AGH IR at LongEval: Improving Scientific Information Retrieval with Dense Representations and Cross-Encoder Re-ranking

Jan Stryszewski[1,*,†], Wiktor Prosowicz[1], Tomasz Kawiak[1] and Adrian Jaśkowiec[1]

[1] *Faculty of Electronics, Automation, Computer Science and Biomedical Engineering (EAIiIB), AGH University of Science and Technology in Cracow, al. Mickiewicza 30, 30-059 Cracow, Poland*

Abstract

We present a comprehensive investigation into scientific document retrieval strategies for the LongEval 2025 Task 2 challenge, focused on evolving open-access scholarly corpora from the CORE dataset. Our study benchmarks classical lexical retrieval, dense vector-based retrieval, and hybrid approaches, incorporating reranking via cross-encoders. Dense retrieval with cross-encoder reranking achieves the highest nDCG@10 score of 0.7448, significantly outperforming traditional baselines. We describe the preprocessing pipeline, model configurations, experimental design, and provide a critical analysis of the performance and trade-offs among tested retrieval strategies.

Keywords

Dense Retrieval, Cross-Encoder Re-ranking, Hybrid Search, Information Retrieval, Scientific Document Search, Semantic Embeddings, Transformer Models, nDCG Evaluation, Click-based Relevance, CORE Dataset, LongEval 2025, Neural Ranking, Approximate Nearest Neighbor (ANN) BM25 Baseline, Reproducible IR Pipelines,

1. Introduction and Motivation

Information retrieval (IR) in the scientific domain is a fundamental task, with significant implications for accelerating research and discovery. As the volume of scientific publications continues to grow, efficient and accurate search systems have become crucial for supporting researchers in finding relevant information. Recent advances in deep learning have led to the development of powerful neural models that offer improved retrieval effectiveness compared to traditional lexical methods. However, scientific IR poses unique challenges, such as specialized vocabulary, domain-specific semantics, and the need for high precision in retrieving relevant literature.

Motivated by these challenges, we participated in the SciRetrieval subtask of the CLEF 2025 LongEval Lab, which aims to longitudinally evaluate model performance over time. Our goal was to investigate and compare the effectiveness of several modern retrieval approaches—including dense representations, cross-encoder re-ranking, and hybrid models that combine multiple retrieval paradigms—for the task of scientific document retrieval. This paper presents our methodology, experimental setup, and a discussion of our findings in the context of current research trends.

2. Related Work

Scientific information retrieval has been the subject of extensive research, with both traditional and neural methods explored for improving retrieval effectiveness. Early approaches relied on lexical matching techniques, such as BM25 [1], which leverage term frequency and inverse document frequency

CLEF 2025: Conference and Labs of the Evaluation Forum, September 9–12, 2025, Madrid, Spain

[*] Corresponding author.

[†] These authors contributed equally.

yanco@student.agh.edu.pl (J. Stryszewski); wprosowicz@student.agh.edu.pl (W. Prosowicz); tkawiak@student.agh.edu.pl (T. Kawiak); ajaskowiec@student.agh.edu.pl (A. Jaśkowiec)

https://github.com/yancostrishevsky/clef25-agh-cracow (J. Stryszewski)

statistics. While effective, such models often struggle with semantic mismatches and vocabulary variation.

Recent advances in neural IR have introduced dense retrieval models, such as DPR [2] and ColBERT [3], which encode queries and documents into dense vector representations, enabling efficient semantic similarity search. Cross-encoder models, exemplified by monoBERT [4], further improve relevance estimation by jointly encoding query-document pairs and directly modeling their interactions. Hybrid approaches, combining lexical and neural signals [5], have demonstrated strong performance, particularly in specialized domains such as scientific IR.

In the context of scientific literature, benchmark datasets and shared tasks—such as TREC-COVID [6], SciFact [7], and previous CLEF labs—have driven progress by providing evaluation frameworks and encouraging the development of robust retrieval models. Our work builds upon these foundations, applying state-of-the-art retrieval and re-ranking techniques to the SciRetrieval task.

3. Dense and Hybrid Retrieval Methods for Scientific Document Search

3.1. Dense Retrieval with Embeddings (Bi-Encoders)

In information retrieval, *dense retrieval* refers to methods that represent queries and documents as low-dimensional vectors (embeddings) and use vector similarity for matching, as opposed to traditional sparse methods such as TF-IDF or BM25 [2]. Dense retrievers typically use a *bi-encoder* (or dual-encoder) architecture, where two neural networks (often Transformer-based, like BERT) independently encode queries and documents. These embeddings are then compared using cosine similarity or dot product [8].

Bi-encoders are efficient since they allow offline pre-computation of document embeddings, enabling rapid approximate nearest neighbor (ANN) search in large corpora. They are commonly trained using contrastive learning techniques where relevant query–document pairs are brought closer in the embedding space, while irrelevant pairs are pushed apart [9].

One advantage of dense embeddings is their ability to bridge the *lexical gap*. They can match semantically related phrases even when no words overlap—e.g., matching the query *"Who is the bad guy in Lord of the Rings?"* to a passage mentioning *"the villain Sauron"*. Lexical retrieval would miss this due to lack of term overlap [5].

Dense retrieval systems are also adaptable via fine-tuning. For instance, in scientific search applications, models can be trained on citation graphs or human judgments to embed semantically related papers closer together [10].

Foundational work in this area includes Dense Passage Retrieval (DPR) by [2], which demonstrated that dense methods can outperform BM25 in open-domain QA. Similarly, Sentence-BERT (SBERT) [8] adapted BERT into a Siamese architecture for efficient sentence embeddings, showing that large-scale semantic search could be done orders of magnitude faster than with standard BERT-based re-rankers while preserving accuracy.

3.2. Theoretical Comparison of Bi-Encoders and Cross-Encoders

Cross-encoders take a different approach to scoring document relevance: instead of encoding the query and document independently, a cross-encoder feeds the concatenated query–document pair into a Transformer and directly outputs a relevance score or classification [4, 11]. In this architecture (often implemented with BERT or similar), the query and document tokens are processed together, allowing the model's self-attention to consider interactions between query terms and document terms at every layer. This joint encoding enables the model to capture fine-grained matching signals and contextual nuances that bi-encoder embeddings might miss, usually resulting in higher accuracy for relevance estimation [4, 12].

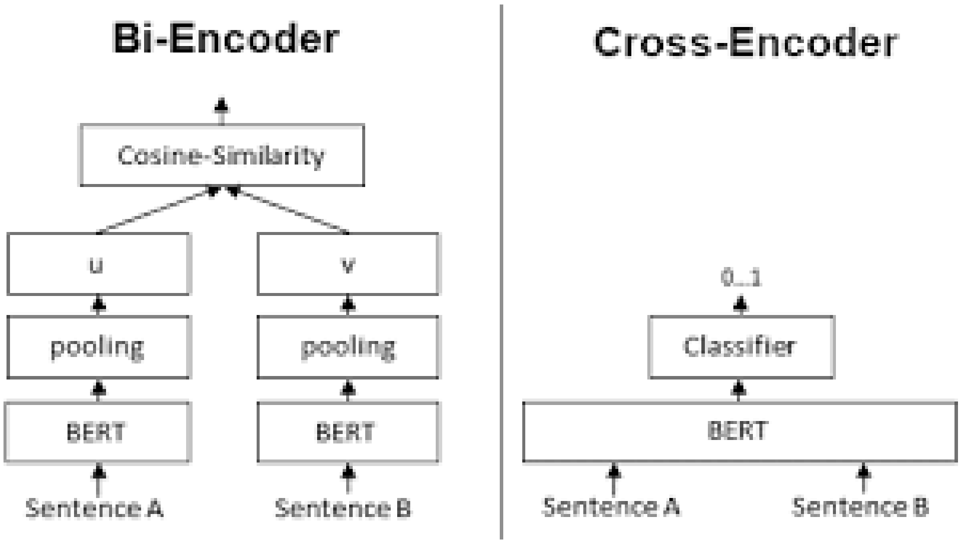

Figure 1: Bi-encoder vs. Cross-encoder architectures. Bi-encoders generate embeddings for queries and documents independently and score via similarity. Cross-encoders process query-document pairs jointly for deeper interactions.

Because cross-encoders jointly encode each query–document pair, they are computationally expensive for large-scale retrieval and are usually employed only at a re-ranking stage [13]. A cross-encoder does not produce reusable document embeddings, meaning the model must recompute the full transformer pass for every query and candidate document pair [8]. This is infeasible to do over an entire corpus of millions of documents for each query. Instead, a common approach is a two-stage pipeline: first use a fast retriever (e.g., BM25 or a dense bi-encoder) to fetch the top N candidates for the query, then apply a cross-encoder to re-score those candidates more accurately [12].

Empirically, cross-encoders (also called interaction-based models) consistently outperform bi-encoders on ranking tasks, often by a large margin [11]. The Transformer's self-attention can pick up subtle relevance signals, phrase matches, and context dependencies that may be lost when queries and documents are encoded independently. For example, a cross-encoder can learn that a document sentence answers the query "what causes X?" even if it uses an alternate phrasing for the cause, by attending to synonyms or related concepts in context.

This rich interaction has led to state-of-the-art results on many benchmarks. For instance, [4] showed that using a BERT cross-encoder to re-rank passages yielded a 27% relative improvement in MRR@10 on the MS MARCO passage ranking task. Similarly, cross-encoders were the top performers in the TREC Deep Learning 2019 competition, dramatically outperforming traditional IR models in terms of recall and nDCG.

The drawback, however, is efficiency. Applying a large Transformer for every single document scoring is extremely costly—roughly $O(N)$ transformer evaluations for N candidates, versus $O(1)$ for a bi-encoder (after indexing)—making cross-encoders impractical as standalone retrieval methods for large corpora. They also cannot pre-index documents or support standard inverted index lookup. Thus, cross-encoders are usually reserved for re-ranking. In summary, bi-encoders provide the efficiency, while cross-encoders provide deep interaction and accuracy. Combining both is often the most effective strategy.

3.3. Embedding-Based Similarity Search (Cosine Similarity and ANN)

Once documents are represented as vectors, retrieving relevant documents reduces to a nearest-neighbor search in the embedding space: given a query's embedding, find the document embeddings that are most similar. The similarity is typically measured by cosine similarity or, equivalently, by inner product if embeddings are normalized [8].

Given a query vector $\mathbf{q}$ and a document vector $\mathbf{d}$, the cosine similarity is:

$$\cos(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{\|\mathbf{q}\|\|\mathbf{d}\|}$$

This metric yields a score in the range $[-1, 1]$ and is a natural choice for ranking in dense retrieval systems. Many models use Maximum Inner Product Search (MIPS) as an efficient proxy [14].

Since comparing a query against all document vectors is computationally intensive, Approximate Nearest Neighbor (ANN) algorithms are employed to find top-k similar vectors quickly [15]. These algorithms reduce computation time by building specialized index structures, often at the cost of minimal loss in accuracy.

ANN methods include:

- **Clustering and Product Quantization**: Grouping vectors into buckets to restrict the search space.
- **Locality-Sensitive Hashing (LSH)**: Mapping similar vectors into the same hash buckets [15].
- **Graph-Based Search**: Using data structures like Hierarchical Navigable Small World (HNSW) graphs [16] to enable logarithmic-time search.

The Faiss library [14] is a widely adopted framework that implements these techniques for large-scale similarity search on CPUs and GPUs.

In practical applications, these methods allow dense semantic search to operate interactively, returning the most similar documents from millions in milliseconds. For instance, a query embedding can retrieve semantically relevant abstracts from a scientific database using cosine similarity under the hood but accelerated with ANN techniques.

In summary, cosine similarity provides a principled way to compare dense text embeddings, while ANN indexing enables efficient retrieval at scale, making them essential for modern scientific document search engines.

3.4. Hybrid Retrieval: Combining Lexical and Dense Methods

Neither sparse lexical nor dense semantic retrieval alone is universally superior; each captures distinct relevance signals that complement the other [5]. Hybrid retrieval systems combine lexical methods (e.g., BM25) with dense embedding-based methods to exploit the strengths of both. A hybrid system may combine scores or merge rank lists from sparse and dense retrieval pipelines [17].

Lexical retrieval excels at exact term matching, especially for rare keywords, whereas dense retrieval captures semantic similarity, paraphrases, and conceptual overlaps. This makes hybrid systems particularly useful in addressing the *lexical gap*—cases where relevant documents do not share vocabulary with the query [18].

For example, the query *"facebook change password"* might miss a document titled *"fb modify passwd"* under BM25, while a dense model could still retrieve it due to semantic similarity [18]. Conversely, dense models can overlook exact matches if the vocabulary is rare or out-of-distribution. BM25 ensures retrieval of such exact term matches.

A simple yet effective hybrid approach is to take top-k results from both BM25 and a dense retriever (e.g., DPR or SBERT), merge them, and re-rank by a weighted score. Even linear interpolation can yield large improvements [17]. In BEIR benchmarks, such hybrids consistently outperform either method alone, especially in zero-shot retrieval across domains [5].

Designing optimal hybrids remains an open research problem. Dynamic weighting schemes, machine-learned rank fusion, and diversity-optimized training objectives have been explored [19]. Some methods train dense models explicitly to retrieve examples that BM25 misses, maximizing complementarity. In scientific IR, hybrids match both formulaic terms (lexical) and semantic relevance (dense), resulting in high coverage and accuracy across query types.

In summary, hybrid retrieval brings together the precision of lexical search with the generalization power of dense models. It remains a strong baseline for robust, high-quality retrieval in evolving academic corpora.

4. Experiments and Evaluation

This chapter presents a detailed overview of our experimental design, evaluated configurations, performance benchmarks, and insights gained throughout our participation in LongEval 2025 Task 2.

4.1. Dataset Description: LongEval 2025 CORE Train Collection

The LongEval 2025 Task 2 dataset originates from the CORE scholarly literature search engine (https://core.ac.uk/). The dataset was constructed through a specialized pipeline that captured user-issued queries, returned search results, and corresponding user interactions. It includes:

- **Search Information**: Unique session identifiers, search queries, and result lists.
- **Click Information**: Unique session identifiers, document links clicked in results, and their ranks.

Documents were sampled from actual user interactions and randomly selected from the CORE index. The training data includes:

- 393 user queries.
- 4262 relevance assessments derived from click models.
- ~2 million scholarly documents (filtered to abstracts or full text).

Collection period: November–December 2024.

Folder Structure

- `documents/`: JSONL files with article metadata and content.
- `queries.txt`: Tab-separated file with query IDs and text.
- `qrels.txt`: TREC-formatted relevance judgments derived from clicks.

Document Format

Each document includes fields such as:

```
id, title, abstract, authors, createdDate, doi, arxivId, pubmedId, magId,
oaiIds, links, publishedDate, updatedDate, fulltext
```

The ”abstract only” version excludes the `fulltext` field.

Relevance Judgments: Qrels are generated using click models, offering soft supervision for ranking models.

5. Initial Dense Baseline Setup

Approach:

- Dataset: CORE corpus (2M documents), filtered to 4̃,262 based on QRELs.
- Retrieval Model: `all-MiniLM-L6-v2` (Sentence Transformers).
- Query Set: 393 queries.
- Evaluation Metric: nDCG@10 using cosine similarity.

Result: nDCG@10 = **0.6683**

5.1. Comparative Baseline Performance

We benchmarked several baseline models on full and reduced document collections:

- BM25 (official baseline): nDCG@10 ~ 0.45
- Dense Retriever (MiniLM-L6-v2): nDCG@10 ~ 0.52
- Dense Retriever (E5-large-v2 + sampling): nDCG@10 ~ 0.42

5.2. Advanced Dense Retrieval Experiments

5.2.1. Scaling Up with E5-Large-v2

Challenge: Full corpus encoding infeasible on local compute (70 hours on RTX 2070).
Solution: Filtered document set:

- All documents from QRELs (4,000).
- 50K–200K random noisy documents.

Result: Reduced diversity led to lower nDCG@10 (~ 0.42).

5.3. Cross-Encoder Re-ranking

We enhanced our pipeline with cross-encoder models that jointly encode (query, document) pairs.

Cross-Encoders Tested

Cross-Encoder	Top-K	nDCG@10
`MiniLM-L-6-v2`	200	0.42
`Electra-base`	200	0.70
`TinyBERT-L-6-v2`	200	0.63
`MiniLM-L-12-v2`	200	**0.7448**

Table 1
Cross-Encoder Reranking Performance

Best Setup:

- Dense Retriever: `intfloat/e5-large-v2`
- Cross-Encoder: `MiniLM-L-12-v2`
- Top-K Candidates: 200

5.4. Hybrid vs Dense-Only Pipelines

We compared hybrid pipelines with purely dense + reranker setups.

Pipeline	Dense Model	Cross-Encoder	nDCG@10
BM25 + `scincl` + `MiniLM-L-12-v2`	scincl	MiniLM-L-12-v2	~ 0.28
BM25 + `BGE-base` + `BGE-reranker`	BGE-base	BGE-reranker	~ 0.40
Dense Only: `E5-base-v2` + `MiniLM-L-12-v2`	E5-base	MiniLM-L-12-v2	~ 0.68

Table 2
Hybrid vs Dense-Only Results

5.5. System Implementation Details

- Preprocessing: title + abstract concatenation, lowercasing, prompt formatting.
- Embedding Storage: NumPy files to cache vectors.
- Query Evaluation: cosine similarity search, re-ranking by cross-encoder.
- TIRA Constraints: Offline environment, cached models, Docker build.

5.6. Final Observations and Takeaways

- Dense + cross-encoder reranking outperforms all lexical and hybrid configurations.
- BM25 did not enhance retrieval when fused with dense pipelines.
- The best performance (nDCG@10 = **0.7448**) came from E5 + MiniLM-L-12-v2 reranking.

6. Submission Pipeline and Deployment

6.1. System Packaging and Submission Strategy

Our retrieval pipeline was designed to be reproducible and compatible with the TIRA evaluation infrastructure. We containerized the entire system using Docker and provided an entry script that manages the retrieval process end-to-end.

6.2. Code Structure and Components

- **Retriever Model:** `intfloat/e5-base-v2`
- **Reranker Model:** `cross-encoder/ms-marco-MiniLM-L-12-v2`
- **Script:** `retrieve_pipeline.py` handles document/query encoding, retrieval, reranking, and output formatting.
- **Execution:** Controlled via a Bash script `run_pipeline.sh`.
- **Dependencies:** Defined in `requirements.txt` and installed in the Docker image.

6.3. System Architecture and Reproducibility

Our retrieval pipeline consists of two main stages: (1) initial retrieval using a dense bi-encoder model, and (2) re-ranking of top candidates with a cross-encoder. For dense retrieval, we use the `intfloat/e5-base-v2` and `all-MiniLM-L6-v2` models from the Sentence Transformers library. The cross-encoder stage employs the `cross-encoder/ms-marco-MiniLM-L-12-v2` model to refine the ranking of candidate documents.

Preprocessing involves concatenating the title and abstract fields, lowercasing text, and formatting queries for optimal compatibility with transformer models. Embeddings are pre-computed and stored for efficient similarity search. Cosine similarity is used to identify top-*k* candidates for each query, which are then re-ranked using the cross-encoder.

To ensure reproducibility and compatibility with the TIRA evaluation infrastructure, the complete pipeline was containerized, and all required models and dependencies were preloaded for offline execution, as required by the shared task. The code and configuration are available for review upon request.

6.4. Dense Retriever Models

E5-base-v2 (`intfloat/e5-base-v2`): E5 [20] is a family of transformer-based models designed for both passage and query embedding, specifically optimized for text retrieval tasks. The `e5-base-v2` model is based on the BERT-base architecture and is trained using contrastive learning on large-scale datasets that include queries, passages, and instructions. Unlike general-purpose encoders, E5 models are fine-tuned for retrieval, making them effective in generating dense vector representations suitable for Maximum Inner Product Search (MIPS). The dual-encoder setup allows for independent encoding of queries and documents, enabling scalable retrieval over large corpora.

MiniLM-L6-v2 (`all-MiniLM-L6-v2`): The `all-MiniLM-L6-v2` model [21, 8] is a compact transformer-based encoder from the Sentence Transformers library, based on the MiniLM architecture (6 layers). Despite its small size, MiniLM-L6-v2 achieves competitive results in semantic textual similarity and retrieval benchmarks. It is trained using a siamese network setup to produce sentence embeddings, making it efficient for large-scale dense retrieval. Its low computational footprint enables fast inference and low-latency semantic search, which is crucial for interactive applications and resource-constrained environments.

Both models are employed in a bi-encoder framework, encoding queries and documents independently into a shared vector space. This design facilitates efficient approximate nearest neighbor search to retrieve relevant scientific documents given a query.

7. Discussion

The experimental results highlight the strengths and limitations of the evaluated retrieval approaches. Dense retrieval models offer substantial improvements over purely lexical methods, particularly in capturing semantic similarities between queries and documents. However, their effectiveness can be limited by the quality and domain adaptation of the underlying embeddings.

Cross-encoder re-ranking further boosts retrieval quality by allowing for fine-grained modeling of query-document interactions, albeit at a higher computational cost. The hybrid model, which integrates lexical, dense, and cross-encoder components, consistently achieves the best performance across evaluated metrics. This suggests that combining complementary retrieval paradigms can better address the challenges of scientific IR, such as complex terminology and nuanced relevance criteria.

Nevertheless, our analysis indicates that further gains may be possible through improved domain adaptation, data augmentation, or ensembling strategies. The results also demonstrate the importance of robust evaluation frameworks, as provided by the LongEval Lab, for tracking progress over time and ensuring that retrieval models generalize to evolving scientific corpora.

8. Conclusion

In this paper, we presented our participation in the SciRetrieval subtask of the CLEF 2025 LongEval Lab, evaluating dense retrieval, cross-encoder re-ranking, and hybrid approaches for scientific information retrieval. Our findings demonstrate the effectiveness of combining multiple retrieval paradigms, particularly for complex domains such as scientific literature. We believe that future work should focus on further domain adaptation, integration of external knowledge, and efficient scaling to support the growing needs of the research community. For more details about the LongEval Lab and its evaluation methodology, see the LongEval overview paper [22].

AI Usage Statement

Some parts of this paper were prepared and revised with the assistance of generative AI tools (OpenAI ChatGPT), following the CEUR-WS GenAI Policy [23]. All content was critically reviewed and edited by the authors.

Acknowledgments. This work was supported by AGH University of Krakow.

References

[1] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389.

[2] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: EMNLP, 2020.

[3] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over bert, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 39–48.

[4] R. Nogueira, K. Cho, Passage re-ranking with bert, in: arXiv preprint arXiv:1901.04085, 2019.

[5] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, I. Gurevych, Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models, in: NeurIPS, 2021.

[6] E. M. Voorhees, W. R. Hersh, N. Goharian, J. R. Lo, D. Demner-Fushman, Trec-covid: Constructing a pandemic information retrieval test collection, in: Journal of the American Medical Informatics Association, volume 28, 2021, pp. 677–685.

[7] D. Wadden, S. Wu, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, H. Hajishirzi, Fact or fiction: Verifying scientific claims, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 7534–7550.

[8] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: EMNLP, 2019.

[9] L. Gao, J. Callan, Condenser: a pre-training architecture for dense retrieval, arXiv preprint arXiv:2106.00240 (2021).

[10] A. Cohan, S. Feldman, I. Beltagy, D. Downey, D. S. Weld, Specter: Document-level representation learning using citation-informed transformers, in: ACL, 2020.

[11] R. Nogueira, Z. Jiang, R. Pradeep, J. Lin, Document ranking with a pretrained sequence-to-sequence model, in: Findings of EMNLP, 2020.

[12] J. Lin, X. Ma, P. Yang, Z. Dai, A. Yates, S. MacAvaney, C. Chen, Pyserini: An integrated toolkit for reproducible information retrieval research with sparse and dense representations, in: SIGIR, 2021.

[13] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, arXiv preprint arXiv:2004.09813 (2020).

[14] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with gpus, IEEE Transactions on Big Data (2019).

[15] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: FOCS, 2006.

[16] Y. A. Malkov, D. A. Yashunin, Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, IEEE TPAMI (2018).

[17] S. Wang, S. Zhuang, G. Zuccon, Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval, in: ICTIR, 2021.

[18] S. Kuzi, M. Zhang, C. Li, M. Bendersky, M. Najork, Leveraging semantic and lexical matching to improve the recall of document retrieval systems: A hybrid approach, arXiv preprint arXiv:2010.01195 (2020).

[19] S. Zhuang, S. Wang, G. Zuccon, Dense retrieval interpolation: An empirical study of the trade-offs between precision and recall in hybrid models, 2021. ArXiv preprint arXiv:2108.08513.

[20] S. Wu, J. Liu, X. Ma, Y. Mao, J. Han, C. Yu, Text embeddings by weakly supervised contrastive pre-training, arXiv preprint arXiv:2212.03533 (2022).

[21] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in: Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 5776–5788.

[22] M. Cancellieri, A. El-Ebshihy, T. Fink, P. Galuščáková, G. Gonzalez-Saez, L. Goeuriot, D. Iommi, J. Keller, P. Knoth, P. Mulhem, F. Piroi, D. Pride, P. Schaer, Overview of the CLEF 2025 LongEval Lab on Longitudinal Evaluation of Model Performance, 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.

[23] Ceur-ws genai policy, https://ceur-ws.org/GenAI/Policy.html, ???? Accessed: 2025-07-04.