RETRIX at CheckThat! 2026: Source Retrieval for Scientific Web Claims

Notebook for the CheckThat! Lab at CLEF 2026

Fabio Baldan[1,*], Davide Donati[1], Alvise Garberino[1], Giancarlo Padoan[1], Marco Tessari[1] and Nicola Ferro[1]

[1]*University of Padua, Italy*

Abstract

We present the results of the RETRIX group for the CheckThat! Lab at CLEF 2026 Task 1, whose goal is to identify the scientific paper implicitly referenced in a social media post. The input consists of short, noisy, and usually informal posts that mention scientific findings without providing explicit links or citations. Our pipeline first translates non-English posts into English, then expands them using an embedding model and terms extracted from the top five documents ranked by similarity. We then apply a hybrid retrieval system that combines dense FAISS search with sparse lexical matching and ColBERT multi-vector scoring. Finally, we perform neural reranking with a fine-tuned cross-encoder model.

Keywords

Fact Checking, Scientific Document Retrieval, Social Media Fact Verification, Neural Reranking, CLEF, Information Retrieval, Cross Encoder

1. Introduction

The rapid spread of scientific claims through social media creates a difficult fact-checking scenario: social media posts often report, simplify or sensationalize research findings while omitting the original source. In many cases, the cited study is not linked explicitly with a URL or a bibliographic reference but only mentioned indirectly through fragments such as the reported finding, the venue, the institution or the authors. As a consequence, verifying such claims requires first identifying the scientific publication that is being implicitly referenced [1].

This problem is the focus of Task 1 of the CLEF 2026 CheckThat! Lab, *Source Retrieval for Scientific Web Claims* [2, 3]: given a social media post containing a scientific claim and an implicit reference to a paper, the goal is to retrieve the correct publication from a fixed collection of 10,000 candidate papers written in English.

The task is challenging for several reasons. First, the linguistic gap between social media and scientific writing is substantial: posts are short, informal, and often contain paraphrases, slogans, hashtags or incomplete statements, whereas scientific papers use more precise and technical language. Second, the reference to the source paper is implicit rather than explicit, so the retrieval system must infer the connection from partial semantic evidence rather than from direct citations. Third, relevant clues may appear in different forms, such as mentions of a scientific result, a medical outcome, a research institution or a publication venue, requiring the system to combine lexical and semantic evidence effectively.

An additional challenge is multilinguality: the official query sets include English, German and French posts while the candidate pool remains shared across languages.

Team RETRIX approaches this problem with a four-stage sequential pipeline.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

fabio.baldan.1@studenti.unipd.it (F. Baldan); davide.donati@studenti.unipd.it (D. Donati); alvise.garberino@studenti.unipd.it (A. Garberino); giancarlo.padoan@studenti.unipd.it (G. Padoan); marco.tessari.8@studenti.unipd.it (M. Tessari); nicola.ferro@unipd.it (N. Ferro)

1. In a preliminary stage, we translate to English all the queries written in French and German using the model utter-project/EuroLLM-9B-Instruct-2512 [4]. This allows us to handle all queries in the same way, independently of the original language and consequently of the semantic and structural differences.
2. Raw tweet-style queries are cleaned of emojis and symbols like "#" and "@" and then enriched via pseudo-relevance feedback: BAAI/bge-large-en-v1.5 [5] encodes both queries and corpus documents into a shared embedding space, retrieves the top-5 semantically similar papers per query, and extracts the most frequent terms from those papers as expansion tokens.
3. The expanded queries are fed into BAAI/bge-m3 [6] operating in hybrid mode, which combines dense inner-product search via FAISS [7] with sparse lexical matching and ColBERT multi-vector scoring to produce an initial ranked list of candidates.
4. Finally, the top candidates are reranked by a neural cross-encoder, in our case nvidia/llama-nemotron-rerank-1b-v2 [8], fine-tuned on the English training dataset of the CLEF CheckThat! Task 1 2026 and all the query datasets of the 2025 CLEF CheckThat! SubTask 4b.

The paper is organized as follows: Section 2 discusses the most relevant research directions related to this task, Section 4 describes in more detail our approach, Section 3 explains our experimental setup and Section 5 discusses our main findings. Finally, Section 6 draws conclusions and outlines future work.

2. Related Work

A starting point is lexical retrieval where probabilistic ranking models such as BM25 remain strong baselines thanks to their efficiency, interpretability, and robustness in candidate generation [9].

Neural retrieval methods have been proposed to mitigate the lexical mismatch between queries and documents by learning dense semantic representations. More broadly, neural information retrieval has shifted attention from exact token overlap toward representation learning and learned matching functions [10]. Within this line of work, dense dual-encoder architectures such as Dense Passage Retrieval (DPR) demonstrated that learned query and document embeddings can be highly effective for first-stage retrieval [11]. Building on this direction, BGE-M3 unified dense, sparse, and multi-vector retrieval in a single multilingual model [6], enabling hybrid scoring that combines embedding similarity with lexical token weights, a combination particularly well-suited to cross-lingual settings where queries and documents share few surface tokens.

Multilingual information retrieval presents additional challenges beyond monolingual settings, as systems must bridge language mismatches before relevance can be estimated. Machine translation provides one practical solution, but it can introduce errors and vocabulary drift, especially for technical or domain-specific terminology [12].

Vocabulary mismatch between short, informal queries and longer scientific documents can be further reduced through query expansion. Classical pseudo-relevance feedback methods such as RM3 [13] address this issue by reweighting query terms using top-ranked documents from an initial retrieval pass. Neural extensions of this paradigm replace or complement lexical evidence with dense representations [14].

Between initial retrieval and final ranking, many modern systems adopt multi-stage pipelines to balance efficiency and effectiveness [10]. Within these pipelines, late-interaction models such as ColBERT preserve fine-grained token-level matching while remaining substantially more scalable than fully pairwise neural scoring over the whole collection [15]. Similarly, transformer-based rerankers have become a common second-stage component for refining an initial candidate set through stronger query-document interactions [16].

Recent work has argued that reranker fine-tuning benefits from hard negatives because they force the model to learn finer-grained relevance boundaries than easy or random negatives provide [17, 18]. Adapting large language models to retrieval tasks is further facilitated by parameter-efficient fine-tuning with LoRA [19], which has been shown to match or approach full fine-tuning quality for passage reranking, while substantially reducing GPU memory requirements [20].

3. Experimental Setup

All experiments were conducted on our own machines: four laptops (no dedicated GPU) and a desktop computer with an AMD Ryzen 9 5900X, 32 GB of RAM and an NVIDIA RTX 3090 GPU (10,496 CUDA cores, 1.70 GHz boost clock, 24 GB of GDDR6X memory, 35.58 TFlops).

3.1. Dataset

The dataset consisted of 10,000 scientific publications that had to be paired with tweet-like posts written in three languages: German, French and English. The queries contain no explicit references such as URLs. Instead, each tweet contains an implicit reference to a scientific paper, for example through mentions of authors or keywords related to the paper. The dataset files are all in JSON format. The scientific papers dataset contains the fields pubkey (unique identifier), title, abstract, venue/location and authors. The query datasets contain the columns index, pubkey (ID of the positive scientific paper that has to be matched with the query) and text.

Table 1
Dataset statistics by language

Split	Language	File	Queries		Length (tokens)			
			#	%	Min	Median	Max	Mean
Train	EN	en_train.json	14977	77.8	2	33	57	31.5
	FR	fr_train.json	2807	14.6	2	33	57	31.5
	DE	de_train.json	1460	7.6	2	33	57	31.5
Total			**19244**	**100**	**2**	**33**	**57**	**31.5**
Dev	EN	en_dev.json	3905	78.2	5	33	60	31.7
	FR	fr_dev.json	702	14.1	5	33	60	31.7
	DE	de_dev.json	386	7.7	5	33	60	31.7
Total			**4993**	**100**	**5**	**33**	**60**	**31.7**
Test	EN	final_en_test.json	6076	74.4	5	33	58	31.9
	FR	final_fr_test.json	1220	14.9	5	33	58	31.9
	DE	final_de_test.json	876	10.7	5	33	58	31.9
Total			**8172**	**100**	**5**	**33**	**58**	**31.9**
Overall Total			**32409**		**2**	**33**	**60**	**31.6**

Figure 1 compares the distribution of token lengths across train, dev and test datasets. All three datasets share similar distributions, with the same median value and similar length value peaks.

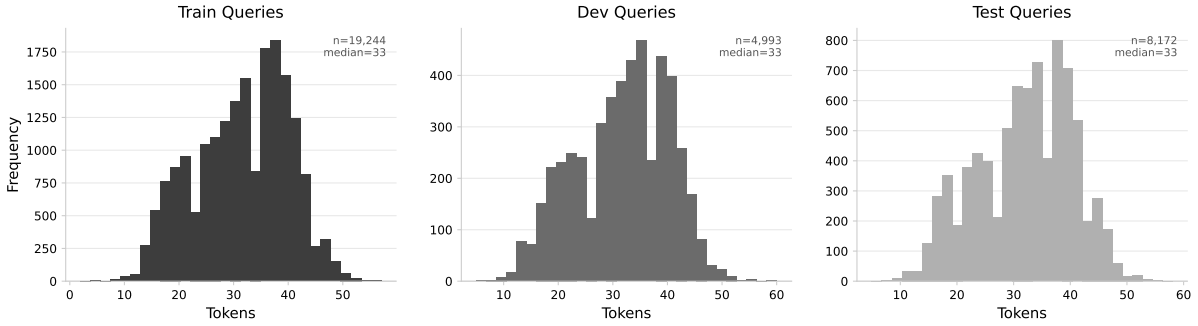


Figure 1: Train, dev and test datasets' distribution of query token length.

4. Methodology

We started with a lexical baseline based on Apache Lucene BM25 and progressively moved to a multi-stage neural pipeline.
The final pipeline (Figure 2) for the proposed information retrieval system consists of four stages:

- **Query translation**: queries in French or German are first translated into English.
- **Query expansion**: queries are expanded with additional context.
- **Retrieval**: relevant documents are retrieved from the corpus using the expanded query.
- **Reranking**: the retrieved documents are reranked based on their relevance to the original query.

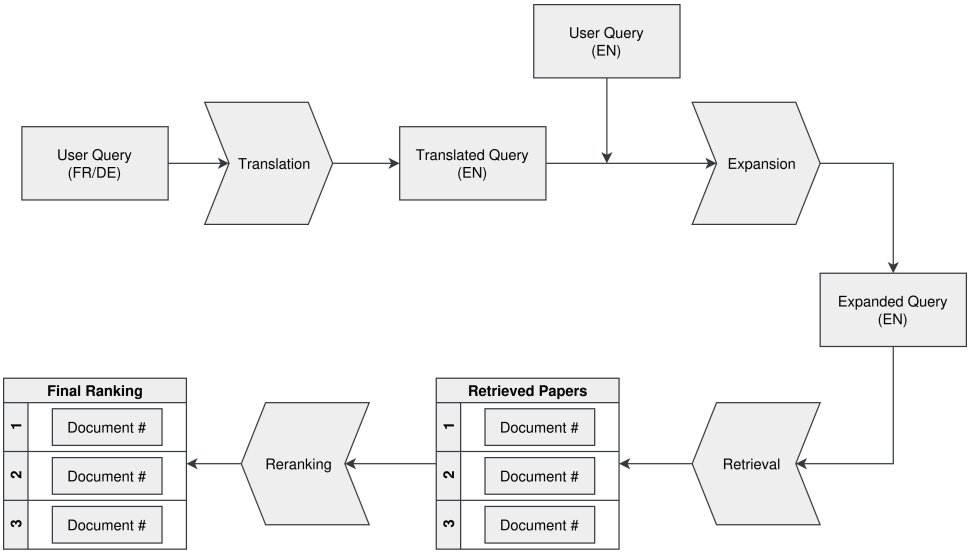


Figure 2: Four-stage pipeline for information retrieval.

4.1. Query Translation

In the preliminary stage of our pipeline, non-English queries (French and German) are translated into English. This is performed on a query-by-query basis using the model utter-project/EuroLLM-9B-Instruct-2512 with the following prompt:

LLM Prompt

```
system: You are a professional translator.
Translate the user's text to English.
Output ONLY the English translation.
Do not include the original text, explanations, labels, or any other content.

user: Translate this {French/German} text to English.
Output only the English translation, nothing else.

{Query original text x}
```

The resulting English queries are then fed into the following step of our pipeline.

4.2. Query Expansion

Following the translation stage, each query is processed by a deterministic query-expansion module designed to reduce query–document vocabulary mismatch between short user claims and the terminology used in the scientific corpus. The module does not rely on generative LLM-based paraphrasing.

Instead, it combines lexical feature extraction with dense semantic retrieval and neighborhood-based term induction.

- The translated English query is first analyzed with spaCy's en_core_web_sm pipeline, which provides token-level linguistic annotations, including stop-word indicators and coarse-grained part-of-speech labels from the Universal POS tag set. Candidate lexical anchors are then identified by retaining tokens tagged as NOUN or PROPN while excluding stop words and tokens shorter than three characters. The resulting set is de-duplicated to produce a compact lexical signature that captures the principal nominal content of the query.
- The semantic expansion stage operates on each document, where every document is represented as the concatenation of its title and abstract. These title–abstract pairs are embedded with BAAI/bge-large-en-v1.5. In accordance with the documented BGE retrieval interface [5], query embeddings are produced by prepending the instruction "*Represent this sentence for searching relevant passages:*" to the translated query, whereas the corpus documents are encoded as retrieval targets. Similarity is then computed on normalized embeddings using the dot product. The five highest-scoring documents are retained as the local semantic neighborhood used for expansion. In the described implementation, candidate expansion terms are extracted from the retrieved documents after lower-casing, ASCII normalization, removal of mention and hashtag symbols, stop-word filtering and exclusion of short tokens (tokens ≤ 2 characters long are discarded). The candidate pool is aggregated across the retrieved neighborhood and assigned weights derived from the semantic similarity of the source documents so that terms originating from more similar neighbors contribute more strongly. Terms already present in the query are removed and the fifteen highest-ranking candidates are retained as explicit expansion terms.

The final expanded query is constructed by concatenating three components: the cleaned (already translated) query, the noun and proper-noun-based keyword set extracted directly from the query and the expansion terms induced from the retrieved document neighborhood. This output expanded query is then used by the downstream retrieval pipeline.

4.3. Retrieval

The retrieval stage is designed as a high-recall candidate-generation step between query expansion and neural reranking. Its objective is not to make the final relevance decision but to ensure that the true source paper is included in a sufficiently large candidate set for the subsequent, more computationally expensive cross-encoder stage. For this reason, after experimenting with a Lucene BM25 baseline, the final pipeline adopts BAAI/bge-m3 in a hybrid retrieval setting.

The retriever searches the fixed scientific papers collection provided by the task organizers. Each paper is represented at the document level by concatenating its title and abstract (batch size = 1024), and the retrieved unit is always the paper pubkey. Although the collection includes metadata such as venue and authors, this hybrid retrieval stage represents every scientific paper as the concatenation of just titles and abstracts to reduce inference time.

We chose BAAI/bge-m3 because it is well suited to this retrieval setting. The model can produce dense vectors, sparse lexical weights, and ColBERT-style multi-vector representations from the same encoder.

This allows the retriever to combine broad semantic matching, exact term evidence, and fine-grained token-level matching without switching between unrelated encoders.

Candidate selection is performed in two steps. First, all documents are encoded as dense BGE-M3 vectors and searched with an inner-product FAISS index [7].

This dense stage acts as an efficient broad filter over the whole collection. The default setting retrieves a large candidate pool of *candidate_multiplier* $\times$ *top_k*, with candidate_multiplier = 3 and top_k = 2000, before the final truncation. The number of candidates passed to the hybrid rescoring stage is:

$$K_c = \min(N, \max(K, K_m))$$

where N is the size of the full paper collection (10,000), $K = top\_k$ is the final candidate-pool size after truncation, $K_m = candidate\_multiplier \times top\_k$ is the initial dense-stage retrieval size and K_c is the resulting number of candidates passed to the hybrid rescoring stage.

This deliberately large candidate set favors recall, which is more important than precision at this stage.

Second, the dense candidates are rescored using the sparse and ColBERT-style representations. The sparse component rewards important overlapping terms, while the ColBERT-style component captures local token correspondences that may be lost when each document is represented by a single dense vector [15]. The ColBERT-style late-interaction score is computed as [15]:

$$s_{colbert}(q, d) = \frac{1}{N_q} \sum_{i=1}^{N_q} \max_{1 \le j \le N_d} q_i^\top d_j$$

where q_i and d_j are the i-th and j-th ColBERT-style token-level vectors of the query and document, respectively, and N_q, N_d are their corresponding numbers of tokens.

The final configuration was selected through a series of development runs.

The best results in our analysis were obtained with a configuration in which BGE-M3 hybrid retrieval used the same expanded query, described in the previous section, for all three retrieval modes. We then tested several weighting configurations. Among them, the run labeled bge_m3_hybrid_513 was retained because it provided the strongest recall-oriented trade-off among the retrievers evaluated without reranking. This configuration used weights of 0.5, 0.15, and 0.35 for dense, sparse, and ColBERT scoring, respectively, with a multiplier of 3 and a maximum hybrid input length of 384 tokens.

A closely related configuration with a larger ColBERT weight achieved a nearly identical MRR@5 but a lower Recall@100. We therefore preferred the configuration that better matched the role of retrieval as a candidate generator. In addition, increasing the reranked candidate depth from a few hundred documents to between 1000 and 2500 documents increased the final recall by giving the neural reranker a larger candidate pool.

The final hybrid retrieval score is therefore computed as:

$$s(q, d) = 0.5\, s_{\text{dense}}(q, d) + 0.15\, s_{\text{sparse}}(q, d) + 0.35\, s_{\text{colbert}}(q, d).$$

where $s_{\text{dense}}(q, d)$, $s_{\text{sparse}}(q, d)$, and $s_{\text{colbert}}(q, d)$ are the dense, sparse, and ColBERT-style scores defined above, respectively.

The largest weight is assigned to the dense score because it provides the main semantic signal. The sparse and ColBERT components add complementary lexical and fine-grained matching evidence. Candidates are sorted by this score in descending order and truncated to the final top_k. The retrieval output is a JSON mapping each query ID to an ordered list of paper pubkeys.

4.4. Reranking

4.4.1. Cross-encoder scoring

We experimented with several rerankers:

- Qwen/Qwen3-Reranker-4B [21];
- Alibaba-NLP/gte-reranker-modernbert-base [22, 23];
- nvidia/llama-nemotron-rerank-1b-v2 [8].

From our findings, the strongest reranker among those three is nvidia/llama-nemotron-rerank-1b-v2, a 1B-parameter LLaMA-based model fine-tuned as a sequence classifier. For each query-document pair, we build an input sequence using the official NVIDIA prompt template:

LLM Prompt

```
question:{query} \n \n passage:{passage}
```

Reranking deeper candidate pools (500-2500 documents) consistently produced better final results than shallower pools, confirming the importance of high recall at the retrieval stage to compensate for mislabeling from weaker previous retrieval-stages and the reranker's ability to distinguish noise from documents that are actually relevant.

At this stage, queries (social-media like posts) are evaluated against the scientific papers corpus. The evaluation is performed taking into account not only the title and the abstract, but also the author of each scientific paper.

4.5. Fine-Tuning the Reranker

To improve the Nemotron reranker on the domain-specific characteristics of the CheckThat! task, we fine-tuned nvidia/llama-nemotron-rerank-1b-v2 using Parameter-Efficient Fine-Tuning (PEFT) with Low-Rank Adaptation (LoRA) on data from both the 2025 and 2026 editions of the task.

4.5.1. Training Data and Hard-Negative Mining

Training examples are query-document pairs with binary labels: a positive document is the gold source paper associated with the query, while negative documents are hard negatives mined using sentence-transformers/static-retrieval-mrl-en-v1 [24, 25, 26]. Hard negatives are selected from the range $[5, 50]$ nearest neighbors (by embedding similarity) that satisfy score ≤ 0.85 and lie at least 0.05 below the positive score in cosine similarity (absolute margin) preventing trivially easy or falsely relevant negatives from degrading the effectiveness of fine-tuning.

Training data combines two years of the task:

- en_train.json of the CheckThat! Lab CLEF 2026: split 90% training / 10% development. The development set is drawn exclusively from 2026 data for consistency with the official CT26 Task 1 evaluation protocol. Due to time and resource constraints, we used just the en_train.json dataset but we hypothesize that also German and French translated queries datasets could further improve the quality of fine-tuning;
- Datasets of task4/subtask_4b, CheckThat! Lab CLEF 2025: all available topics (dev, test_gold and train splits) are used entirely for training.

This design ensures that the dev set used to track MRR@5 during training is distribution-aligned with the contest test queries. Each training record contains one query, one positive passage and up to five hard negatives.

4.5.2. LoRA Architecture

To fine-tune the 1B-parameter base model with limited GPU memory, we apply LoRA [19] with rank $r = 16$ and scaling factor $\alpha = 32$. Adapters are inserted into all attention and feed-forward projection layers: q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj and down_proj. The classification head (score) is included in modules_to_save and is fully updated during training. The task type is SEQ_CLS, matching the sequence-classification objective of the reranker.

4.5.3. Training Objective

The loss function used is BinaryCrossEntropyWithLogits, where the model produces a single scalar logit z per input pair. Training minimizes binary cross-entropy with logits over all positive ($y = 1$) and negative ($y = 0$) pairs in each batch:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\Big[y_i \log \sigma(z_i) + (1 - y_i)\log(1 - \sigma(z_i))\Big]$$

where B is the effective batch size and σ is the sigmoid function. Gradient accumulation over 4 steps is used to simulate a larger effective batch size of 32 without additional memory overhead.

4.5.4. Optimization

The model had been trained with AdamW, a learning rate of 2×10^{-5}, a linear warm-up over 10% of the total training steps and gradient clipping at norm 1.0. Training runs for 3 epochs with bfloat16 mixed precision and gradient checkpointing enabled to reduce VRAM usage. The checkpoint with the best MRR@5 on the 2026 development set is retained.

4.5.5. Inference

At inference time, the LoRA adapter weights are merged into the base model via merge_and_unload() before scoring, so no PEFT overhead is incurred during evaluation. The scoring procedure and prompt format are identical to those of the base reranker (Section 4.4), and the fine-tuned model is used as a drop-in replacement in the reranking stage.

4.6. Evaluation

We evaluated systems on the CheckThat! Task 1 query sets by matching retrieved pubkeys against the gold pubkey for each query. The official evaluation metric for the task is MRR@5. We also tracked Recall@100 to monitor candidate-generation quality and to verify that reranking gains do not come from overly aggressive filtering.

5. Results and Discussion

5.1. Overview

This section summarizes the empirical evidence that supports our pipeline choices and diagnoses remaining failure modes.

Confidence intervals reported in descriptive tables use the normal approximation over per-query scores and are explicitly labeled as descriptive summaries rather than inferential tests. They do not adjust for multiple comparisons. "nvidia/llama-nemotron-rerank-1b-v2" and "Nemotron" are used interchangeably.

5.2. Experimental Results on the TEST Set

Hybrid variants differ only in the linear combination of dense, sparse and ColBERT-style BGE-M3 signals. We abbreviated bge_m3_hybrid_315's system to H315 (same convention for other hybrid-retrieval systems), similar convention for Nemotron's systems (N), Qwen3-4B (Q) and gte-reranker-modernbert-base (GTE).

As can be seen in table 3, H424 is the weakest reported hybrid configuration, while H513 and H315 perform substantially better on early-precision metrics. H513 has an advantage in Recall@100 relative to H315, which motivates its selection as the candidate generator for the full pipeline: retrieval should prioritize coverage that the reranker can refine. Neural reranking (Nemotron) yields the largest improvements: moving from H513 to N1000 increases MRR@5 by roughly 0.145 and increases Recall@100 notably, showing that deeper reranking improves both precision and coverage. However, gains in MRR@5 beyond top-500 are marginal, suggesting diminishing returns for very deep pools.

Table 2
Official TEST set results metrics. The aggregate row is a micro-average over all queries.

Language	MRR@5	Recall@100	MAP	nDCG@10
EN	0.7117	0.9235	0.7202	0.7469
FR	0.7400	0.9328	0.7479	0.7689
DE	0.6806	0.9155	0.6907	0.7213
Aggregate	0.7126	0.9240	0.7212	0.7474

Table 3
Run comparison on the en_train.json file. Δ takes as reference the H513 run. The weights of the runs with BGE-M3 are in the following order: dense, sparse, Colbert.

Run	Configuration	MRR@5	Recall@100	MAP	nDCG@10	Δ MRR@5
BM25	BM25 retrieval with stopword removal (stoplist_en_ranksnl_large.txt), trim filter, removal of the # symbol, no removal of @, accent normalization (ASCIIFoldingFilter), LowerCaseFilter, EnglishPossessiveFilter, KstemFilter, no use of WordDelimitedGraphFilter, searcher with OR boolean queries and BAAI/bge-large-en-v1.5 query expansion, no rerank.	0.5031	0.7965	0.5156	0.5439	-0.0536
H424	BGE-M3 weights 0.40/0.20/0.40	0.5067	0.8277	0.5201	0.5531	-0.0500
H514	BGE-M3 weights 0.50/0.10/0.40	0.5141	0.8327	0.5273	0.5595	-0.0426
H414	BGE-M3 weights 0.45/0.10/0.45	0.5171	0.8327	0.5302	0.5621	-0.0396
H513	**BGE-M3 weights 0.50/0.15/0.35**	**0.5567**	**0.8392**	**0.5688**	**0.5956**	**0.0000**
H315	BGE-M3 weights 0.35/0.15/0.50	0.5572	0.8365	0.5692	0.5957	+0.0005
Q100	H513 candidates reranked with Qwen3-4B, top-100 retained	0.6624	0.8392	0.6696	0.6953	+0.1057
GTE100	H513 candidates reranked with gte-reranker-modernbert-base, top-100 retained	0.6511	0.8392	0.6594	0.6830	+0.0944
N100	H513 candidates reranked with Nemotron, top-100 retained	0.6847	0.8392	0.6909	0.7146	+0.1280
N500	H513 candidates reranked with Nemotron, top-500 retained	0.7010	0.8986	0.7096	0.7342	+0.1443
N1000	**H513 candidates reranked with Nemotron, top-1000 retained**	**0.7018**	**0.9109**	**0.7111**	**0.7357**	**+0.1451**

5.3. Multilingual DEV Split Validation and Experimental Results on the FINAL Set

DEV runs were not used to re-tune the TRAIN analysis pipelines. They are presented as external validation with appropriately wider uncertainty bands.

The development results suggest that the final pipeline generalizes reasonably well across languages after translation, but not uniformly. French achieved the highest scores with MRR@5 of 0.7599 and Recall@100 of 0.9288. German is the weakest with MRR@5 of 0.6736 and the widest confidence interval. Rank-bucket counts corroborate these summaries: French has the largest fraction of rank-1 successes and the fewest missing cases (496/40), while German shows fewer rank-1 hits and a higher unresolved fraction relative to its query count (237/29). In practice, this pattern suggests that the reranker generalizes well to translated queries in French, that English remains competitive and that German's lower point estimates and wider intervals are plausibly driven by both smaller sample size and translation/terminology drift. These DEV findings validate the pipeline qualitatively and motivate the decision to use a deeper top-2000 pool in the final TEST submission to favor recall.

Table 5 shows that all the previous ideas and assumptions have been confirmed in the FINAL set. Increasing the depth of the reranking, managing metadata such as authors, fine-tuning, and selecting the reranker led to results that allowed us to finish fifth in the rankings, despite limited computational resources. Qwen/Qwen3-Reranker-4B appears to have difficulty interpreting additional information such as authors correctly. This may be due to its nature as an LLM-based reranker, whereas other cross-encoders perform better.

Table 4
Fine-tuned Nemotron LoRA validation on labeled development data. The final column reports rank-bucket counts for strong successes and unresolved failures.

Split	Queries	MRR@5	95% CI	Recall@100	MAP	nDCG@10	Rank 1 / Missing
EN DEV	3905	0.7160	[0.7030, 0.7290]	0.9129	0.7243	0.7483	2574 / 241
FR DEV	702	0.7599	[0.7310, 0.7888]	0.9288	0.7672	0.7894	496 / 40
DE DEV	386	0.6736	[0.6305, 0.7168]	0.8860	0.6820	0.7080	237 / 29

Table 5
Runs comparison on the final_en_test.json set, FT stands for fine-tuned. All the topk100 runs are based on bi513_encoder_results_bge_large_topk1000, while the topk2000 is based on bi513_encoder_results_bge_large_topk2000.

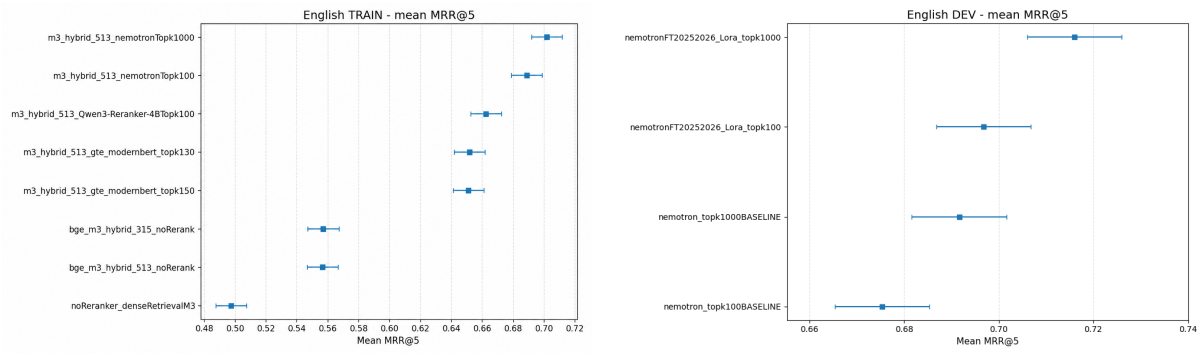
Run	MRR@5	F1@1	MAP	nDCG@5	Recall@100
nemotronFT_topk2000_withAuthors	**0.7117**	**0.6501**	**0.7202**	**0.7346**	**0.9235**
nemotronFT_topk100_withAuthors	0.6918	0.6358	0.6972	0.7118	0.8371
nemotron_topk100_withAuthors	0.6782	0.6229	0.6845	0.6982	0.8371
nemotron_topk100_noAuthors	0.6647	0.6058	0.6721	0.6858	0.8371
Qwen3-Reranker-4B_topk100_noAuthors	0.6421	0.5810	0.6505	0.6644	0.8371
gte_modernbert_topk100_withAuthors	0.6382	0.5841	0.6467	0.6588	0.8371
Qwen3-Reranker-4B_topk100_withAuthors	0.6337	0.5734	0.6430	0.6556	0.8371
gte_modernbert_topk100_noAuthors	0.6304	0.5757	0.6395	0.6507	0.8371
bi513_encoder_results_bge_large_topk1000	0.5047	0.4378	0.5195	0.5308	0.8371
bi414_encoder_results_bge_large_topk1000	0.5035	0.4371	0.5181	0.5297	0.8348

5.4. Statistical Analysis

Descriptive confidence intervals (Figure 3) summarize uncertainty around mean per-query scores but do not by themselves imply statistical significance and do not correct for multiple comparisons. For inference, we follow a stage-aligned strategy: test retrieval-stage decisions on Recall@100 (coverage), and test reranking-stage decisions on MRR@5 (early precision). Because per-query metrics are bounded and strongly non-normal, we use the non-parametric Friedman repeated-measures test as the omnibus alternative to a repeated-measures ANOVA, followed by pairwise Wilcoxon signed-rank tests with Holm correction [27, 28].

5.4.1. Descriptive Statistics

As shown below, reranking shifts the entire distribution: the median increases from 0.5 (H513) to 1.0 for Nemotron runs, and the first quartile rises from 0.0 to 0.25, indicating that a substantially larger fraction of queries receive a non-zero early-precision score after reranking.



(a) English TRAIN. (b) English DEV.

Figure 3: Mean MRR@5 with normal-approximation 95% confidence intervals for the English TRAIN and DEV statistical-analysis runs.

Table 6
Descriptive statistics of per-query MRR@5 on the TRAIN English split. Confidence intervals use the normal approximation over queries.

Run	Mean	Median	Std. dev.	Variance	Q1	Q3	95% CI
H513	0.5567	0.5	0.4614	0.2129	0.00	1.00	[0.5493, 0.5641]
N100	0.6847	1.0	0.4303	0.1852	0.25	1.00	[0.6778, 0.6916]
N500	0.7010	1.0	0.4229	0.1788	0.25	1.00	[0.6943, 0.7078]
N1000	0.7018	1.0	0.4223	0.1784	0.25	1.00	[0.6950, 0.7086]

5.4.2. Significance Testing

The Friedman omnibus test rejects the null hypothesis that all English TRAIN systems share the same per-query MRR@5 distribution ($\chi^2 = 21368.46, p < 0.001$; 14,977 queries). Table 7 reports the pairwise Wilcoxon signed-rank tests (Holm-corrected) most relevant to our pipeline decisions, aligned with the stage each comparison targets.

H513 is the best-supported hybrid candidate generator on Recall@100, significantly outperforming all other tested hybrid configurations. H513 and H315 are indistinguishable on MRR@5, so choosing between them should rely on retrieval-stage diagnostics and operational constraints rather than early-precision differences. The Nemotron reranker yields a large, statistically supported MRR@5 gain over the retained hybrid baseline (+0.1451), while the N500-to-N1000 difference is statistically detectable but practically negligible, suggesting N500 is preferable when latency or cost is a concern.

Table 7
Holm-corrected pairwise Wilcoxon signed-rank tests on the English TRAIN split, aligned with pipeline stage (retrieval → Recall@100; reranking → MRR@5). Δ differences are computed as the second system minus the first or as H513 minus each comparator for the aggregated retrieval-stage row.

Stage	Metric	Comparison	Δ	p_{holm}	Interpretation
Retrieval	Recall@100	H513 vs. {H315, H414, H424, H514}	+0.0026 to +0.0115	all < 0.01	H513 significantly better in all four comparisons
Reranking	MRR@5	H315 vs H513	-0.0005	1.0000	No significant difference
Reranking	MRR@5	H424 vs H513	+0.0500	8.58×10^{-116}	H513 significantly better
Reranking	MRR@5	H513 vs N1000	+0.1451	< 0.001	N1000 significantly better
Reranking	MRR@5	N1000 vs N500	-0.0008	0.0497	Significant but negligible in size

5.4.3. Rank-Bucket and Error Analysis

Table 8 reports the rank position of the gold document for the retained hybrid and reranked runs. A gold paper is counted as *missing* only when its gold `pubkey` is absent from every document available in the ranked file for that query. For runs available only as submitted top-100 lists, missing means absent from that file and not necessarily absent from the original upstream retrieval pool.

Table 8
Rank buckets for the retained hybrid and reranked English runs. Percentages are computed within each split: 14,977 labeled English queries, 3,905 English DEV queries, and 6,076 English TEST queries.

Split	Run	Queries	Rank 1	Ranks 2–5	Ranks 6–10	Ranks 11–100	Rank > 100	Missing
EN TRAIN	H513 (topk100)	14977	7459 (49.8%)	2280 (15.2%)	715 (4.8%)	2114 (14.1%)	0 (0.0%)	2409 (16.1%)
EN TRAIN	N1000	14977	9697 (64.7%)	2057 (13.7%)	598 (4.0%)	1291 (8.6%)	405 (2.7%)	929 (6.2%)
EN DEV	N1000 FT	3905	2574 (65.9%)	538 (13.8%)	144 (3.7%)	309 (7.9%)	99 (2.5%)	241 (6.2%)
EN TEST	N2000 FT	6076	3950 (65.0%)	926 (15.2%)	229 (3.8%)	506 (8.3%)	0 (0.0%)	465 (7.7%)

Overall, reranking substantially improves early precision while leaving two residual failure modes: ranking failures, where the gold paper is present but below the top-5 cutoff, and missing cases, which cannot be repaired by reranking alone and are likely associated with severe vocabulary mismatch, translation drift or claims whose wording emphasizes entities weakly represented in titles and abstracts.

5.4.4. Performance by Query Characteristics

Analyzing the N1000 run on the English TRAIN split by query length and by a lightweight named-entity heuristic (capitalization patterns, acronyms, institution/venue keywords), we found that, contrary to intuition, longer queries do not perform better. Short queries ($\leq$20 tokens) reach 0.760 MRR@5, medium-length queries 0.713 and long queries ($\geq$35 tokens) 0.672, likely because longer tweet-like claims add conversational material that does not aid retrieval. Entity-rich queries reach 0.733 MRR@5 versus 0.664 for entity-sparse queries, consistent with the value of combining lexical anchors with semantic matching. These patterns are exploratory heuristics rather than formal statistical findings.

5.5. Reproducibility

In the repository *https://bitbucket.org/upd-dei-stud-prj/seupd2526-retrix/src/master/* are stored almost all the scripts, programs and runs used in the project (the file size limit for the platform is 100 MB).

6. Conclusions and Future Work

The results of the described system support three main conclusions.

First, analyzer and BM25 tuning provide a useful lexical baseline, especially thanks to its low computational footprint, with the best analyzer-focused run reaching 0.503 MRR@5 and 0.796 Recall@100 in the English run analysis.

Second, moving to BGE-M3 hybrid retrieval produces a stronger candidate generator: the retained configuration, bge_m3_hybrid_513, reaches 0.557 MRR@5 and 0.839 Recall@100.

Third, reranking the hybrid candidates with nvidia/llama-nemotron-rerank-1b-v2 gives the largest observed improvement among the reported single-stage runs, reaching 0.702 MRR@5 and 0.911 Recall@100 when reranking the top 1000 candidates. Alternative rerankers such as Qwen3-Reranker-4B and Alibaba-NLP/gte-reranker-modernbert-base also improve results, but remain below the Nemotron runs in the reported MRR@5 values.

The development results show that fine-tuning the Nemotron reranker is promising. Fine-tuning improves performance by a small but appreciable amount (an average gain of 0.03 MRR@5 points in our experiments).

Several directions emerge for future improvement. First, investing more time in analysis and fine-tuning of the reranker and all other components of the system would probably bring better results. Many techniques, such as data augmentation can be experimented with, and there is room for improvement.

A second direction concerns the queries that remain entirely missing after reranking. These cases cannot be recovered by any reranker because the correct publication is absent from the candidate set. Manual inspection of a sample suggests three dominant failure patterns: claims referencing quantitative findings not mentioned in the abstract, implicit references that name no concrete entities and translation drift that changes the intended meaning of domain-specific terms. Addressing the first two patterns would probably require managing content beyond title, abstract, and authors, as well as other metadata. Addressing the translation failure pattern points toward a dedicated post-editing step or a domain-adapted translation model fine-tuned on biomedical text.

Finally, the German dataset achieves MRR@5 of 0.68 in the evaluation results: a wide gap relative to English and French. Controlled ablations that isolate translation errors from retrieval errors would clarify how much of this gap is attributable to the translation stage versus vocabulary mismatch in the retrieval stage itself.

7. Declaration on Generative AI

During the writing of this paper, the authors used Grammarly and DeepL for grammar and spelling checks and Claude Sonnet 4.6 for writing-style improvement, citation management, and formatting

assistance. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication content.

References

[1] J. M. Struß, S. Schellhammer, S. Dietze, V. V., V. Setty, T. Chakraborty, P. Nakov, A. Anand, P. Chungkham, S. Hafid, D. Sahnan, K. Todorov, The clef-2026 checkthat! lab: Advancing multilingual fact-checking, in: R. Campos, A. Jatowt, Y. Lan, M. Aliannejadi, C. Bauer, S. MacAvaney, A. Anand, Z. Ren, S. Verberne, N. Bai, M. Mansoury (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2026, pp. 325–335.

[2] S. Schellhammer, S. Hafid, Y. S. Kartal, K. Boland, D. Dimitrov, K. Todorov, S. Dietze, Overview of the CLEF-2026 CheckThat! lab task 1 on source retrieval for scientific web claims, in: [3], 2026.

[3] E. S. Salido, A. Barrón-Cedeño, Alba García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CLEF 2026, Jena, Germany, 2026.

[4] UTTER Project, EuroLLM-9B-Instruct-2512, https://huggingface.co/utter-project/EuroLLM-9B-Instruct-2512, 2024. Hugging Face model card. Accessed: 2026-05-07.

[5] BAAI (Beijing Academy of Artificial Intelligence), BGE-large-en-v1.5: A General Embedding Model for English, Hugging Face, 2023. URL: https://huggingface.co/BAAI/bge-large-en-v1.5, accessed: 07-05-2026.

[6] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024. arXiv:2402.03216.

[7] Faiss, Faiss documentation, https://faiss.ai/, 2026. Accessed: May 2026.

[8] NVIDIA, Llama Nemotron Reranking 1B V2: A Multilingual Cross-Encoder for Document Reranking, Hugging Face, 2026. URL: https://huggingface.co/nvidia/llama-nemotron-rerank-1b-v2, accessed: 07-05-2026.

[9] S. E. Robertson, U. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends in Information Retrieval (FnTIR) 3 (2009) 333–389.

[10] S. Marchesin, A. Purpura, G. Silvello, Focal elements of neural information retrieval models. An outlook through a reproducibility study, Information Processing & Management 57 (2020) 102109:1–102109:29.

[11] 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: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. URL: https://aclanthology.org/2020.emnlp-main.550/. doi:10.18653/v1/2020.emnlp-main.550.

[12] 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 (ACL), Berlin, Germany, 2016, pp. 1715–1725. doi:10.18653/v1/P16-1162.

[13] V. Lavrenko, W. B. Croft, Relevance based language models, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New Orleans, LA, USA, 2001, pp. 120–127. doi:10.1145/383952.383972.

[14] X. Wang, C. Macdonald, N. Tonellotto, I. Ounis, ANCE-PRF: Indexing disentanglement for pseudo-relevance feedback with approximate nearest neighbor search, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 5765–5778. doi:10.18653/v1/2021.emnlp-main.453.

[15] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over bert, 2020. URL: https://arxiv.org/abs/2004.12832.

[16] R. Nogueira, Z. Jiang, R. Pradeep, J. Lin, Document Ranking with a Pretrained Sequence-to-Sequence Model, in: Findings of the Association for Computational Linguistics: EMNLP 2020,

Association for Computational Linguistics, Online, 2020, pp. 708–718. URL: https://aclanthology.org/2020.findings-emnlp.63/. doi:10.18653/v1/2020.findings-emnlp.63.

[17] H. Meghwani, A. Agarwal, P. Pattnayak, H. L. Patel, S. Panda, Hard Negative Mining for Domain-Specific Retrieval in Enterprise Systems, https://arxiv.org/abs/2505.18366, 2025. arXiv:2505.18366, accessed: May 2026.

[18] T. Aarsen, Training and Finetuning Reranker Models with Sentence Transformers, https://huggingface.co/blog/train-reranker, 2025. Hugging Face blog. Accessed: May 2026.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2022. URL: https://arxiv.org/abs/2106.09685.

[20] A. Nijasure, T. Chowdhury, S. Ghodratnama, How relevance emerges: Interpreting LoRA fine-tuning in reranking llms, arXiv preprint arXiv:2504.08780 (2025). arXiv:2504.08780.

[21] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, J. Zhou, Qwen3 embedding: Advancing text embedding and reranking through foundation models, arXiv preprint arXiv:2506.05176 (2025).

[22] X. Zhang, Y. Zhang, D. Long, W. Xie, Z. Dai, J. Tang, H. Lin, B. Yang, P. Xie, F. Huang, et al., 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, 2024, pp. 1393–1412.

[23] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, M. Zhang, Towards general text embeddings with multi-stage contrastive learning, arXiv preprint arXiv:2308.03281 (2023).

[24] T. Aarsen, static-retrieval-mrl-en-v1: Static embeddings with bert tokenizer finetuned for retrieval with matryoshka loss, Hugging Face, 2025. URL: https://huggingface.co/sentence-transformers/static-retrieval-mrl-en-v1, accessed: 07-05-2026.

[25] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.

[26] A. Kusupati, G. Bhatt, A. Rege, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain, A. Farhadi, Matryoshka representation learning, 2024. arXiv:2205.13147.

[27] M. Friedman, The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance, Journal of the American Statistical Association 32 (1937) 675–701.

[28] F. Wilcoxon, Individual Comparisons by Ranking Methods, Biometrics Bulletin 1 (1945) 80–83.