# Deep Retrieval at CheckThat! 2025: Identifying Scientific Papers from Implicit Social Media Mentions via Hybrid Retrieval and Re-Ranking

Notebook for the CheckThat! Lab at CLEF 2025

Pascal J. Sager[1,2,3,*], Ashwini Kamaraj[1], Benjamin F. Grewe[3] and Thilo Stadelmann[2,4]

[1]University of Zurich, Rämistrasse 71, 8006 Zurich, Switzerland
[2]Centre for Artificial Intelligence, Zurich University of Applied Sciences, Technikumstrasse 71, 8401 Winterthur, Switzerland
[3]Institute of Neuroinformatics, ETH Zurich and University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland
[4]European Centre for Living Technology, Dorsoduro 3246, 30123 Venice, Italy

## Abstract

We present the methodology and results of the *Deep Retrieval* team for subtask 4b of the CLEF CheckThat! 2025 competition, which focuses on retrieving relevant scientific literature for given social media posts. To address this task, we propose a hybrid retrieval pipeline that combines lexical precision, semantic generalization, and deep contextual re-ranking, enabling robust retrieval that bridges the informal-to-formal language gap. Specifically, we combine BM25-based keyword matching with a FAISS vector store using a fine-tuned INF-Retriever-v1 model for dense semantic retrieval. BM25 returns the top 30 candidates, and semantic search yields 100 candidates, which are then merged and re-ranked via a large language model (LLM)-based cross-encoder.

Our approach achieves a mean reciprocal rank at 5 (MRR@5) of 76.46% on the development set and 66.43% on the hidden test set, securing the **1st** position on the development leaderboard and ranking **3rd** on the test leaderboard (out of 31 teams), with a relative performance gap of only 2 percentage points compared to the top-ranked system. We achieve this strong performance by running open-source models locally and without external training data, highlighting the effectiveness of a carefully designed and fine-tuned retrieval pipeline.

## Keywords

Information retrieval, Scientific document search, Social media fact verification, Fact-checking, CLEF



## 1. Introduction

In the age of online misinformation, tracing social media claims back to their original scientific sources is crucial for automated fact-checking and evidence-based verification [1, 2]. However, this task is inherently challenging due to the linguistic and structural gap between informal, user-generated content and formal scientific literature. Social media posts often paraphrase, summarize, or loosely reference scientific findings, rarely using standardized terminology or explicit citations. These ambiguities make it difficult to reliably identify the corresponding scientific publications.

Bridging this gap requires retrieval systems that can handle domain-specific vocabulary, implicit references, and abstract semantics [2]. Subtask 4b of the CLEF CheckThat! 2025 competition [3, 4, 5] exemplifies this challenge, focusing on retrieving scientific sources for social media claims. Figure 1 illustrates the task and our proposed solution: A *hybrid retrieval pipeline* designed specifically for cross-domain scientific source retrieval. Our method integrates:

1. **Lexical retrieval** with BM25 [6, 7] to capture explicit term overlap (e.g., named entities, keywords);
2. **Semantic retrieval** using a FAISS-based [8, 9] vector store to compare dense embeddings obtained with a fine-tuned INF-Retriever-v1 [10] model, enabling the detection of semantic overlaps;

*CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain*
*Corresponding author.
pascaljosef.sager@uzh.ch (P. J. Sager); ashwini.kamaraj@uzh.ch (A. Kamaraj); bgrewe@ethz.ch (B. F. Grewe); stdm@zhaw.ch (T. Stadelmann)
0000-0002-8084-2317 (P. J. Sager); 0000-0001-8560-2120 (B. F. Grewe); 0000-0002-3784-0420 (T. Stadelmann)

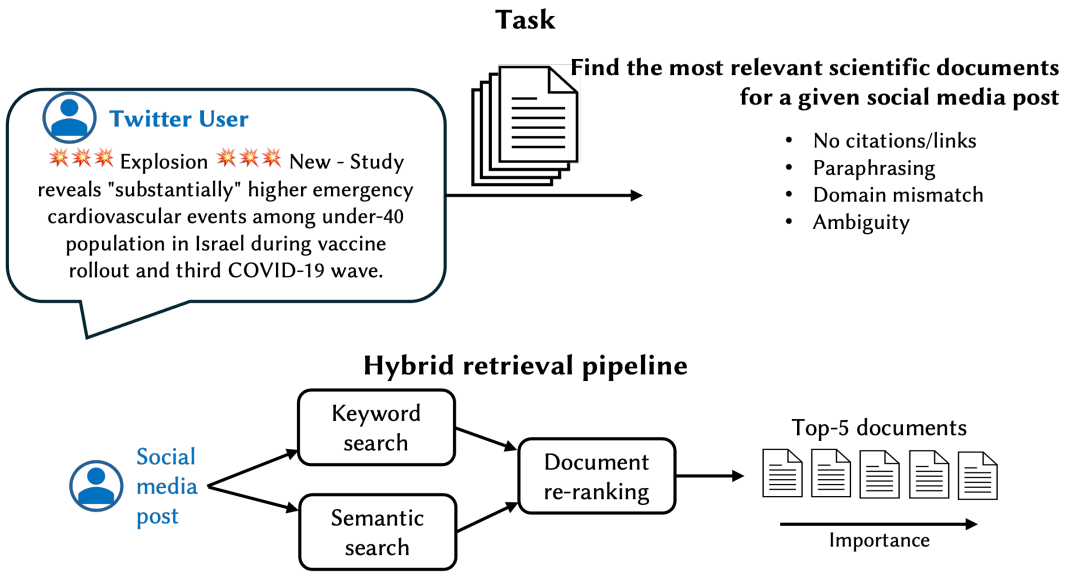


**Figure 1:** The **top** of the figure displays the task of finding relevant publications from a large pool of scientific documents, given a social media post. The **bottom** part of the figure provides an overview of our pipeline, consisting of three modules to predict the top five scientific documents ordered by their importance.

3. **Re-ranking** with a large language model (LLM)-based cross-encoder [11, 12], which jointly encodes and scores pairs of social media posts and documents to refine relevance using deep contextual understanding.

This architecture is designed to harness the complementary strengths of these different retrieval methods.

We evaluate our pipeline on the CheckThat! 2025 Subtask 4b dataset, achieving a Mean Reciprocal Rank at 5 (MRR@5) of 76.46% on the development set (ranked **1st** on the leaderboard) and 66.43% on the test set (ranked **3rd** on the leaderboard out of 31 teams), with only a 2 percentage points lower score than the top-performing team. **Importantly, we achieved this strong score without using any external training data, metadata, external knowledge sources, or closed-source models,** making our approach broadly applicable and easily transferable to other domains and tasks. Overall, our main contributions are:

1. A robust hybrid information retrieval (IR) architecture tailored for scientific source retrieval from informal social media content;
2. Empirical evidence demonstrating the effectiveness of embedding fine-tuning and LLM-based re-ranking in bridging informal-to-formal domain gaps;
3. A comprehensive experimental analysis, including ablations and a comparison to a commercial baseline.

By publishing this well-engineered pipeline, we aim to support efforts to counter misinformation and offer a practical, open-source blueprint for cross-domain document retrieval.

## 2. Related Work

**Fact-Checking and Scientific Source Retrieval.** Automated fact-checking critically depends on robust document retrieval methods to identify evidence that supports or refutes a given claim [13]. The evolution of this field has progressed from early strategies utilizing structured knowledge bases and curated news sources [14] to approaches that exploit unstructured, domain-specific corpora [15]. A particularly challenging scenario involves retrieving scientific literature to verify claims originating

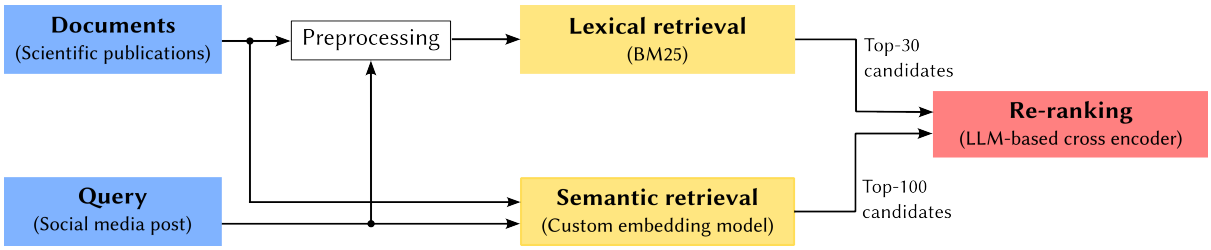


**Figure 2:** Overview of our retrieval pipeline. Scientific documents are indexed using two parallel retrieval mechanisms: A lexical retriever and a semantic retriever. Given a query (e.g., a social media post), the lexical retriever returns 30 candidate papers, and the semantic retriever outputs its top 100 candidates. These 130 candidates are then re-ranked using a cross-encoder.

from social media, due to the frequent lexical and conceptual mismatch between informal language and the academic writing style [16, 17, 18].

**Sparse vs. Dense Retrieval.** Retrieval methods are commonly grouped into sparse and dense approaches. *Sparse* approaches like BM25 [6, 7] rely on term overlap and excel with strong lexical alignment, using probabilistic relevance frameworks with saturation parameters and document length normalization for robust ranking. Conversely, *dense* retrieval uses neural networks to encode text into vector representations, enabling semantic similarity matching through metrics such as cosine similarity [19, 20]. Dense models are particularly advantageous in scenarios where claims are paraphrased or loosely aligned with scientific language, as is often the case in user-generated content. Although dense retrieval has historically required domain-specific fine-tuning [21, 22], recent foundation models pre-trained on diverse corpora exhibit strong generalization [10], increasingly blurring the distinction between general-purpose and domain-adapted retrieval.

**Hybrid Retrieval and LLM Re-Ranking.** Hybrid retrieval frameworks can combine sparse and dense retrieval by adding a subsequent re-ranking stage to merge their results and improve retrieval quality. Neural re-rankers [23] have demonstrated substantial improvements in ranking accuracy across multiple domains. Recently, large language models (LLMs) have been employed as cross-encoders, jointly encoding claim–document pairs to capture nuanced semantic relationships [11, 12]. In this work, we adopt such a hybrid retrieval architecture by combining sparse retrieval via BM25, dense neural retrieval, and LLM-based re-ranking to leverage the different strengths of these retrieval methods.

## 3. Methodology

Figure 2 illustrates our hybrid retrieval pipeline. In the official dataset, each document consists of a title, abstract, and limited metadata (e.g., authors, publication venue), but no full text. For indexing, we represent each document using only the title and abstract, concatenated in the format `[Title \n Abstract]`, explicitly ignoring metadata fields.

The document corpus is processed along two parallel paths: One path applies lexical preprocessing followed by a BM25-based sparse retriever; the other encodes raw text into dense vectors for semantic retrieval. At query time, the social media post undergoes similar dual processing. Both retrieval branches return ranked candidate sets, which are merged and re-ranked using a cross-encoder. We describe each component in the following.

### 3.1. Lexical Retrieval

We use BM25 [6, 7] for sparse retrieval, and rank documents based on n-gram overlap and frequency statistics. Lexical methods are particularly effective for matching query terms to titles and commonly used scientific expressions.

**Pre-Processing.** In contrast to the baseline BM25 provided by the challenge organizers [5], we apply additional normalization steps to improve match quality. Our pipeline includes lowercasing, punctuation removal, and subword tokenization using byte pair encoding (BPE) [24]. We chose subword tokenization over lemmatization to maximize n-gram overlaps between informal query terms and formal document vocabulary. Hashtags are removed, while symbols such as percentages (%) are preserved to maintain scientific meaning. This design choice specifically addresses the domain gap between informal social media queries and formal scientific language by increasing the chance of partial term matches. We detail additional pre-processing experiments, which were excluded from the pipeline, in Appendix A.

**Retrieval.** At inference time, the BM25 retriever returns the top-30 documents ranked by relevance. This candidate set provides strong lexical matches for downstream re-ranking and complements the semantic retriever.

### 3.2. Semantic Retrieval

To overcome vocabulary mismatches and paraphrasing issues, we implement dense retrieval based on transformer-derived embeddings [25], capturing semantic similarity between queries and documents.

**Embedding Model and Fine-tuning.** We initialize our dense retriever with the `INF-Retriever-v1` model [10], a fine-tuned variant of `gte-Qwen2-7B-instruct` [26], chosen for its strong long-text retrieval performance and open-source availability. We fine-tune it on the CLEF CheckThat! training set using the multiple negatives ranking (MNR) loss [27], training it to assign higher similarity scores to (social media post, document) pairs with known associations than to random negatives.

Fine-tuning uses a maximum input length of 8,192 tokens to avoid truncation. Queries and documents are tokenized independently, embeddings use last-token pooling, and LoRA adapters [28] are applied to the final eight transformer layers to reduce memory and training time [29]. We optimize with AdamW [30] using cosine learning rate decay and gradient accumulation. Full details of the fine-tuning setup are provided in Appendix B.

**Vector Store.** We pre-compute document embeddings and store them in a FAISS index [8, 9]. The embeddings are normalized using the L2 norm, allowing cosine similarity to be computed efficiently via dot products. At inference time, the social media post is encoded into a dense vector using the same model, and the top 100 most similar documents are retrieved. We avoid chunking abstracts, as empirical results have shown that full-document retrieval performs better.

### 3.3. Re-Ranking

While dense and sparse retrievers are computationally faster, the subsequent re-ranking process is computationally intensive. Unlike embedding models, which independently embed each document and query into vectors and compute similarity using a distance metric, the re-ranker processes each query-document *pair* jointly to directly output a similarity score. The computational cost of these pairwise comparisons limits re-ranking to small candidate subsets, making the initial retrieval stage essential for filtering documents.

**Ranking.** We evaluated various re-ranking models (see Appendix C) and selected `BAAI/bge-reranker-v2-gemma` [11, 12], an LLM-based cross encoder built on Gemma [31], as it performed best. To balance cost and performance, we re-rank the top 100 dense and top 30 BM25 candidates, favoring the former due to stronger individual performance (see Section 4). Empirically, increasing the number of BM25 candidates beyond 30 did not improve re-ranking performance but substantially increased computational cost, whereas increasing the dense candidates to 100 led to substantial gains. After removing duplicates within the 130 candidates for each query, the re-ranker scores all remaining candidates from scratch, and the top five results are returned as output.

**Table 1**
Performance comparison on the development and test sets across retrieval and re-ranking stages. Precision is not reported for the re-ranking stage, and Elasticsearch was only evaluated on the development set. Best results in each category are in bold. "FT" denotes fine-tuning.

| | DEVELOPMENT SET | | | | TEST SET | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR@k (%) | | Precision@k (%) | | MRR@k (%) | | Precision@k (%) | |
| **Approach** | **k=1** | **k=5** | **k=30** | **k=100** | **k=1** | **k=5** | **k=30** | **k=100** |
| Lexical Retrieval (BM25) | | | | | | | | |
| BM25 baseline | 50.50 | 55.18 | 72.43 | 78.36 | 38.11 | 43.11 | 61.48 | 69.43 |
| **BM25 + pre-processing** | **57.50** | **62.19** | **79.86** | **85.14** | **45.57** | **51.47** | **71.78** | **79.25** |
| Semantic Retrieval (FAISS + cosine) | | | | | | | | |
| INF-Retriever-v1 | 58.66 | 65.21 | 86.50 | 92.04 | 47.23 | 54.48 | 79.88 | 87.28 |
| **INF-Retriever-v1 + FT** | **60.86** | **67.19** | **87.86** | **93.12** | **49.93** | **56.72** | **81.95** | **89.21** |
| Entire Pipeline | | | | | | | | |
| Elasticsearch with RRF | 63.72 | 69.35 | - | - | - | - | - | - |
| **Pipeline w. re-ranking** | **71.92** | **76.46** | - | - | **60.37** | **66.43** | - | - |

## 4. Results

The CLEF CheckThat! 2025 Subtask 4b evaluates systems using *mean reciprocal rank at* 5 *(MRR@5)*, which reflects how highly the correct source is ranked among the top five retrieved documents. Since MRR@5 is sensitive to ranking order, we prioritize optimizing the lexical and semantic retrievers for *precision*. Unlike MRR, Precision@*k* measures the proportion of relevant documents in the top-*k* results regardless of their order, ensuring that each retrieval stage yields high-quality candidate sets suitable for downstream re-ranking.

All experiments were conducted on the official datasets provided by the task organizers [5]. The corpus includes 7,718 documents. The development set comprises 1,400 queries, and the test set contains 1,446 queries. Our complete system achieves an MRR@5 score of 76.46% on the development set and 66.43% on the test set. Table 1 summarizes development and test set results across individual and combined retrieval stages. We evaluate MRR@1 and MRR@5, along with Precision@30 and Precision@100[1]. Although the absolute performance on the development set is generally higher than on the test set by approximately 10 percentage points, the relative gains achieved through our methods, such as pre-processing and fine-tuning, are consistent across both sets.

**Lexical Retrieval.** Our BM25 retriever with additional normalization and subword tokenization yields an 8.4-point gain in MRR@5 and a 10.3-point gain in Precision@30 over the official baseline on the test set, similar to the improvements observed on the development set (7.0 points in MRR@5, 7.4 points in Precision@30). Our preprocessing reduces noise and increases n-gram overlap, leading to better alignment between informal social media posts and formal scientific documents.

**Semantic Retrieval.** On the development set, employing `INF-Retriever-v1` yields an absolute improvement of 10.03 percentage points in MRR@5 over the BM25 baseline. Fine-tuning the retriever further increases MRR@5 by 1.98 points, reaching a final score of 67.19%. In terms of Precision@100, the base model achieves a 13.7-point gain compared to the BM25 baseline, with fine-tuning contributing an additional 1.1-point improvement. These gains are similar on the test set: `INF-Retriever-v1` improves MRR@5 by 11.4 points over the BM25 baseline, and fine-tuning adds a further 2.2-point gain,

[1]These metrics correspond to the best-performing configuration: BM25 returns the top 30 documents, and the semantic retriever contributes the top 100.

culminating in an MRR@5 of 56.72%. Precision@100 follows a similar trend, with respective gains of 17.85 and 1.93 percentage points. These consistent improvements across both development and test splits highlight the effectiveness and robustness of semantic retrieval, particularly when fine-tuning is applied. We also experimented with data augmentation techniques, including HyDE-generated queries and alternative document variants. However, these did not yield further gains. A discussion on data augmentation is provided in Appendix D.

**Re-Ranking.** Our complete pipeline with *bge-reranker-v2-gemma* as re-ranker achieves an MRR@5 of 76.46% on the development set, providing a +9.3 percentage points gain over our best individual retrieval method. To isolate the effectiveness of our re-ranking approach, we compare it against a hybrid baseline using Elasticsearch (see Appendix E for implementation details). Similar to our pipeline, this baseline uses BM25 for keyword search and the fine-tuned embedding model for semantic search, followed by reciprocal rank fusion (RRF) for re-ranking [32]. Although RRF provides a small boost over standalone retrieval (+2.2 percentage points), it underperforms the cross-encoder by 7.1 percentage points, highlighting the added value of learning-to-rank methods.

On the test set, our pipeline achieves 66.43% MRR@5, with the re-ranker achieving similar improvements of +9.7 percentage points over the best individual retrieval method, confirming that these gains generalize across datasets.

## 5. Discussion and Conclusion

In this paper, we presented a hybrid retrieval pipeline for attributing scientific sources to social media claims. Our system combines BM25 retrieval, dense semantic search with a fine-tuned encoder, and LLM-based cross-encoder re-ranking. Our results on subtask 4b of the CLEF CheckThat! 2025 competition demonstrate the effectiveness of this architecture: We ranked **1st** on the development set and **3rd** on the test set. Key findings include:

1. **Hybrid retrieval is essential:** Neither lexical nor semantic retrieval alone was sufficient. BM25 reached MRR@5 of 51.47%; the fine-tuned semantic retriever achieved 56.72%. Applying a cross-encoder to re-rank top candidates increased MRR@5 to 66.43% (a 23.3 percentage point improvement over the baseline), confirming the benefit of hybrid retrieval followed by learned ranking.
2. **In-domain fine-tuning improves performance:** Fine-tuning the dense retriever improved MRR@5 by approx. +2 percentage points and led to a Precision@100 of 89.21%. While pre-trained models perform well out of the box, domain adaptation further improves alignment between informal queries and scientific abstracts.
3. **Engineering matters:** Achieving these results required substantial engineering and experimentation efforts. We optimized hyperparameters, evaluated multiple data augmentation strategies (Appendices A and D), and evaluated alternative re-ranking models (Appendix C).

Despite the focus on CLEF's benchmark task, the proposed architecture is designed with broader applicability in mind. All components are modular and utilize open-source models, eliminating reliance on commercial APIs, thereby enabling deployment on local infrastructure [29]. This ensures compatibility with privacy-sensitive or offline environments and facilitates customization.

**Limitations and Future Work.** The current pipeline does not incorporate document-level metadata, such as author names, publication venues, or timestamps, which could improve retrieval precision and disambiguation. In addition, we do not integrate external resources such as web search engines or large-scale knowledge bases. Future work could explore metadata-aware retrieval and web-based search strategies to further enhance retrieval performance.

## Acknowledgments

We thank Prof. Dr. Simon Clematide and Andrianos Michail for guiding the research, engineering, and writing process. We also thank the Department of Computational Linguistics at the University of Zurich and the Centre for Artificial Intelligence at the Zurich University of Applied Sciences for providing computational resources.

## Declaration on Generative AI

During the creation of this work, the authors used ChatGPT[2] to refine the pre-written text. Further, the authors used Grammarly[3] for spell checking. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

[1] M. Brüggemann, I. Lörcher, S. Walter, Post-normal science communication: Exploring the blurring boundaries of science and journalism, Journal of Science Communication 19 (2020) A02. doi:10.22323/2.19030202.

[2] F. Alam, S. Shaar, F. Dalvi, H. Sajjad, A. Nikolov, H. Mubarak, et al., Fighting the COVID-19 Infodemic: Modeling the Perspective of Journalists, Fact-Checkers, Social Media Platforms, Policy Makers, and the Society, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 611–649. doi:10.18653/v1/2021.findings-emnlp.56.

[3] F. Alam, J. M. Struß, T. Chakraborty, S. Dietze, S. Hafid, K. Korre, A. Muti, P. Nakov, F. Ruggeri, S. Schellhammer, V. Setty, M. Sundriyal, K. Todorov, V. V., The CLEF-2025 CheckThat! Lab: Subjectivity, Fact-Checking, Claim Normalization, and Retrieval, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2025, pp. 467–478.

[4] F. Alam, J. M. Struß, T. Chakraborty, S. Dietze, S. Hafid, K. Korre, A. Muti, P. Nakov, F. Ruggeri, S. Schellhammer, V. Setty, M. Sundriyal, K. Todorov, V. Venktesh, Overview of the CLEF-2025 CheckThat! Lab: Subjectivity, Fact-Checking, Claim Normalization, and Retrieval, in: J. Carrillo-de Albornoz, J. Gonzalo, L. Plaza, A. García Seco de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), 2025.

[5] S. Hafid, Y. S. Kartal, S. Schellhammer, K. Boland, D. Dimitrov, S. Bringay, K. Todorov, S. Dietze, Overview of the CLEF-2025 CheckThat! Lab Task 4 on Scientific Web Discourse, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 2025.

[6] S. E. Robertson, K. S. Jones, Relevance weighting of search terms, Journal of the American Society for Information Science 27 (1976) 129–146. doi:10.1002/asi.4630270302.

[7] S. Robertson, H. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389. doi:10.1561/1500000019.

[8] J. Johnson, M. Douze, H. Jegou, Billion-Scale Similarity Search with GPUs, IEEE Transactions on Big Data 7 (2021) 535–547. doi:10.1109/TBDATA.2019.2921572.

[9] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, et al., The Faiss library, 2024. doi:10.48550/ARXIV.2401.08281, version Number: 3.

[10] J. Yang, J. Wan, Y. Yao, W. Chu, Y. Xu, et al., inf-retriever-v1 (2025). URL: https://huggingface.co/infly/inf-retriever-v1. doi:10.57967/HF/4262.

[2] https://chatgpt.com

[3] https://www.grammarly.com

[11] C. Li, Z. Liu, S. Xiao, Y. Shao, Making Large Language Models A Better Foundation For Dense Retrieval, 2023. doi:10.48550/ARXIV.2312.15503, version Number: 1.

[12] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation, in: Findings of the Association for Computational Linguistics ACL 2024, Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 2024, pp. 2318–2335. doi:10.18653/v1/2024.findings-acl.137.

[13] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a Large-scale Dataset for Fact Extraction and VERification, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 809–819. doi:10.18653/v1/N18-1074.

[14] A. Vlachos, S. Riedel, Fact Checking: Task definition and dataset construction, in: Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, Association for Computational Linguistics, Baltimore, MD, USA, 2014, pp. 18–22. doi:10.3115/v1/W14-2508.

[15] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. Van Zuylen, A. Cohan, et al., Fact or Fiction: Verifying Scientific Claims, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 7534–7550. doi:10.18653/v1/2020.emnlp-main.609.

[16] D. Wadden, K. Lo, B. Kuehl, A. Cohan, I. Beltagy, L. L. Wang, et al., SciFact-Open: Towards open-domain scientific claim verification, in: Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 4719–4734. doi:10.18653/v1/2022.findings-emnlp.347.

[17] A. Barrón-Cedeño, F. Alam, A. Galassi, G. Da San Martino, P. Nakov, T. Elsayed, et al., Overview of the CLEF–2023 CheckThat! Lab on Checkworthiness, Subjectivity, Political Bias, Factuality, and Authority of News Articles and Their Source, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, volume 14163, Springer Nature Switzerland, Cham, 2023, pp. 251–275. doi:10.1007/978-3-031-42448-9_20, series Title: Lecture Notes in Computer Science.

[18] A. Barrón-Cedeño, F. Alam, J. M. Struß, P. Nakov, T. Chakraborty, T. Elsayed, et al., Overview of the CLEF-2024 CheckThat! Lab: Check-Worthiness, Subjectivity, Persuasion, Roles, Authorities, and Adversarial Robustness, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, volume 14959, Springer Nature Switzerland, Cham, 2024, pp. 28–52. doi:10.1007/978-3-031-71908-0_2, series Title: Lecture Notes in Computer Science.

[19] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, et al., Dense Passage Retrieval for Open-Domain Question Answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. doi:10.18653/v1/2020.emnlp-main.550.

[20] G. Izacard, E. Grave, Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 874–880. doi:10.18653/v1/2021.eacl-main.74.

[21] J. Lee, M. Sung, J. Kang, D. Chen, Learning Dense Representations of Phrases at Scale, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 6634–6647. doi:10.18653/v1/2021.acl-long.518.

[22] J. Maillard, V. Karpukhin, F. Petroni, W.-t. Yih, B. Oguz, V. Stoyanov, et al., Multi-Task Retrieval for Knowledge-Intensive Tasks, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 1098–1111. doi:10.18653/v1/2021.acl-long.89.

[23] R. Nogueira, K. Cho, Passage Re-ranking with BERT, 2019. doi:10.48550/ARXIV.1901.04085,

version Number: 5.

[24] R. Sennrich, B. Haddow, A. Birch, Neural Machine Translation of Rare Words with Subword Units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1715–1725. doi:10.18653/v1/P16-1162.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is All you Need, in: Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[26] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, et al., Qwen2 Technical Report, 2024. doi:10.48550/arXiv.2407.10671, arXiv:2407.10671 [cs].

[27] M. Henderson, R. Al-Rfou, B. Strope, Y.-h. Sung, L. Lukacs, R. Guo, et al., Efficient Natural Language Response Suggestion for Smart Reply, 2017. doi:10.48550/arXiv.1705.00652, arXiv:1705.00652 [cs].

[28] E. J. Hu, y. shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, et al., LoRA: Low-Rank Adaptation of Large Language Models, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[29] L. Tuggener, P. Sager, Y. Taoudi-Benchekroun, B. F. Grewe, T. Stadelmann, So you want your private LLM at home? A survey and benchmark of methods for efficient GPTs, in: 2024 11th IEEE Swiss Conference on Data Science (SDS), IEEE, Zurich, Switzerland, 2024, pp. 205–212. doi:10.1109/SDS60720.2024.00036.

[30] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[31] Gemma Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, et al., Gemma 2: Improving Open Language Models at a Practical Size, 2024. URL: https://arxiv.org/abs/2408.00118. doi:10.48550/ARXIV.2408.00118, version Number: 3.

[32] G. V. Cormack, C. L. A. Clarke, S. Buettcher, 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, ACM, Boston MA USA, 2009, pp. 758–759. doi:10.1145/1571941.1572114.

[33] Gemma Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, L. Rouillard, T. Mesnard, G. Cideron, J. bastien Grill, S. Ramos, E. Yvinec, M. Casbon, E. Pot, I. Penchev, G. Liu, F. Visin, K. Kenealy, L. Beyer, X. Zhai, A. Tsitsulin, R. Busa-Fekete, A. Feng, N. Sachdeva, B. Coleman, Y. Gao, B. Mustafa, I. Barr, E. Parisotto, D. Tian, M. Eyal, C. Cherry, J.-T. Peter, D. Sinopalnikov, S. Bhupatiraju, R. Agarwal, M. Kazemi, D. Malkin, R. Kumar, D. Vilar, I. Brusilovsky, J. Luo, A. Steiner, A. Friesen, A. Sharma, A. Sharma, A. M. Gilady, A. Goedeckemeyer, A. Saade, A. Feng, A. Kolesnikov, A. Bendebury, A. Abdagic, A. Vadi, A. György, A. S. Pinto, A. Das, A. Bapna, A. Miech, A. Yang, A. Paterson, A. Shenoy, A. Chakrabarti, B. Piot, B. Wu, B. Shahriari, B. Petrini, C. Chen, C. L. Lan, C. A. Choquette-Choo, C. Carey, C. Brick, D. Deutsch, D. Eisenbud, D. Cattle, D. Cheng, D. Paparas, D. S. Sreepathihalli, D. Reid, D. Tran, D. Zelle, E. Noland, E. Huizenga, E. Kharitonov, F. Liu, G. Amirkhanyan, G. Cameron, H. Hashemi, H. Klimczak-Plucińska, H. Singh, H. Mehta, H. T. Lehri, H. Hazimeh, I. Ballantyne, I. Szpektor, I. Nardini, J. Pouget-Abadie, J. Chan, J. Stanton, J. Wieting, J. Lai, J. Orbay, J. Fernandez, J. Newlan, J. yeong Ji, J. Singh, K. Black, K. Yu, K. Hui, K. Vodrahalli, K. Greff, L. Qiu, M. Valentine, M. Coelho, M. Ritter, M. Hoffman, M. Watson, M. Chaturvedi, M. Moynihan, M. Ma, N. Babar, N. Noy, N. Byrd, N. Roy, N. Momchev, N. Chauhan, N. Sachdeva, O. Bunyan, P. Botarda, P. Caron, P. K. Rubenstein, P. Culliton, P. Schmid, P. G. Sessa, P. Xu, P. Stanczyk, P. Tafti, R. Shivanna, R. Wu, R. Pan, R. Rokni, R. Willoughby, R. Vallu, R. Mullins, S. Jerome, S. Smoot, S. Girgin, S. Iqbal, S. Reddy, S. Sheth, S. Põder, S. Bhatnagar, S. R. Panyam, S. Eiger, S. Zhang, T. Liu, T. Yacovone, T. Liechty, U. Kalra, U. Evci, V. Misra, V. Roseberry, V. Feinberg, V. Kolesnikov, W. Han, W. Kwon, X. Chen, Y. Chow, Y. Zhu, Z. Wei, Z. Egyed, V. Cotruta, M. Giang, P. Kirk, A. Rao, K. Black, N. Babar, J. Lo, E. Moreira, L. G. Martins, O. Sanseviero, L. Gonzalez, Z. Gleicher, T. Warkentin, V. Mirrokni,

E. Senter, E. Collins, J. Barral, Z. Ghahramani, R. Hadsell, Y. Matias, D. Sculley, S. Petrov, N. Fiedel, N. Shazeer, O. Vinyals, J. Dean, D. Hassabis, K. Kavukcuoglu, C. Farabet, E. Buchatskaya, J.-B. Alayrac, R. Anil, Dmitry, Lepikhin, S. Borgeaud, O. Bachem, A. Joulin, A. Andreev, C. Hardin, R. Dadashi, L. Hussenot, Gemma 3 technical report, 2025. doi:10.48550/arXiv.2503.19786.

[34] A. Shakir, D. Koenig, J. Lipp, S. Lee, Boost Your Search With The Crispy Mixedbread Rerank Models, 2024. URL: https://www.mixedbread.ai/blog/mxbai-rerank-v1.

[35] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, C-pack: Packaged resources to advance general chinese embedding, 2023. arXiv:2309.07597.

[36] L. Gao, X. Ma, J. Lin, J. Callan, 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), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 1762–1777. doi:10.18653/v1/2023.acl-long.99.

[37] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, et al., The Llama 3 Herd of Models, 2024. doi:10.48550/ARXIV.2407.21783, version Number: 3.

## A. Additional Experiments on Lexical Retrieval

**Table 2**
Comparison of lexical retrieval approaches with optional query augmentation on 1,000 samples from the training set. Preprocessing includes lowercasing, punctuation removal, and subword tokenization.

| | Precision@k (%) | | | | |
|---|---|---|---|---|---|
| **Approach** | **k=1** | **k=5** | **k=10** | **k=15** | **k=20** |
| BM25 Baseline | 49.90 | 60.50 | 64.70 | 67.10 | 69.70 |
| BM25 + Preprocessing | 56.00 | 70.10 | 74.30 | 76.80 | 78.10 |
| Query Expansion + Preprocessing | 56.80 | **71.50** | **76.80** | **79.00** | **80.60** |
| Query Rewriting + Preprocessing | **57.80** | 70.90 | 75.10 | 77.50 | 79.60 |

To explore potential improvements in lexical retrieval, we experimented with two query reformulation strategies using the Gemma3 12B language model [33]. These methods aim to reduce the linguistic mismatch between informal social media posts and formal scientific abstracts.

1. **Query Rewriting**: Reformulating social media posts to correct grammar and match the formal language style of scientific abstracts while preserving the original query semantics (see Listing 1).
2. **Query Expansion**: Augmenting the original social media post with 2-3 contextually relevant sentences to increase n-gram overlap with scientific abstracts (see Listing 2).

Among the evaluated methods, query expansion yielded the highest performance, achieving a Precision@20 of 80.6%, an improvement of 2.5 percentage points over BM25 with preprocessing. Query rewriting also led to performance gains, with a Precision@20 of 79.6% (a 1.5 percentage point improvement).

However, both methods incur significant computational overhead due to the reliance on a large language model. Specifically, inference time increased by approximately a factor of 60. Furthermore, given that our pipeline includes a subsequent re-ranking stage, the marginal precision gains from these query reformulations diminish in the final results of the entire pipeline. This unfavorable cost-benefit trade-off renders these methods impractical for integration into the final pipeline, so we excluded them.

```
Translate informal text into precise academic language, preserving
original meaning.

Transformation Guidelines:
- Correct the original tweet's spelling and grammar errors while
    maintaining its style
- Convert colloquial language to precise academic terminology
- Convert hastags into proper words
- Do not add anything new. Only correct the mistakes in the original
    tweet.

Output format:
Return a single string

Example:
Original Tweet: "Just saw amazin new study - mice w/ #Alzheimers showed
45% improvemnt in memory after new drug treatment!! Game changer for
#neurodegeneration research imo"

Output:
Just saw amazing new study - mice with Alzheimers showed 45% improvement
in memory after new drug treatment!! Game changer for
neurodegeneration research in my opinion

Transform the following tweet:
{tweet}
```

Listing 1: Prompt template to rewrite a social media post.

```
Translate informal text into precise academic language, according to the
transformation guidelines.

Transformation Guidelines:

First, correct the original tweet's spelling and grammar errors while
maintaining its style. Then transform the tweet into academic language
using these rules:

-Convert colloquial language to precise academic terminology
-Maintain semantic accuracy of the original message
-Use passive voice and objective scientific tone
-Eliminate informal expressions and subjective qualifiers
-Transform hashtags into their full, proper form (e.g., "#COVID19" -> "
    COVID-19 pandemic")
-Expand abbreviations and acronyms to their full forms
-Include key research terms that would appear in academic database
    searches
-Preserve all factual claims, statistics, and findings mentioned
-Structure as a concise academic abstract (2-3 sentences)

Output format:
Return a single continuous string with both versions separated by " || "
     as follows:
[Corrected Tweet] || [Academic Version]

Example:
Original Tweet: "Just saw amazin new study - mice w/ #Alzheimers showed
45% improvemnt in memory after new drug treatment!! Game changer for
#neurodegeneration research imo"

Output:
"Just saw amazing new study - mice with #Alzheimers showed 45%
improvement in memory after new drug treatment!! Game changer for
#neurodegeneration research in my opinion || A recent pharmacological
intervention demonstrated significant efficacy in an Alzheimer's
disease mouse model, with subjects exhibiting a 45% improvement in
memory function following administration of the novel compound.
These findings represent a potentially significant advancement in
neurodegenerative disease research, particularly regarding
therapeutic approaches for memory deficit amelioration in Alzheimer's
pathology."

Transform the following tweet:
{tweet}
```

Listing 2: Prompt template to expand the social media post.

## B. Embedding Model Fine-Tuning Details

To fine-tune the semantic embedding model, we initialize from `INF-Retriever-v1` [10], a transformer encoder pre-trained for dense retrieval tasks. Fine-tuning is performed by applying low-rank adaptation (LoRA) [28] to the query and value projection layers of the self-attention modules in the top 8 transformer layers (layers 20–27) using a rank $r = 8$, scaling $\alpha = 32$, and dropout of 0.1. The inputs are tokenized independently for queries (social media posts) and documents (title + abstract). The maximum sequence length is $8,192$ tokens, allowing for the processing of social media posts and documents without

truncation.

We use the multiple negatives ranking (MNR) loss [27]. Given a batch of $N$ query–document pairs, each query is trained to score highest on its corresponding document, while all other $N-1$ documents in the batch act as negatives. We extract embeddings using *last-token pooling*, which selects the hidden state of the final token in each sequence. Embeddings are $L_2$-normalized, and cosine similarity is computed via dot product.

We use AdamW [30] as optimizer with a learning rate of $1 \times 10^{-5}$. We use 20 linear warmup steps and then decay the learning rate to 0 using a cosine scheduler. We train on 2 A-100 GPUs using DDP with a per-device batch size of 4 and 16 gradient accumulation steps (resulting in an effective batch size of 64). We use gradient clipping with norm = 1.0 and use FP16 mixed precision. We evaluate *retrieval quality* (i.e., run the vector store) on the development set after each epoch by measuring Precision@100. The final model checkpoint is selected based on the best performance, which is obtained after epoch 2.

## C. Comparison of Re-Ranking Models

**Table 3**
Performance comparison of different re-ranking models using top 50 semantic retrieval candidates on the development set. The chosen model and best scores are displayed in bold.

| Re-ranking Model | MRR@1 (%) | MRR@5 (%) |
|---|---|---|
| *Baseline* | | |
| Semantic Retrieval | 61.50 | 67.56 |
| *Traditional Cross-Encoders* | | |
| mxbai-rerank-large-v2 | 61.43 | 66.80 |
| bge-reranker-large | 62.50 | 67.54 |
| *LLM-based Cross-Encoders* | | |
| **bge-reranker-v2-gemma** | 71.36 | **76.03** |
| bge-reranker-v2-minicpm [Layer 28] | 71.57 | 75.87 |
| bge-reranker-v2-minicpm [Layer 32] | **71.79** | 76.02 |

We conducted a comparative evaluation of various re-ranking models on the development set to identify the most effective approach for our retrieval pipeline. The evaluated re-ranking models include traditional cross-encoders (`mxbai-rerank-large-v2` [34], bge-reranker-large [35]) and LLM-based re-rankers (`bge-reranker-v2-gemma` [11, 12], `bge-reranker-v2-minicpm` [11, 12]), which use pre-trained language models as base for relevance scoring. The `bge-reranker-v2-minicpm` model supports layer-wise inference optimization, allowing computation to terminate at intermediate layers rather than processing through the full network. We experimented with two different intermediate layer configurations, terminating after layer 28 and layer 32. We selected layer 32 based on our preliminary experiment with 100 samples across all available layers, which showed that layer 32 achieved the best performance. Additionally, we included layer 28, as this is recommended by the official BGE re-ranker repository. All re-rankers are evaluated on the development set using semantic retrieval candidates as input. As shown in Table 3, LLM-based re-rankers outperformed traditional cross-encoders by a considerable margin. This performance gap likely stems from LLMs' extensive pre-training on diverse text corpora, enabling them to comprehend both formal and informal language patterns. Between the three LLM-based re-rankers, `bge-reranker-v2-gemma` achieved the best MRR@5 performance (76.03% vs. 76.02% and 75.87 %). Although the margin is small, we selected `BAAI/bge-reranker-v2-gemma` as our final model.

## D. Data Augmentation for Semantic Retrieval

To enrich semantic retrieval, we experimented with two text augmentation strategies: hypothetical document embeddings (HyDE) [36] and additional documents (AD). Both methods leverage the Llama 3.2 7B model [37] to generate auxiliary text representations.

For HyDE, we prompted the model to generate a hypothetical scientific article (title and abstract) based on a given social media post, aiming to bridge the domain gap between informal social media language and formal scientific discourse (see Listing 3). For AD, we augmented the document corpus by generating (1) a summary and (2) a synthetic social media post for each document. These variants were stored alongside the original document in the vector index (Listings 4 and 5).

```
You are an expert in scientific research. Based on the following tweet,
generate a hypothetical scientific paper that includes only a title and
an abstract. The abstract should succinctly summarize the research
objective, methodology,  key findings, and conclusions.

Tweet: {tweet}

{format_instructions}
```

Listing 3: Prompt template to generate hypothetical document embeddings.

```
Summarize the following document:

Title: {title}
Abstract: {page_content}

Make sure to include keywords that are likely to be found later by a
search.

{format_instructions}
```

Listing 4: Summary Prompt Template for AD

```
Generate a hypothetical Twitter tweet about the following document:

Title: {title}
Abstract: {page_content}

Make sure it looks like a typical tweet from an average person and is
not too long.

{format_instructions}
```

Listing 5: Tweet Prompt template to generate additional documents.

The results on the development set are displayed Table 4. As discussed in Section 4, our primary objective for semantic retrieval is to ensure high precision, providing strong candidates for downstream re-ranking. We find that augmentation strategies offer modest improvements for off-the-shelf models but yield limited or no benefit when applied to the fine-tuned retriever. We hypothesize that the limited benefit observed from these augmentation methods stems from the fine-tuned model's already high semantic fidelity, which reduces the marginal gains achievable through additional data augmentation. Therefore, these methods were excluded from the final pipeline.

**Table 4**
Detailed performance comparison on the development set of semantic retrieval with additional text pre-processing. "AD" denotes additional documents, "HyDE" stands for hypothetical document embeddings. The best results per category are in bold.

| | MRR@k (%) | | Precision@k (%) | |
|---|---|---|---|---|
| **Approach** | **k=1** | **k=5** | **k=30** | **k=100** |
| INF-Retriever-v1 | 58.66 | 65.21 | **86.50** | **92.04** |
| + HyDE | 48.89 | 56.06 | 80.37 | **87.44** |
| + AD | **60.73** | **66.69** | **87.09** | 90.87 |
| + HyDE + AD | 51.63 | 58.46 | 81.56 | 85.36 |
| INF-Retriever-v1 + Fine-tuning | 60.86 | **67.19** | **87.86** | **93.12** |
| + HyDE | 51.53 | 58.79 | 83.01 | 89.60 |
| + AD | **61.88** | **67.89** | **88.49** | 91.50 |
| + HyDE + AD | 53.44 | 60.56 | 83.60 | 87.32 |

# E. Hybrid Search using Elasticsearch

In addition to our main retrieval pipeline, we explored a fully integrated alternative using Elasticsearch[4]. This system unifies indexing, retrieval, and ranking into a single framework, while still capturing both lexical and semantic signals.

We build an Elasticsearch pipeline closely mirroring our original architecture: it incorporates (1) a BM25 retriever with fuzzy matching, (2) a $k$-nearest neighbor (kNN) semantic retriever using our fine-tuned embedding model (configured with $k = 50$ and 200 candidates), and (3) a fusion stage based on reciprocal rank fusion (RRF) to combine results [32]. The RRF configuration uses a window size of 100 and a rank constant of 20, allowing it to integrate signals from both retrieval branches efficiently. Unlike our main system, which uses a cross-encoder for deep re-ranking, the Elasticsearch pipeline relies on this lightweight re-scoring mechanism.

**Table 5**
Elasticsearch (ES) hybrid pipeline results on the development set. "AD" denotes additional documents, "HyDE" stands for hypothetical document embeddings. The best results are displayed in bold.

| Configuration | MRR@1 (%) | MRR@5 (%) |
|---|---|---|
| ES | 60.53 | 66.69 |
| ES + HyDE | 45.99 | 53.56 |
| ES + AD | **63.72** | **69.35** |
| ES + AD + HyDE | 51.65 | 58.55 |

**Performance.** Similar to the evaluation of our custom pipeline described in Appendix D, we evaluate different variants of this Elasticsearch—based method, leveraging raw and extended documents, as well as with and without query expansion. Table 5 presents the results obtained on the development set.

The best Elasticsearch configuration (ES + AD) achieves MRR@5 of 69.35%, slightly superior to our custom pipeline's semantic retriever. However, it lags behind the full system with cross-encoder re-ranking (MRR@5 = 76.46%). This highlights the benefit of contextual re-scoring for fine-grained relevance. Nonetheless, the Elasticsearch-based approach remains a viable, scalable option for latency-sensitive applications.

[4]https://www.elastic.co