

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *In Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901).

Cormack, G. V., Clarke, C. L. A., & Buettcher, S. (2009). Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. *In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 758–759). https://doi.org/10.1145/1571941.1572114

Daren Zheng, & Chenyu Li. (2024). Behavior-Level Jailbreak Resistance via Multi-Stage Refusal + Utility Preservation. *Journal of Advanced Computing Systems* , 4(1), 83-99. https://doi.org/10.69987/JACS.2024.40107

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *In Proceedings of NAACL-HLT* 2019 (pp. 4171–4186). https://doi.org/10.18653/v1/N19-1423

Gao, L., Ma, X., Lin, J., & Callan, J. (2023). Precise zero-shot dense retrieval without relevance labels. *In Proceedings of ACL* 2023 (pp. 1762–1777). https://doi.org/10.18653/v1/2023.acl-long.99

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems,* 20(4), 422–446. https://doi.org/10.1145/582415.582418

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. *In Proceedings of EMNLP* 2020 (pp. 6769–6781). https://doi.org/10.18653/v1/2020.emnlp-main.550

Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. *In Proceedings of SIGIR 2020* (pp. 39–48). https://doi.org/10.1145/3397271.3401075

Kuo, M.-J., Zheng, D., & Hires, J. (2025). Federated topic-preference learning for knowledge-grounded chat with differential privacy. *Journal of Technology Informatics and Engineering*, 4(2). https://doi.org/10.51903/jtie.v4i2.502

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *In Advances in Neural Information Processing Systems* (Vol. 33, pp. 9459–9474).

Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M., & Balahur, A. (2018). WWW'18 open challenge: Financial opinion mining and question answering. *In Companion Proceedings of the Web Conference 2018* (pp. 1941–1942). https://doi.org/10.1145/3184558.3192301

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. *Cambridge University Press*.

Muennighoff, N., Tazi, N., Magne, L., & Reimers, N. (2023). MTEB: Massive text embedding benchmark. *In Proceedings of EACL 2023* (pp. 2014–2037). https://doi.org/10.18653/v1/2023.eacl-main.148

Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. *arXiv*. https://arxiv.org/abs/1901.04085

Nogueira, R., Jiang, Z., & Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. *In Findings of EMNLP 2020* (pp. 708–718). https://doi.org/10.18653/v1/2020.findings-emnlp.63

Pradeep, R., Sharifymoghaddam, S., & Lin, J. (2023a). RankVicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv*. https://arxiv.org/abs/2309.15088

Pradeep, R., Sharifymoghaddam, S., & Lin, J. (2023b). RankZephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv.* https://arxiv.org/abs/2312.02724

Qin, Z., Jagerman, R., Hui, K., Zhuang, H., Wu, J., Yan, L., Shen, J., Liu, T., Liu, J., Metzler, D., Wang, X., & Bendersky, M. (2023). Large language models are effective text rankers with pairwise ranking prompting. *arXiv*. https://arxiv.org/abs/2306.17563

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *In Proceedings of EMNLP-IJCNLP 2019* (pp. 3982–3992). https://doi.org/10.18653/v1/D19-1410

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC-3) (pp. 109–126). *National Institute of Standards and Technology.*

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389. https://doi.org/10.1561/1500000019

Sun, W., Yan, L., Ma, X., Wang, S., Ren, P., Chen, Z., Yin, D., & Ren, Z. (2023). Is ChatGPT good at search? Investigating large language models as re-ranking agents. *In Proceedings of EMNLP 2023* (pp. 14918–14937). https://doi.org/10.18653/v1/2023.emnlp-main.923

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *In Advances in Neural Information Processing Systems* (Vol. 34, pp. 7981–7997).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *In Advances in Neural Information Processing Systems* (Vol. 30).

Voorhees, E. M. (2002). The philosophy of information retrieval evaluation. In Revised Papers from CLEF 2001 (pp. 355–370). *Springer*. https://doi.org/10.1007/3-540-45691-0_34

Wang, L., Yang, N., Huang, X., Jiao, B., Jiang, D., Majumder, R., & Wei, F. (2022). Text embeddings by weakly-supervised contrastive pre-training. *arXiv*. https://arxiv.org/abs/2212.03533

Xinzhuo Sun, Jing Chen, Binghua Zhou, & Meng-Ju Kuo. (2024). ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source Conflicting Evidence. *Journal of Advanced Computing Systems* , 4(7), 50-64. https://doi.org/10.69987/JACS.2024.40705