Retrieval Enhancements for RAG: Insights from a Deployed Customer Support Chatbot

Daniel González Juclà[1], **Mohit Tuteja**[2], **Marcos Esteve Casademunt**[1], **Keshav Unnikrishnan**[2], **Yasir Usmani**[2], **Arvind Roshaan**[2]

[1]Thomson Reuters Labs, Zug, Switzerland
[2]Thomson Reuters Labs, Bangalore, India

Abstract

Retrieval-Augmented Generation (RAG) systems depend critically on retrieval quality to enable accurate, contextually relevant LLM responses. While LLMs excel at synthesis, their RAG performance is bottlenecked by document relevance. We evaluate advanced retrieval techniques including embedding model comparison, Reciprocal Rank Fusion (RRF), embedding concatenation and list-wise and adaptive LLM-based re-ranking, demonstrating that zero-shot LLMs outperform traditional cross-encoders in identifying high-relevance passages.

We also explore context-aware embeddings, diverse chunking strategies, and model fine-tuning. All methods are rigorously evaluated on a proprietary dataset powering our deployed production chatbot, with validation on three public benchmarks: FiQA, HotpotQA, and SciDocs. Results show consistent gains in Recall@10, closing the gap with Recall@50 and yielding actionable pipeline recommendations. By prioritizing retrieval enhancements, we significantly elevate downstream LLM response quality in real-world, customer-facing applications.

1 Introduction

To enhance our RAG-based system's retrieval performance, we observed that when relevant documents are ranked within the top three results, the LLM generates accurate and comprehensive responses in over 92% of cases. However, the recall@3 for retrieved documents was notably lower, underscoring a critical bottleneck in the retrieval phase. This insight drove our investigation into advanced retrieval strategies to improve overall system performance, with a deliberate emphasis on enhancing recall metrics. We specifically focus on the retrieval component, as LLMs have demonstrated the ability to generate accurate responses when relevant documents are present in their context. Importantly, this research intentionally limits its scope to retrieval enhancements and does not evaluate the full end-to-end RAG pipeline, prioritizing improvements in document relevance to lay a stronger foundation for downstream generation tasks.

2 Related Work

Retrieval-Augmented Generation (RAG) has emerged as a pivotal framework for enhancing large language models (LLMs) by integrating external knowledge sources to improve response accuracy and relevance. The foundational work by (Lewis et al., 2020) introduced RAG, combining parametric and non-parametric memory to effectively tackle knowledge-intensive tasks. A comprehensive survey by Gao et al. (2024) reviews over 100 RAG studies, categorizing them into Naive, Advanced, and Modular RAG paradigms, and provides insights into advancements in retrieval, generation, and augmentation techniques.

The retrieval phase is central to RAG's efficacy. Recent innovations include Hypothetical Document Embeddings (HyDE), introduced in Gao et al. (2023), which enhance zero-shot dense retrieval by generating hypothetical documents that better capture query intent. Reciprocal Rank Fusion (RRF) (Cormack et al., 2009) aggregates rankings from multiple retrievers, proving valuable in hybrid search.

Context-aware embeddings have been investigated to capture nuanced query-document relationships. Anthropic's Contextual Retrieval method (Anthropic, 2024), along with Zhang et al. (2025b), significantly reduce retrieval failures by incorporating chunk-level context and improving precision (Rau et al., 2025). The impact of context length is studied in (Joren et al., 2025), which introduces the concept of sufficient context, and (Li et al., 2024),

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track), pages 169–180

which compares RAG with long-context LLMs and proposes a hybrid approach.

Model fine-tuning using LLMs to generate synthetic samples is explored in Appendix D. A common pipeline, as noted by Rosa et al. (Rosa et al., 2022), retrieves top-k candidates using bi-encoders and re-ranks them with cross-encoders.

LLMs have shown exceptional performance in complex tasks (Brown et al., 2020), prompting exploration of their use in re-ranking (Qin et al., 2024). SlideGar (Rathee et al., 2025a) and related work (Rathee et al., 2025b) demonstrate adaptive retrieval guidance, while (Gangi Reddy et al., 2024) propose FIRST, a listwise re-ranking method using output logits. LLMs also excel at needle-in-the-haystack tasks (Team et al., 2024).

To our knowledge, no prior study evaluates LLMs as direct re-rankers over the top 50 candidates from state-of-the-art embedding models. We address this gap by focusing exclusively on retrieval enhancement to maximize downstream LLM performance.

3 Datasets

We tested all the approaches on four datasets. Three of the datasets come from the BeIR benchmark (Thakur et al., 2021b), and we curated a proprietary internal dataset for our downstream use case. Below is a brief description of each dataset:

FiQA-2018: The Financial Question Answering dataset (FiQA-2018) focuses on question-answering in the financial domain. It contains 14,166 query-document pairs, with 648 queries and a corpus of 57,638 documents. Queries are financial questions, and documents are relevant passages or answers, often sourced from financial texts. The dataset uses binary relevance judgments, with an average of 2.6 documents per query. FiQA is designed to evaluate retrieval models' ability to handle domain-specific queries.

HotpotQA: HotpotQA is a question-answering dataset emphasizing multi-hop reasoning. Queries require reasoning over multiple documents (specifically 2) to provide answers, supported by sentence-level facts for explainability. The corpus is Wikipedia-based containing 5,233,329 documents. We however have constructed a smaller corpus from the Dev set (distractor) setting with 66,581 documents, to keep the size of corpus in the same range as the other datasets. We report the performance of our experiments on the 7405 questions from the Dev set (distractor) setting. This setting tests the models' capabilities in retrieving both the relevant documents needed for multi-hop reasoning.

SciDocs: SciDocs is a citation prediction dataset in the scientific domain, comprising 1,000 queries and a corpus of 25,657 documents. It focuses on retrieving documents relevant to scientific queries, with binary relevance judgments and an average of 4.9 documents per query. SciDocs evaluates models' performance in retrieving precise, domain-specific scientific information, making it suitable for testing retrieval in academic contexts.

Help Articles: Our product assistant chat-bot answers questions related to the company's product usage, tax & finance related queries in general. This content comes from a lot of help and support articles available on publicly accessible company web-pages/ PDFs. We extracted text from 15,848 such web-pages and some PDF articles. PDF text chunking was done using LLMSherpa as it's layout-aware chunking helps preserve structural coherence (e.g., sections, tables). These source documents, particularly the PDFs have higher average token count than all the BeIR datasets hence chunking is needed for models with smaller context windows. We also collected Subject Matter Expert (SME) feedback on 310 user queries and the model's responses. This is the same dataset used to build and deploy our production chatbot, which has been successfully answering live customer queries in the wild.

Extended summary stats for each dataset used can be found in Table 1.

4 Methodology

4.1 Help Articles Data Preparation

We collected data from the company's public URLs and help and support PDFs. For pages containing tables, these were extracted and converted into markdown format before being passed to the models for embedding creation. This led to better retrieval performance for queries that needed information in the tables.

For generating recommendations, we utilized five different promising embedding models (see Section 4.2) to create an unbiased set of documents to be shared with SMEs. The top 10 retrieved articles from each model were collected. We followed a systematic process: selecting and stacking the rank 1 article from each model and removing dupli-

Dataset	Queries count	Rel D/Q	Chunks 512	Chunks 2048	Total documents	Median tokens	Tokens p75	Max tokens
FiQA	648	2.6	60,314	57,658	57,638	115	206	3,471
HotpotQA	7,405	2	66,790	66,581	66,581	486	690	8,263
SciDocs	1,000	4.9	27,234	25,736	25,657	187	245	6,980
Help Articles	310	3.6	28,870	20,243	15,848	237	480	125,248

Table 1: Dataset statistics including total number of queries, chunk counts for different chunk sizes, total documents, median token count, 75th percentile token count, and maximum token count

cates, then proceeding similarly with rank 2 articles, and so on until we obtained 10 unique recommendations for each question in our dataset. Finally, we randomized the order of these top 10 recommendations before sharing them with the SMEs.

For SME feedback, we asked experts to rank the articles based on their relevance to each query. They could use the links provided, but also add their own in the ranked list if they found that the answers were coming from beyond the list provided. This option was utilized by the SMEs in 20% of the cases. We also collected feedback on the overall answer quality.

4.2 Embedding models

The selected embedding models consist of some of the top performing models on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023) at the time of writing this paper:

- stella 1.5B (Zhang et al., 2025a)
- text-embedding-3-large (OpenAI, 2024)
- gemini-large-03-07 (Lee et al., 2025)
- Alibaba-NLP/gte-Qwen2-1.5B-instruct (Li et al., 2023)
- BM25 (Robertson and Zaragoza, 2009)

We consider BM25 as it is still a widely used keyword-based approach. Also, we had seen in our earlier tests for one of our company's products that it does add some value when some specific error codes are mentioned in the query by the user while facing issues with the product.

Additional models were initially experimented with, but these were later dropped as they had a larger/more recent model from the same provider available and/or were performing lower. These models are:

- stella 400M (Zhang et al., 2025a)
- text-embedding-ada-002 (OpenAI, 2022)
- text-embedding-005-gemini (Lee et al., 2025)
- finBERT (Araci, 2019) and a fine-tuned version with proprietary data. More information about this fine-tuning can be found in Appendix D

4.3 Chunking for embeddings

We generated document chunks of varying lengths (512 & 2048 tokens) and evaluated performance across these configurations. The BeIR datasets contained relatively concise documents with limited token counts, resulting in minimal performance variation between configurations. Nevertheless, models utilizing 2048-token segments consistently demonstrated superior performance compared to 512-token, as this length preserves the coherence of documents that marginally exceed the 512-token threshold. We conducted additional experiments with 1024 and 4096-token segments, which can be found in Table 6 in the Appendix, but for clarity and conciseness, we present performance metrics exclusively for the 2048 token configuration.

4.4 Reciprocal Rank Fusion(RRF)

For each dataset and candidate embedding model, we evaluated retrieval effectiveness using Recall@50 and Recall@10 metrics. This initial assessment revealed a substantial performance disparity between Recall@50 and Recall@10 across all datasets. Subsequently, we identified the highest-performing model for each dataset (based on Recall@50) and implemented pairwise RRF between that model and each alternative candidate model. While more comprehensive model combinations were feasible, we prioritized solution stability and deployment simplicity while still achieving significant performance enhancements.

4.5 Embedding Concatenation

Additionally, we investigated embedding concatenation as a lightweight fusion mechanism to integrate complementary signals from multiple embedding models. Specifically, we normalized all dense embeddings to unit length and performed pairwise concatenation between the best-performing model on each dataset and each of the remaining three dense embedding models, yielding multiple augmented representations per candidate passage.

Notably, the performance gains observed were comparable to those achieved with Reciprocal

Model	FIQA		SciDocs		Help Articles		HotpotQA	
	R@50	R@10	R@50	R@10	R@50	R@10	R@50	R@10
gemini_large_03_07	**81.8%**	**65.7%**	45.5%	**27.1%**	**88.0%**	**69.4%**	**97.8%**	**94.6%**
Stella 1.5B	81.7%	63.2%	**46.5%**	26.9%	86.4%	68.5%	95.91%	89.9%
text-embedding-3-large	78.0%	63.0%	42.5%	25.1%	86.3%	67.0%	94.6%	87.9%
gte-Qwen2-1.5B	80.3%	61.8%	43.7%	24.7%	83.7%	63.0%	92.4%	85.2%
BM25	38.1%	23.9%	21.6%	12.3%	54.3%	34.8%	70.1%	61.2%

Table 2: Recall metrics (Recall@50 and Recall@10) for different models across FIQA, SciDocs, Help Articles, and HotpotQA datasets. The model chunk size is 2048 in each case.

	FIQA		SciDocs		Help Articles		HotpotQA	
Model	Recall@50	Recall@10	Recall@50	Recall@10	Recall@50	Recall@10	Recall@50	Recall@10
Champion (gemini_large_03_07)	81.8%	65.7%	45.5%	27.1%	88.0%	69.4%	**97.8%**	**94.6%**
Champion + Stella 1.5B	84.3%	**66.7%**	**47.3%**	**27.5%**	89.1%	71.5%	97.7%	93.4%
Champion + text-embedding-3-large	82.3%	66.6%	45.4%	26.8%	89.2%	**73.2%**	97.7%	92.6%
Champion + gte-Qwen2-1.5B	**84.8%**	65.8%	46.3%	27.2%	**89.8%**	70.8%	97.4%	90.8%
Champion + BM25	75.1%	49.0%	41.0%	20.8%	87.0%	58.1%	97.4%	89.1%

Table 3: Reciprocal Rank Fusion results. Recall after combining the retrieval results from the champion model (gemini_large_03_07) in Table 2 with the rest of the candidates.

	FIQA		SciDocs		Help Articles		HotpotQA	
Model	Recall@50	Recall@10	Recall@50	Recall@10	Recall@50	Recall@10	Recall@50	Recall@10
Champion	**84.3%**	66.7%	47.3%	27.5%	89.2%	73.2%	97.8%	94.6%
Champion + Cross Encoder	84.3%	61.6%	47.3%	22.9%	89.2%	69.4%	97.8%	94.3%
Champion + LLM Reranking	84.0% ± 0.0	64.7% ± 0.4	47.4% ± 0.0	24.9% ± 0.2	**89.2%** ± 0.0	**74.7%** ± 0.0	**97.8%** ± 0.0	**96.4%** ± 0.0
Champion + SlideGAR	84.0% ± 0.0	**66.7%** ± 0.5	**47.4%** ± 0.0	**27.7%** ± 0.1	89.2% ± 0.0	72.2% ± 0.0	97.8%	96.1%
BM25	38.1%	23.9%	21.6%	12.3%	54.3%	34.9%	70.1%	61.2%
BM25 + Cross Encoder	40.2%	34.9%	21.7%	14.9%	67.9%	55.2%	71.0%	69.2%
BM25 + LLM Reranking	38.1%	23.9%	21.7%	16.5%	67.9%	55.3%	70.1%	61.2%

Table 4: Re-ranking results. Different re-ranking methods applied to the best approach from Table 3 for each dataset. For FIQA and SciDocs it is gemini_large_03_07 + Stella 1.5B, for Help Articles gemini_large_03_07 + text-embedding-3-large, and for HotpotQA gemini_large_03_07. Confidence intervals (95%) are shown where available. All values are rounded to one decimal place. Highest mean value in each column is **bolded**.

Rank Fusion (RRF), with detailed results reported in Appendix F. This suggests that embedding concatenation and RRF are *equally effective* fusion strategies for combining retrieval signals.

However, for operational simplicity and consistency in downstream re-ranking experiments (cross-encoder and LLM-based), we selected RRF as the primary fusion method. This choice allows us to build a unified pipeline where all re-ranking strategies are evaluated on top of the same high-quality top-50 candidate set. We therefore generate all further results using the RRF-enhanced retrieval outputs only.

4.6 Reranking Strategies

We performed re-ranking on the champion models for each data set obtained post RRF (Table 3). Details on the results of the same can be found in Table 4.

4.6.1 Cross-encoder Reranking

To address the notable performance gap between Recall@10 and Recall@50, we employed cross-encoder reranking—a widely recognized methodology for refining the ranking of *top_k* retrieved documents. This investigation incorporates *Alibaba-NLP/gte-reranker-modernbert-base* (Zhang et al., 2024) in its comparative analysis, as it has a very competitive performance in several text embedding and text retrieval evaluation tasks. This cross-encoder architecture could enable more sophisticated semantic matching than initial retrieval models alone.

4.6.2 LLM Reranking

We provide an LLM with the top 50 retrieved documents and ask it to return an ordered list with the indices of the top 10 most relevant documents for the provided query. To optimize the performance on this LLM task, experiments are carried out with:

- **Models**: We primarily use gpt-4.1 for reranking. We have also experimented with gpt-4.1-mini as a cost-effective alternative to gpt-4.1 in Table 7. A cost analysis per query can be found in Appendix A, to demonstrate the feasibility depending on the user's budget.

- **Prompts**: Different LLM prompt-tuning methods have been evaluated, including zero-shot, one-shot and meta-prompting. Passing more than one full example has not been evaluated as each example contains 50 documents, hence being costly.

Another LLM based re-ranking strategy tested

was SlideGAR.

4.7 Meta-Prompting

Hou et al. (2023) introduced meta-prompting as a technique used to improve or generate a task-specific prompt, often leveraging examples from a dataset. We use a similar approach to come up with a prompt to learn from hard examples in the training set:

(1) For a subsample of the training set (1000 samples), retrieve the top 50 documents.

(2) If recall@50 >= 0.5 (there are relevant articles within the top 50), run LLM reranking.

(3) If recall@50 - recall@10 after re-ranking > 0.3, there was a re-ranking failure: use this example to run meta-prompting and update the system prompt.

Appendix B contains Figure 1 with the meta-prompt used to obtain an enhanced system prompt.

5 Results

5.1 Evaluation Metrics

While evaluating performance on the Help Articles dataset, we observed that whenever relevant documents were present within even the top three retrieval results, the LLM generated accurate and comprehensive responses in over 92% of cases. This paper's investigation is thus a direct attempt to close the substantial gap between Recall@10 and Recall@50, which was identified as the primary performance challenge. Although we focus on the recall metrics in this paper, we have still provided nDCG scores for our main experiments in appendix G to provide a more complete picture for the IR community.

5.2 Retrieval: Embedding models

Table 2 presents the retrieval results using various embedding models and BM25. Gemini embeddings consistently outperform all other embedding models, with Stella 1.5B following closely behind. These findings align with MTEB rankings, where both models appear in the top 10. Interestingly, text-embedding-3-large demonstrated superior performance compared to Qwen2-1.5B when retrieving 10 documents. As expected, BM25 ranks lowest among all approaches.

5.3 Retrieval: Reciprocal Rank Fusion

Since gemini_large_03_07 emerged as the best embedding model in almost all datasets and metrics, we designated it as the champion model and combined its retrieval results with all other approaches using Reciprocal Rank Fusion. Table 3 displays these findings. Gemini's Recall@10 improved across all four datasets, with gains of up to 3.8 percentage points achieved via different retriever ensembles, demonstrating that a well-combined ensemble can surpass even the strongest individual model.

While many traditional RAG pipelines employ hybrid search combining an embedding model with BM25, our results clearly indicate that including BM25 in the combination significantly diminishes overall retrieval performance compared to using either a single embedding model or a combination of two embedding models.

5.4 Re-ranking

Table 4 presents the results of applying various re-ranking techniques to the best models from Table 3 for each dataset. Some interesting observations are:

(1) With sufficiently powerful embedding models, cross-encoders appear to be no longer necessary, as they actually decrease the recall@10 across all datasets.

(2) LLM Re-ranking with GPT4.1 outperforms all other approaches in 2 of the 4 datasets, while remaining competitive in the others. This represents impressive performance for a zero-shot, out-of-the-box model, especially considering it is being compared to models specifically trained for retrieval and re-ranking tasks. This suggests that future, more powerful LLMs might achieve even better results, and that fine-tuning an LLM specifically for re-ranking could be worthwhile, given that its base version already matches top performances.

(3) SlideGAR demonstrated performance comparable to the champion model. It outperformed LLM re-ranking in 2 datasets while being surpassed in the other 2.

Additional LLM re-ranking ablation studies can be found in Appendix C, where One-shot and Meta-prompting techniques demonstrated slight improvements in re-ranking performance.

6 Conclusion & Future Research

Our comprehensive analysis of advanced retrieval strategies for Large Language Models (LLMs) within Retrieval-Augmented Generation (RAG) systems has yielded several critical insights and

actionable strategies. Despite achieving notable improvements, a persistent gap remains between Recall@10 and Recall@50 across various datasets, indicating significant room for optimization in document retrieval accuracy.

The implementation of Reciprocal Rank Fusion (RRF) and LLM re-ranking has demonstrated decent gains, underscoring their effectiveness in enhancing retrieval performance. Cross-encoder reranking also contributed positively, albeit variably across different setups. These results solidify the importance of these advanced techniques in refining the retrieval process.

We make the following strategic recommendations for Building an Effective Retrieval Pipeline:

1. **Initial Testing**: Conduct thorough testing with top-performing embedding models across different chunk sizes to understand their baseline performance.

2. **RRF or Concatenation**: Select a champion model and apply either pairwise RRF or embedding concatenation with other candidates. Both methods yield comparable gains in Recall@10.

3. **Advanced Re-ranking**: With the refined model from the fusion phase, experiment with adaptive and list-wise LLM re-ranking, along with cross-encoder re-ranking, to further optimize the retrieval outputs.

Our study also specifically highlighted limitations in the traditional BM25 algorithm. Despite its widespread use, BM25 was found to perform poorly compared to state-of-the-art embedding models, especially when not combined with advanced re-ranking techniques. This is particularly evident in scenarios that are not heavily keyword-focused, where the semantic richness of queries and documents is poorly captured by the purely lexical approach of BM25. The findings suggest that unless the user's dataset and queries are heavily keyword-intensive, BM25 is unlikely to improve retrieval performance significantly and might even degrade it when combined with more sophisticated models.

We carried out some experiments using HyDE but the results were not promising (details are in Appendix E). We saw minimal gains from contextual embeddings on the Help Articles dataset, but could not test on other data-sets owing to the lower chunk sizes in those. Our hypothesis still is that contextual embeddings could add value where chunking across long documents is needed.

In conclusion, our research highlights the critical interplay between various retrieval and re-ranking strategies in enhancing the performance of RAG systems. The outlined strategic approach for constructing retrieval pipelines provides a structured pathway for future implementations. Further investigations into contextual embeddings and their application in handling extensive document sizes remain a promising avenue for advancing the state-of-the-art in retrieval technologies. This continual evolution in retrieval methodologies is crucial for leveraging the full capabilities of LLMs in generating contextually relevant and accurate responses.

Limitations

Our evaluation encompassed four diverse datasets, providing meaningful insights across different retrieval scenarios, though additional domain-specific applications could further validate our findings. As with all research in this rapidly evolving field, our results represent a snapshot of current capabilities, with the understanding that embedding models and LLMs continue to advance.

While our computational approach allowed us to evaluate several leading embedding models and re-ranking techniques, we necessarily focused on the most promising candidates rather than exhaustively testing all available models. This strategic approach enabled deeper analysis of high-performing systems while acknowledging that specialized domain-specific embedding models might offer advantages in certain contexts.

Our findings regarding BM25's diminished utility when combined with modern embedding models reflect patterns observed across our test datasets, though specific use cases involving highly technical or specialized vocabulary may still benefit from lexical matching approaches. Similarly, while cross-encoders did not improve performance in our experiments, alternative implementations might yield different results in specific contexts.

For LLM re-ranking, we primarily leveraged GPT-4.1, which demonstrated impressive capabilities. While resource considerations limited our ability to test all available LLMs, the strong performance of GPT-4.1 suggests promising directions for future work.

Finally, our results point to LLM fine-tuning

for re-ranking as a compelling research direction. While implementation and testing of this approach fell outside our current scope, the strong zero-shot performance of LLMs suggests significant potential for further performance gains through targeted fine-tuning.

References

Anthropic. 2024. Introducing contextual retrieval.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *Preprint*, arXiv:1908.10063.

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

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 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*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. 2024. FIRST: Faster improved listwise reranking with single token decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8642–8652, Miami, Florida, USA. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. 2017. In defense of the triplet loss for person re-identification. *Preprint*, arXiv:1703.07737.

Yutai Hou, Hongyuan Dong, Xinghao Wang, Bohan Li, and Wanxiang Che. 2023. Metaprompting: Learning to learn better prompts. *Preprint*, arXiv:2209.11486.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. 2025. Sufficient context: A new lens on retrieval augmented generation systems. In *The Thirteenth International Conference on Learning Representations*.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, and 28 others. 2025. Gemini embedding: Generalizable embeddings from gemini. *Preprint*, arXiv:2503.07891.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893, Miami, Florida, US. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. *Preprint*, arXiv:2210.07316.

OpenAI. 2022. New and improved embedding model. Accessed: 2025-07-02.

OpenAI. 2024. New embedding models and api updates. Accessed: 2025-07-02.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Mandeep Rathee, Sean MacAvaney, and Avishek Anand. 2025a. Guiding retrieval using llm-based listwise rankers. *Preprint*, arXiv:2501.09186.

Mandeep Rathee, Sean MacAvaney, and Avishek Anand. 2025b. Guiding retrieval using llm-based listwise rankers. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part I*, page 230–246, Berlin, Heidelberg. Springer-Verlag.

David Rau, Shuai Wang, Hervé Déjean, Stéphane Clinchant, and Jaap Kamps. 2025. Context embeddings for efficient answer generation in retrieval-augmented generation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, page 493–502, New York, NY, USA. Association for Computing Machinery.

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

Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. In defense of cross-encoders for zero-shot retrieval. *arXiv preprint arXiv:2212.06121*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Qwen Team. 2025. Qwen2.5-1m: Deploy your own qwen with context length up to 1m tokens.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021a. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021b. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *Preprint*, arXiv:2104.08663.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *Preprint*, arXiv:2112.07577.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. *Preprint*, arXiv:2401.00368.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, and 9 others. 2025. Qwen2.5-1m technical report. *arXiv preprint arXiv:2501.15383*.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025a. Jasper and stella: distillation of sota embedding models. *Preprint*, arXiv:2412.19048.

Dun Zhang, Panxiang Zou, and Yudong Zhou. 2025b. Dewey long context embedding model: A technical report. *Preprint*, arXiv:2503.20376.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.

A Appendix: LLM Reranking cost

This appendix contains the approximate cost of LLM re-ranking with proprietary models, in order to demonstrate its financial feasibility for cost-effective LLMs. The costs in Table X correspond to 1 query, with an LLM re-ranking 50 documents of 512 tokens each (25,000 tokens in total approximately). The cost per token in the output is negligible as it is just a list with 10 indices, so the cost from the input tokens is what is measured. As per the table's creation date. The OpenAI's pricing page shows the following prices:

- gpt-4.1 nano: $0.10 per 1M input tokens
- gpt-4.1 mini: $0.40 per 1M input tokens
- gpt-4.1: $2 per 1M input tokens

LLM	Cost per query
gpt-4.1-nano	$0.0025
gpt-4.1-mini	$0.01
gpt-4.1	$0.05

Table 5: **LLM re-ranking cost.** Price per query for different OpenAI models, assuming 50 documents of 512 tokens.

With open-source LLMs, even fine-tuned for this task, the cost of LLM re-ranking would just consist on the infrastructure to host-them.

	Stella 1.5B		BM25	
Tokens per chunk	**Recall@50**	**Recall@10**	**Recall@50**	**Recall@10**
512	81.9%	63.0%	53.9%	33.5%
1024	83.0%	65.7%	54.6%	34.5%
2048	84.9%	66.5%	54.3%	34.8%
4096	85.5%	67.5%	54.6%	35.2%

Table 6: Chunk size comparison on Help Articles. Recall@50 and @10 for 4 common chunk sizes with two of the candidate retrieval models.

B Appendix: Meta-prompt

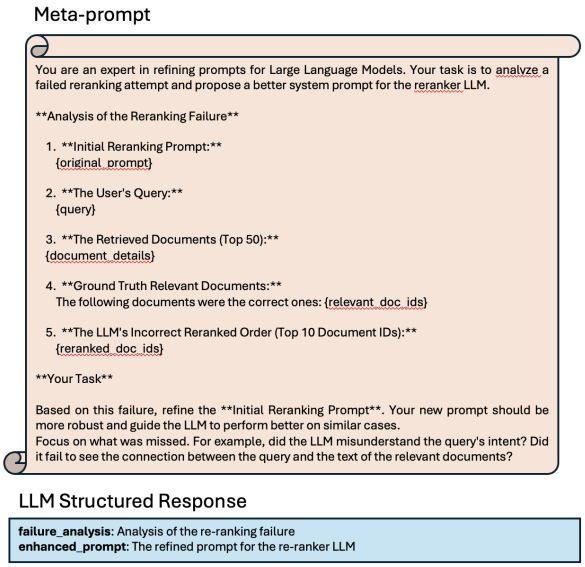

Figure 1: **Meta prompt.** Prompt used to refine the system prompt to improve the performance of LLM re-ranking. Asking the LLM to first provide a failure analysis allows it to reason over how to improve the system prompt, which is generated afterwards.

C Appendix: LLM Reranking ablation study

Different LLM prompting techniques have been explored in order to improve its performance, and these results can be found in Table 7.

D Appendix: Finetuning embeddings

One of the strategies explored is to finetune the embeddings with the objective to improve Recall in our internal Dataset (Help Articles). Given that we didn't have enough samples to be considered as training data we explore the use of techniques described in (Wang et al., 2022) and (Wang et al., 2024) to generate synthetic triplets. Two different prompts have been explored for generation of samples :

1. Given a specific document, generate a triplet of (query, positive chunk, hard negative chunk). GPT4.1 has been used following a similar approach as the one described in (Wang et al., 2024) to generate around 5k samples. We fine-tuned a stella 400M(Zhang et al., 2025a) using the library sentence transformers (Thakur et al., 2021a) for 1 epoch with a learning rate of 6.25e-6, batch size of 8, linear warmup of 500 steps and TripletLoss(Hermans et al., 2017). An example of the prompt and the response can be seen on Figures 2 and 3

2. Given a document generate a set of questions for that document. Qwen2.5-7B-Instruct-1M (Yang et al., 2025) (Team, 2025) has been used to generate 55,257 pairs (query,document). We fine-tuned a stella 400M (Zhang et al., 2025a) using the library sentence transformers for 1 epoch with a learning rate of 6.25e-6, batch size of 8, linear warmup of 500 steps and MultipleNegativesRankingLoss (Henderson et al., 2017). An example of the prompt can be seen on Figure 4

In addition, we explored fine-tuning a finBERT model (Araci, 2019) using GPL (Wang et al., 2022) but, as we will describe later, the results were underperforming compared to Stella and other SOTA models.

As presented in Table 8, the stella 400M model demonstrates strong performance on Help Articles achieving high recall@50. A larger chunksize of 2048 generally proves beneficial for stella models.

While the base stella 400M model already exhibits robust performance, finetuning with Qwen (Yang et al., 2025) (Team, 2025) questions further enhances recall metrics, positioning it as a particularly effective choice for article retrieval.

In contrast, finBERT models, even with effective finetuning such as Generative Pseudo-Labeling (Wang et al., 2022), perform substantially poorer across all evaluated metrics compared to the stella variants. This performance disparity underscores a fundamental difference in their suitability for this

	FIQA		SciDocs	
Model	**Recall@50**	**Recall@10**	**Recall@50**	**Recall@10**
Champion (gemini_large_03_07 + Stella 1.5B)	84.3%	**66.7%**	47.3%	27.5%
Champion + LLM Reranking GPT 4.1	84.3%	63.5%	47.3%	26.5%
Champion + LLM Reranking GPT 4.1 mini	84.3%	63.1%	47.3%	24.9%
Champion + One-shot LLM Reranking GPT 4.1	84.3%	63.3%	47.3%	27.4%
Champion + One-shot + Meta-prompting GPT 4.1	84.3%	64.8%	47.3%	**28.4%**

Table 7: LLM Re-ranking ablation study. Champion model is the combination of gemini_large_03_07 and Stella 1.5B through RRF. One-shot corresponds to the hardest example found in the train-set. Meta-prompting references the use of an enhanced prompt found through meta-prompting

Model	chunk_size	ndcg@5	ndcg@10	recall@10	recall@50
stella 400M finetuned on Qwen questions	2048	46.1%	51.2%	63.1%	85.3%
stella 400M	2048	48.2%	52.4%	63.3%	83.5%
stella 400M finetuned on Qwen questions	512	41.9%	47.6%	59.0%	82.8%
stella 400M finetuned GPT triplets	2048	45.3%	49.9%	58.1%	80.1%
stella_400M	512	43.7%	49.0%	59.3%	79.7%
stella 400M finetuned GPT triplets	512	42.3%	48.0%	55.8%	76.2%
finBERT GPL	512	20.9%	25.0%	34.7%	62.5%
finBERT	512	9.8%	12.5%	15.4%	36.8%

Table 8: Retrieval metrics on Help Articles dataset for finetuned models

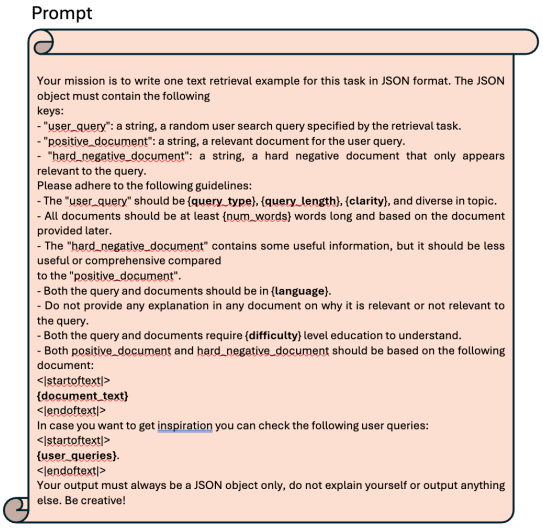

Figure 2: Description of the prompt for triplet generation, the different variables follow the same values as (Wang et al., 2024)

document text is the document we want to obtain the triplet for and **user queries** is a set of queries that are extracted from our internal database

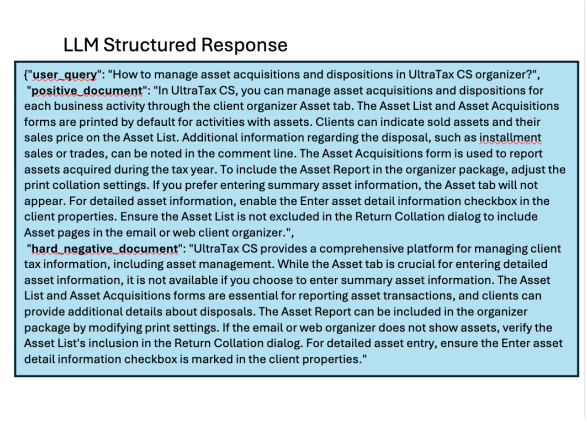

Figure 3: Example of response for the triplet generator

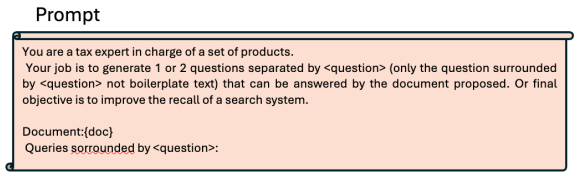

Figure 4: Description of the prompt for query generation

specific information retrieval task.

For the triplet generation strategy, results were underperforming compared to vanilla stella 400M, we think that hard negative selection should be improved, for instance by, not choosing the hard negative from the same document as the positive pair.

Fine-tuning embeddings shows that improving over the baseline model could be done by generating synthetic samples over a custom dataset. Improving small languages models could be interesting in setups where the amount of documents to index makes it prohibitively costly to execute bigger models such as stella 1.5B or some proprietary models.

E Appendix: HyDE

We evaluated the HyDE approach on a subset of Help Articles (300 queries). The hypothetical documents for the queries were generated using gpt-4o. The embeddings of the dataset, queries and

	FIQA		SciDocs		Help Articles		HotpotQA	
Model	**Recall@50**	**Recall@10**	**Recall@50**	**Recall@10**	**Recall@50**	**Recall@10**	**Recall@50**	**Recall@10**
Champion (gemini_large_03_07)	81.8%	65.7%	45.5%	27.1%	88.0%	69.4%	**97.8%**	**94.6%**
Champion + Stella 1.5B	**84.5%**	**66.8%**	**47.8%**	**27.8%**	88.0%	**71.8%**	97.6%	93.7%
Champion + text-embedding-3-large	81.7%	66.6%	45.0%	27.1%	88.4%	71.0%	97.3%	92.8%
Champion + gte-Qwen2-1.5B	84.1%	65.2%	46.5%	27.4%	**88.8%**	71.6%	96.4%	91.6%

Table 9: Embedding concatenation results. Recall after combining the retrieval results from the champion model (gemini_large_03_07) in Table 2 with the rest of the candidates.

	FIQA		SciDocs		Help Articles		HotpotQA	
Model	**nDCG@50**	**nDCG@10**	**nDCG@50**	**nDCG@10**	**nDCG@50**	**nDCG@10**	**nDCG@50**	**nDCG@10**
Champion model (gemini_large_03_07)	61.7%	56.9%	32.5%	25.6%	62.94%	58.21%	**91.5%**	**90.6%**
Champion model + stella 1.5B	**64.5%**	**59.3%**	**33.7%**	**26.2%**	**64.60%**	**60.62%**	90.8%	89.7%
Champion model+ text-embedding-3-large	63.1%	58.6%	32.3%	25.6%	64.36%	60.25%	89.8%	88.6%
Champion + gte-Qwen2-1.5B	63.6%	57.9%	32.9%	25.7%	64.85%	60.81%	89.0%	87.6%

Table 10: Embedding concatenation results. nDCG after combining the retrieval results from the champion model (gemini_large_03_07) in Table 2 with the rest of the candidates.

the hypothetical documents were all generated using text-embedding-ada-002. We considered text-embedding-ada-002 as the baseline in this experiment, i.e., the query embeddings were used to obtain the 50 most relevant documents from the dataset. In the HyDE approach, the embeddings of the hypothetical documents were used to obtain the 50 most relevant documents from the dataset. We find that the Recall@10 of the baseline is 66.4% while that of HyDE is 62.8%, significantly degrading the performance over the baseline. The Recall@50 of the baseline is 82.8% while that of HyDE is 82.4%.

F Appendix: Embedding Concatenation Results

We investigated embedding concatenation as a lightweight alternative to Reciprocal Rank Fusion (RRF) for combining signals from multiple dense retrievers. All embeddings were normalized to unit length and concatenated pairwise between the champion model (gemini_large_03_07) and each of the remaining three dense models.

As shown in Tables 9 and 10, the performance of embedding concatenation is nearly identical to that of RRF across both Recall@50/10 and nDCG@50/10 metrics on all four datasets (FIQA, SciDocs, Help Articles, HotpotQA). Differences are within ±0.3 percentage points, indicating no statistically or practically significant advantage of one method over the other.

This equivalence supports our recommendation to treat RRF and embedding concatenation as equally viable fusion strategies. However, all downstream re-ranking results (cross-encoder and LLM-based) are reported using RRF only, to maintain consistency in the evaluation pipeline and simplify deployment.

We therefore conclude that either method can be used interchangeably in production RAG systems, with the final choice guided by engineering constraints (e.g., index size for concatenation vs. rank aggregation logic for RRF).

G Appendix: nDCG Results for Reference

This appendix reports nDCG@50 and nDCG@10 for all experiments in Tables 2, 3, and 4, included for reference only to support the information retrieval (IR) community. While our primary evaluation uses Recall (Section 5.1), nDCG provides a complementary view of ranking quality by assigning higher weights to relevant documents placed earlier in the list. Notably, the top-performing models and fusion strategies are nearly identical under both Recall and nDCG. Results can be found in Tables 11, 12 and 13

Model	FIQA		SciDocs		Help Articles		HotpotQA	
	NDCG@50	NDCG@10	NDCG@50	NDCG@10	NDCG@50	NDCG@10	NDCG@50	NDCG@10
gemini_large_03_07	**61.7%**	**56.9%**	**32.5%**	**25.6%**	**62.9%**	**58.2%**	**91.5%**	**90.6%**
stella 1.5B	61.2%	55.6%	**32.5%**	25.1%	62.5%	**58.2%**	87.6%	85.9%
text-embedding-3-large	59.7%	55.1%	29.8%	23.4%	61.2%	56.8%	85.3%	83.4%
gte-Qwen2-1.5B	59.3%	53.9%	30.4%	23.3%	56.7%	51.6%	83.8%	81.8%
bm25	22.7%	18.8%	15.2%	11.8%	32.1%	27.0%	57.9%	55.4%

Table 11: NDCG metrics (NDCG@50 and NDCG@10) for different models across FIQA, SciDocs, Help Articles, and HotpotQA datasets. The model chunk size is 2048 in each case.

	FIQA		SciDocs		Help Articles		HotpotQA	
Model	NDCG@50	NDCG@10	NDCG@50	NDCG@10	NDCG@50	NDCG@10	NDCG@50	NDCG@10
Champion (gemini_large_03_07)	61.7%	56.9%	32.5%	25.6%	62.9%	58.2%	91.5%	90.6%
Champion + stella 1.5B	**64.1%**	**58.9%**	**33.4%**	**25.9%**	**64.3%**	**60.4%**	90.4%	89.2%
Champion + text-embedding-3-large	62.9%	58.2%	32.3%	25.3%	63.0%	59.2%	89.5%	88.1%
Champion + gte-Qwen2-1.5B	63.5%	57.9%	32.7%	25.5%	62.9%	58.7%	88.6%	86.8%
Champion + bm25	46.6%	39.1%	27.0%	19.5%	63.9%	60.2%	83.4%	81.0%

Table 12: Reciprocal Rank Fusion results. NDCG after combining the retrieval results from the champion model (gemini_large_03_07) in Table 11 with the rest of the candidates.

	FIQA		SciDocs		Help Articles		HotpotQA	
Model	NDCG@50	NDCG@10	NDCG@50	NDCG@10	NDCG@50	NDCG@10	NDCG@50	NDCG@10
Champion	**64.1%**	**58.9%**	**33.4%**	25.9%	63.0%	59.2%	91.6%	90.6%
Champion + Cross Encoder	58.2%	51.5%	30.3%	21.3%	61.1%	55.9%	91.7%	90.7%
Champion + LLM Reranking	63.9%±0.1	57.8%±0.2	32.7%±0.2	24.0%±0.2	**65.3%**±0.0	**61.2%**±0.0	94.0%±0.0	93.6%±0.0
Champion + SlideGAR	62.8%±0.2	57.1%±0.2	33.3%±0.0	**25.6%**±0.1	62.6%±0.0	58.9%±0.0	**94.2%**	**93.7%**
bm25	22.7%	18.8%	15.2%	11.8%	32.1%	27.0%	57.9%	55.4%
BM25+ Cross Encoder	33.3%	31.8%	15.5%	11.9%	50.3%	47.1%	69.1%	68.6%
BM25+ GPT 4.1	22.7%	18.8%	19.2%	16.4%	51.7%	48.5%	57.9%	55.4%

Table 13: Re-ranking results. Different re-ranking methods applied to the best approach from Table 3 for each dataset. For FIQA and SciDocs it is gemini_large_03_07 + stella 1.5B, for Help Articles gemini_large_03_07 + stella 1.5B, and for HotpotQA gemini_large_03_07.