

Evaluating Retrieval Accuracy and Latency for Enterprise RAG Systems on FiQA

Aleksandar Mihajlovic

amihajlovic1990@gmail.com

Singidunum University <https://orcid.org/0009-0009-0623-8080>

Research Article

Keywords: retrieval-augmented generation, information retrieval, FiQA, BM25, dense retrieval, reciprocal rank fusion, latency, reproducibility

Posted Date: October 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11452876/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare potential competing interests as follows: The author is employed as an Associate Director - Forward Deployed Engineer at Modon Holding in Abu Dhabi, United Arab Emirates. This employment is disclosed as a potential competing interest. The views expressed in this article are the author's own.

Evaluating Retrieval Accuracy and Latency for Enterprise RAG Systems on FiQA

Aleksandar Mihajlovic

Technical Faculty, Singidunum University

Belgrade, Serbia

Abstract

Retrieval design influences which evidence a retrieval-augmented generation (RAG) system can access and how long that evidence takes to obtain. This study compares TF-IDF, BM25, a dynamically quantized MiniLM dense retriever, and reciprocal rank fusion (RRF) of BM25 and MiniLM on the complete BEIR FiQA test split. All 648 test questions were evaluated against 57,638 documents using the supplied relevance judgments. Outcomes were nDCG@10, the proportion of questions with a relevant document in the first five results (Hit@5), recall, reciprocal rank, and local CPU retrieval latency. Accuracy uncertainty was estimated with 10,000 paired query bootstrap samples; latency was measured over three randomized passes. Mean nDCG@10 was 0.179 for TF-IDF, 0.235 for BM25, 0.334 for MiniLM, and 0.337 for RRF. RRF minus MiniLM was +0.003 (95% confidence interval -0.015 to +0.022). Hit@5 ranged from 30.2% to 54.0%; median retrieval latency ranged from 5.74 to 23.92 ms. The results quantify implementation-specific effectiveness and latency trade-offs and show why fusion should be evaluated against its strongest constituent. FiQA is a public financial forum benchmark, and the evaluated model may have encountered overlapping training material. These findings concern retrieval relevance under the supplied judgments, not generated-answer accuracy or production enterprise performance.

Keywords: retrieval-augmented generation; information retrieval; FiQA; BM25; dense retrieval; reciprocal rank fusion; latency; reproducibility

Introduction and related research

Retrieval-augmented generation combines a language model with information retrieved from an external collection [1]. In an enterprise application, the retriever determines which records, policies, or technical passages become available to the generator. A relevant passage that arrives too late can make a system impractical, while a fast search that misses the needed evidence leaves the generator without an adequate basis for its answer. Retrieval quality and retrieval latency therefore warrant joint measurement before introducing the additional variability of answer generation.

The main design choice considered here is how to rank candidate documents. TF-IDF represents a document through weighted vocabulary terms [2]. BM25 combines inverse document frequency with term-frequency saturation and document-length normalization [3]. Sentence embedding models encode queries and documents into vectors whose similarity can capture relationships beyond exact vocabulary overlap [4,5]. Reciprocal rank fusion combines ranked lists without requiring their underlying scores to have comparable scales [6]. These approaches are established methods; their relative performance depends on the collection, model, preprocessing, and implementation.

BEIR made this dependence explicit by evaluating retrieval methods across heterogeneous tasks and domains [7]. Its results support retaining lexical baselines when assessing neural retrieval, rather than assuming that a model architecture guarantees better relevance. FiQA provides an openly available financial question-answering setting in which questions must be matched to relevant answer posts [8]. The BEIR adaptation

supplies a fixed document collection and test relevance judgments [9], making it suitable for a reproducible comparison without access to confidential enterprise data.

Retrieval relevance is only one dimension of RAG evaluation. RAGAs distinguishes retrieval quality from the faithfulness and quality of generated responses [10]. Research on long-context language models also shows that the presence of relevant information does not ensure that a model will use it reliably [11]. Accordingly, this report uses accuracy to mean success in retrieving documents judged relevant to a question. It does not equate a retrieval hit with a correct, complete, current, or policy-compliant generated answer.

The study asks three questions: how the four configurations compare on early-ranked relevance; how much local CPU latency each configuration incurs when query encoding is included; and whether lexical and dense retrieval identify complementary relevant evidence that rank fusion preserves. The contribution is an auditable comparison of four explicit configurations on the full FiQA test split, including paired uncertainty estimates, repeated latency measurements, and a query-level complementarity analysis. Enterprise RAG is the motivating application; FiQA remains a financial forum proxy rather than an enterprise deployment sample.

Experimental methodology

Dataset and evaluation unit

The experiment used the FiQA dataset distributed by BEIR [8,9]. The downloaded archive contained 57,638 distinct document identifiers, 6,648 query records, and relevance files for 5,500 training questions, 500 development questions, and 648 test questions. Evaluation included every test question and all 57,638 corpus documents. The test file contained 1,706 positive query-document judgments, or an average of 2.63 judged relevant documents per question. The corpus contained 39 duplicate text records beyond the first occurrence; their distinct document identifiers were retained to preserve the benchmark. The corpus and supplied relevance labels were not rewritten, manually augmented, or filtered by model performance.

Each original answer post was treated as a single retrieval unit. Title and body were joined with one space and outer whitespace was stripped; the title field is empty in this dataset. No overlapping passage windows, metadata filters, query rewriting, reranker, access-control filtering, or answer generator were introduced. Relevance was binary, with a supplied positive judgment treated as relevant and all other retrieved documents scored as nonrelevant for evaluation. This convention can penalize genuinely useful documents absent from the judgment set, so the metrics quantify agreement with the available labels rather than exhaustive semantic relevance.

No model was fine-tuned and no hyperparameter was selected using FiQA training, development, or test relevance outcomes. The retrieval settings were fixed before test scores were examined. Model-runtime and offline batching choices were made for feasibility on the available CPU, without using relevance labels. The pretrained model card lists Stack Exchange material among its training sources [12], while FiQA uses financial forum content [8]. Possible overlap was not audited. The evaluation is therefore a fixed test-split comparison without additional task training, not a claim of contamination-free zero-shot generalization.

Retrieval configurations

Both lexical methods used the same scikit-learn CountVectorizer vocabulary. Text was lowercased and split into Unicode word tokens of at least two characters, using the regular expression provided in the supplementary code. No stop-word removal or stemming was applied. Vocabulary and document-frequency statistics were computed from the corpus alone. Single-character terms were omitted by this tokenizer. These choices define a simple reproducible lexical baseline and should not be interpreted as an optimized production search configuration.

TF-IDF used sublinear term frequency: a positive count was transformed to one plus its natural logarithm. Smoothed inverse document frequency was the natural logarithm of the ratio of one plus the document count to one plus document frequency, followed by adding one. Document and query vectors were normalized to unit Euclidean length and scored by their dot product. BM25 used k_1 equal to 1.2 and b equal to 0.75, with positive inverse document frequency computed as the natural logarithm of one plus the ratio of N minus document frequency plus 0.5 to document frequency plus 0.5. Each distinct query term contributed once. BM25 weights were precomputed in a sparse column index and accumulated over query-term postings.

Dense retrieval used sentence-transformers/all-MiniLM-L6-v2 [5,12], pinned to revision 1110a243fdf4706b3f48f1d95db1a4f5529b4d41. The model has six transformer layers and produces 384-dimensional sentence representations. The original safetensors weights were loaded locally, then PyTorch dynamic qint8 quantization was applied to linear layers. Embeddings, attention computations outside those quantized layers, pooled vectors, and the search matrix remained in floating-point form. The model was in evaluation mode with gradients disabled. This configuration is called MiniLM int8 throughout the results; its equivalence to an unquantized deployment was not tested.

The provided tokenizer truncated each query and document to a maximum sequence length of 256 tokens, including special tokens. Attention-mask-weighted mean pooling and unit-length normalization produced one vector per document. Corpus encoding used batches of 32, sorted by ascending character length with original corpus order breaking ties, to reduce padding. Original document identifiers and their positions in the search matrix were preserved. Each online query was encoded individually. Dense scores were exact dot products over the entire float32 matrix; no approximate nearest-neighbor index was used.

The hybrid configuration independently retrieved the top 100 BM25 and top 100 dense candidates for each query. Each candidate received the sum of the inverse of 60 plus its one-based rank in each list in which it appeared [6]. An absent document contributed zero from that list. The merged scores determined a new top-100 ranking. All score ties were resolved by original corpus order. BM25 and dense search ran sequentially within each hybrid request, so its measured latency includes both retrieval branches and fusion. The hybrid did not reuse outputs from separately timed baseline requests.

Outcome measures and uncertainty

The primary relevance outcome was mean normalized discounted cumulative gain at rank 10 (nDCG@10). A relevant document contributed a gain discounted by the inverse base-two logarithm of one plus its rank; this total was divided by the best possible gain for that question’s number of judged relevant documents. Secondary measures were Hit@5, mean reciprocal rank at 10 (MRR@10), and recall at 1, 5, 10, and 100. Hit@5 is the fraction of questions with at least one judged relevant document among the first five results. Recall divides the number of retrieved judged relevant documents by all judged relevant documents for that question. MRR@10 uses the inverse rank of the first relevant result within the first ten, or zero if none is present. All relevance outcomes were macro-averaged across questions.

Uncertainty was estimated by resampling the 648 questions with replacement 10,000 times, using seed 20261021. The same sampled question indices were used across methods to form paired percentile intervals for mean differences. The 2.5th and 97.5th percentiles are reported as 95% confidence intervals. These intervals describe variation under query resampling; they do not include model-training, corpus-sampling, or infrastructure uncertainty. Four planned contrasts were retained in the supplementary results: MiniLM minus BM25, RRF minus BM25, RRF minus MiniLM, and BM25 minus TF-IDF. The analysis is descriptive, without multiplicity-adjusted hypothesis tests or claims of universal superiority.

Two additional RRF constants, 10 and 100, were evaluated using the recorded constituent rankings as an exploratory sensitivity analysis. The primary configuration remained 60 regardless of the outcome. These

alternative fusion settings were not separately timed. Complementarity was described by counting questions for which BM25 and MiniLM individually did or did not place a judged relevant document in the top five. The union of their successes is an oracle coverage diagnostic over up to ten candidates; it is not an achievable five-document ranking without an additional selection rule.

Latency protocol and computing environment

The benchmark ran in a Linux virtualized environment exposing an AMD EPYC 9V74 processor, with a CPU quota equivalent to eight cores and an 8 GiB memory limit. Online inference used one PyTorch thread, one inter-op thread, and one BLAS thread; corpus embedding used eight PyTorch threads. No GPU or remote embedding API was used. The recorded software versions were Python 3.12.14, NumPy 2.3.5, SciPy 1.17.0, scikit-learn 1.8.0, PyTorch 2.14.1+cpu, Transformers 5.20.2, and Tokenizers 0.23.3. The full environment and file hashes are included in Supplementary File S1.

All indexes were resident in memory before timing. Each configuration was warmed up with the first 20 test queries in numerical query-ID order. Three complete passes then processed all 648 queries in seeded randomized order; the order of the four methods was independently randomized within each query. There were 1,944 latency observations per configuration. A monotonic high-resolution clock measured elapsed time from the query string to the returned ranking indices, including tokenization or query encoding, scoring, stable full sorting, and hybrid fusion as applicable. Final conversion of indices to document identifiers, document-body materialization, disk reads, network transit, queueing, authentication, and text generation were excluded.

Median and 95th-percentile latency were computed from the pooled request measurements and are descriptive of this host and implementation. They are not throughput estimates or production service-level guarantees. Repeated passes were used for timing only; identical queries did not become additional independent accuracy observations. All three passes produced identical rankings within each configuration. Full-array sorting and exact dense matrix search were retained for deterministic comparison, so the benchmark does not compare optimized search engines or approximate indexes.

Results

Retrieval accuracy and quality

Table 1 reports retrieval outcomes for the 648 test questions. BM25 + MiniLM RRF had the highest mean nDCG@10 in this run (0.337). MiniLM minus BM25 was 0.099 (0.075 to 0.124); RRF minus BM25 was 0.102 (0.086 to 0.119). The comparison of RRF with MiniLM was 0.003 (-0.015 to 0.022). The paired interval included zero, so this experiment did not establish an nDCG advantage for either configuration. Parentheses in these comparisons contain paired 95% confidence intervals, in absolute metric units.

Table 1 Retrieval effectiveness on the complete FiQA test split

Method	nDCG@10 95% CI	Hit@5	Recall@10	MRR@10	Recall@100
TF-IDF	0.179 (0.158 to 0.200)	30.2%	0.244	0.216	0.491
BM25	0.235 (0.211 to 0.258)	39.8%	0.296	0.293	0.513
MiniLM int8	0.334 (0.308 to 0.361)	54.0%	0.406	0.400	0.679
BM25 + MiniLM RRF	0.337 (0.311 to 0.363)	53.1%	0.415	0.411	0.681

BM25 retrieved at least one judged relevant document in the top five for 258 questions, MiniLM for 350, and RRF for 344. BM25 and MiniLM both succeeded on 204 questions; 54 were BM25-only successes and 146 were dense-only successes. Both missed on 244 questions. The union of their separate top-five successes covered 404 questions (62.3%), compared with 53.1% for the five fused results. This difference identifies a selection problem: complementary candidates can exist without being retained in the same short ranked list.

RRF $nDCG@10$ was 0.349 with constant 10, 0.337 with the primary constant 60, and 0.335 with constant 100. Corresponding $Hit@5$ values were 56.9%, 53.1%, and 52.6%. These descriptive results show sensitivity to how much the fusion rule favors the top of each constituent list; they were not used to replace the primary setting.

Retrieval latency

Table 2 summarizes the observed retrieval times. BM25 had the lowest median latency (5.74 ms). MiniLM required 17.90 ms at the median and 21.12 ms at the 95th percentile. RRF required 23.92 and 28.03 ms respectively. Its median was 4.17 times the BM25 median. These measurements include individual query encoding for dense retrieval and both sequential branches for RRF. Figure 1 relates the measured median time to $nDCG@10$.

Table 2 Local retrieval latency in milliseconds

Method	Median	95th percentile	Mean	Requests
TF-IDF	6.93	8.40	6.64	1,944
BM25	5.74	7.25	5.41	1,944
MiniLM int8	17.90	21.12	18.27	1,944
BM25 + MiniLM RRF	23.92	28.03	23.98	1,944

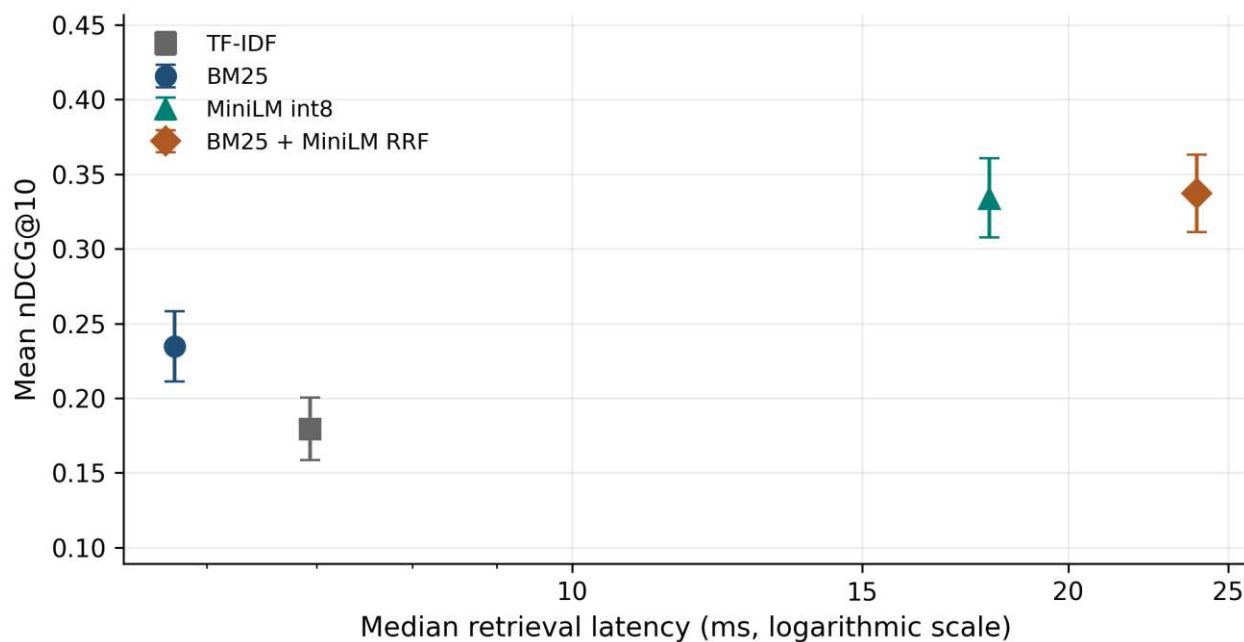


Figure 1 Retrieval effectiveness and local latency. Vertical bars show 95% query-bootstrap confidence intervals for $nDCG@10$. The latency axis is logarithmic. Horizontal timing uncertainty is not estimated. All points refer to the same corpus, test queries, host, and online thread settings.

The float32 document matrix contained 57,638 by 384 values and occupied 84.4 MiB before file headers. Offline corpus encoding took 406.8 seconds under the separate eight-thread build configuration. That preparation cost is excluded from online latency. 11,040 documents (19.2%) exceeded the dense encoder’s 256-token limit before truncation. The median untruncated token count was 120 and the 95th percentile was 481. Consequently, the lexical methods could use text beyond the encoder’s input window.

Discussion and limitations

MiniLM improved mean nDCG@10 over BM25 by 0.099, while its median retrieval time rose from 5.74 to 17.90 ms. Adding RRF increased the MiniLM median by 6.02 ms (33.6%). Its primary nDCG point estimate improved by only 0.003, and the paired interval included zero. Thus, this run supports a measurable dense-versus-lexical trade-off but does not establish that the primary fusion setting improves on dense retrieval alone. The interval is also too wide to establish equivalence. RRF’s higher nDCG point estimate and lower Hit@5 illustrate why choosing a single accuracy label can obscure differences between ranking quality and evidence coverage.

The findings support treating retrieval selection as an empirical design decision. Lexical ranking provides a useful reference because it exploits exact terminology with limited per-query computation. A pretrained dense representation can connect differently worded questions and documents, but its observed performance also reflects its training data, sequence limit, and runtime precision. Rank fusion adds a candidate-combination mechanism rather than a guarantee of improved early-ranked relevance. Its value depends on both the strength and the complementarity of the constituent rankings.

The 54 BM25-only and 146 dense-only top-five successes provide direct evidence of complementary labeled coverage on this dataset. However, the union count cannot be compared to RRF as if both methods supplied the same number of documents. The union can contain ten documents, while Hit@5 evaluates five. A follow-up experiment could compare reranking or learned fusion under the same final context budget, using a development split to choose the fusion rule and a new held-out evaluation to assess it.

The latency findings apply to in-memory local retrieval. A production RAG request also incurs context construction, permissions checks, network and queue delays, and generation time. The relative importance of the measured retrieval differences depends on those additional stages. Running lexical and dense searches concurrently could change the hybrid’s latency, but parallel hybrid serving was not tested. Likewise, an approximate dense index or a production lexical engine could change both relevance and timing. The reported times therefore characterize these implementations rather than the theoretical speed of a retrieval family.

Several limits constrain generalization. First, FiQA consists of financial forum answer posts, not internal enterprise knowledge with access policies, document versions, regulated workflows, or multilingual terminology. Public labels do not establish that a retrieved answer is current, factually correct, or suitable for professional financial decisions. A result on FiQA should inform the design of a representative enterprise test set, not replace that test set.

Second, the experiment measures retrieval relevance and does not instantiate the generation stage. A relevant document can be ignored, misread, or combined incorrectly by an LLM. Conversely, a model may answer from its parameters even when judged relevant evidence was missed. Higher Hit@5 or nDCG@10 cannot therefore be translated into an answer-accuracy improvement. A subsequent end-to-end study should hold the generator, prompt, context budget, and decoding settings fixed, then separately assess answer correctness, evidence support, citation quality, and total response latency [10,11].

Third, only one compact English embedding model and one quantization configuration were tested. No cross-encoder, late-interaction retriever, learned sparse retriever, or recent large embedding model was evaluated.

The model’s possible exposure to overlapping Stack Exchange content prevents a clean unseen-domain claim [12]. The 256-token dense window and unrestricted lexical input also mean this is a comparison of practical configurations rather than an isolation of representation type alone. Chunking, longer input windows, and a full-precision control are material extensions.

Fourth, the available judgments may be incomplete. Unjudged documents were treated as nonrelevant, and no manual relevance audit or user study was conducted. The percentile bootstrap assumes that test questions are exchangeable; related questions could weaken that assumption. No confidence intervals for latency quantiles, independent-host replications, cold-start measurements, or concurrent-load measurements were obtained. The benchmark was not preregistered. These limitations favor interpreting the effect sizes and raw query-level results over making general claims about statistical or operational superiority.

For an enterprise pilot, the useful next step is to preserve the same separation of questions: whether a retriever supplies adequate evidence, whether the generator uses that evidence correctly, and whether the complete service meets its response-time requirement. The supplied scripts make the first question reproducible on public data. Applying them to a permission-appropriate internal benchmark, followed by a controlled answer-generation evaluation, would test whether the observed trade-offs persist in the intended workflow.

Conclusion

This study evaluated four retrieval configurations on all 648 FiQA test questions and 57,638 documents. BM25 + MiniLM RRF achieved the highest observed nDCG@10, while BM25 had the lowest median local retrieval latency. The paired RRF-minus-MiniLM nDCG difference was +0.003, with a 95% interval from -0.015 to +0.022. Complementarity analysis showed that combining candidate sources and preserving their useful evidence in a small final ranking are distinct problems. The report supplies measured outcomes and reproduction materials for choosing a retrieval baseline. Validation on internal enterprise questions and generated answers is required before drawing conclusions about an operational enterprise RAG system.

Declarations

Data and code availability

FiQA is publicly available through the BEIR dataset distribution [9]. The retrieval implementation, pinned input hashes, dependency versions, reproduction instructions, per-query metrics, all top-100 rankings, raw timing observations, confidence-interval summaries, and validation checks accompany this report in Supplementary File S1. The download script retrieves the corpus and pretrained model from their original distributors. Corpus text and model weights are not redistributed in the supplementary archive. The report introduces no new manually labeled dataset.

Ethics and consent

This computational study analyzed an existing public benchmark. It recruited no participants, conducted no intervention, and used no private enterprise records. No new participant consent was collected. The report makes no claim of institutional ethics approval or exemption.

Use of artificial intelligence

OpenAI ChatGPT with Codex was used to develop the experimental implementation, execute the local computation, assist literature discovery, analyze outputs, and draft the manuscript and figures. Numerical results were produced by the supplied deterministic retrieval and evaluation code using the downloaded benchmark, not generated as illustrative values by a language model. Figures were plotted from the recorded

results. Aleksandar Mihajlovic approved the final manuscript and accepts responsibility for its content, including the implementation, analysis, interpretation, and references. AI tools are not listed as authors.

Funding

This study was funded personally by the author. No external funding or grant support was received.

Competing interests

The author is employed as an Associate Director - Forward Deployed Engineer at Modon Holding in Abu Dhabi, United Arab Emirates. This employment is disclosed as a potential competing interest. The views expressed in this article are the author's own.

Author contributions

Aleksandar Mihajlovic conceived the study topic and comparative research question, researched industry needs and the relevance of retrieval strategies to enterprise RAG systems, and defined the study's academic and practical motivation. The work was undertaken as part of his PhD studies at the Technical Faculty, Singidunum University, Belgrade, Serbia, with the aim of contributing reproducible evidence to the AI research community and enterprise RAG practice. He funded the study personally and approved the final manuscript. Computational implementation, execution, analysis support, and initial manuscript preparation used the AI assistance disclosed above.

References

- [1] Lewis P, Perez E, Piktus A, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*. 2020;33:9459-9474. <https://arxiv.org/abs/2005.11401>
- [2] Manning CD, Raghavan P, Schütze H. *Introduction to Information Retrieval*. Cambridge University Press; 2008. Chapter 6. <https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html>
- [3] Robertson S, Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*. 2009;3(4):333-389. <https://doi.org/10.1561/15000000019>
- [4] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of EMNLP-IJCNLP*. 2019:3982-3992. <https://doi.org/10.18653/v1/D19-1410>
- [5] Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. 2020. <https://arxiv.org/abs/2002.10957>
- [6] Cormack GV, Clarke CLA, Büttcher S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of SIGIR*. 2009:758-759. <https://doi.org/10.1145/1571941.1572114>
- [7] Thakur N, Reimers N, Rücklé A, Srivastava A, Gurevych I. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *NeurIPS Datasets and Benchmarks*. 2021. <https://arxiv.org/abs/2104.08663>
- [8] Maia M, Handschuh S, Freitas A, Davis B, McDermott R, Zarrouk M, Balahur A. WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. *Companion Proceedings of The Web Conference*. 2018:1941-1942. <https://doi.org/10.1145/3184558.3192301>
- [9] BEIR contributors. FiQA dataset distribution and dataset catalog. <https://github.com/beir-cellar/beir> and <https://public.ukp.informatik.tu-darmstadt.de/thakur/BEIR/datasets/fiqa.zip>. Accessed 21 October 2026. Dataset archive SHA-256: 32c7df99ed21252fdb2cf3f5673502a8d245ee0c44c4a133570d92ce2b3ad02.
- [10] Es S, James J, Espinosa Anke L, Schockaert S. RAGAs: Automated Evaluation of Retrieval Augmented Generation. *Proceedings of EACL System Demonstrations*. 2024:150-158. <https://doi.org/10.18653/v1/2024.eacl-demo.16>

- [11] Liu NF, Lin K, Hewitt J, Paranjape A, Bevilacqua M, Petroni F, Liang P. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*. 2024;12:157-173. https://doi.org/10.1162/tacl_a_00638
- [12] Sentence Transformers contributors. all-MiniLM-L6-v2 model card and weights. Hugging Face. Revision 1110a243fdf4706b3f48f1d95db1a4f5529b4d41. <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2/tree/1110a243fdf4706b3f48f1d95db1a4f5529b4d41>. Accessed 21 October 2026.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryS1Reproducibility.zip](#)