SAGASU at CheckThat! 2026: Combining Sparse and Dense Approaches for Implicit Source Retrieval

Notebook for the CheckThat! Lab at CLEF 2026

Greta Brocco[1], Daniele Daccordo[1], Van Sang Ho[1], Davide Pinzan[1], Aurora Sancetta[1] and Nicola Ferro[1]

[1]*University of Padua, Italy*

Abstract

This paper outlines the procedures followed by Team SAGASU for the "Source retrieval for scientific web claims" task of the CheckThat! Lab in the 2026 edition of CLEF: the proposed task requires to align implicit references in social media posts to the related scientific literature. Our approach combines classical sparse IR methodologies and dense techniques. In particular, a dense model is used to apply reranking to the results of a sparse one. We also tackle the problem of Cross-Language Information Retrieval (CLIR) using machine traslation models. In addition, we evaluated the hypothesis of integrating Large Language Models (LLMs) in order to generate more effective queries. The approach hybridizing sparse with a dense re-ranker led us to reach a Mean Reciprocal Rank at cutoff 5 (MRR@5) over the test set of 0.5606, 0.5348 and 0.4443 respectively for English, French and German.

Keywords

information retrieval, source retrieval, claim verification

1. Introduction

Since its conception, the Web has become an increasingly viable source for accessing scientific information. Social media, in particular, contribute to the diffusion of scientific information to a wide public, allowing for discussing and sharing claims among a broad audience. Social media posts often impose constraints on the amounts of characters that can be written, and online discourse usually features more simplistic language, which can make expressing fully accurate claims difficult. Moreover, some content on social media is purposefully crafted with misleading information to generate engagement, which can then lead to the spread of misinformation. In this context, systems that can retrieve verified sources for claims expressed on the web are of the essence.

Task 1 of CheckThat! Lab at CLEF 2026 [1, 2] addresses this problem, focusing on the retrieval of scientific documents that are implicitly referenced by social media posts; that is, given a social media post that exhibits a scientific claim, the single paper that represents the original source of that information should be retrieved by the system [3]. The dataset provided by CheckThat! Lab includes the social media posts and the collection of reference papers. The dataset includes English, French and German social media posts and documents. Documents inside the collection are comprised of the title and abstract of the paper, as well as the authors and venue in which the paper was published. In our approach, an initial pipeline employing sparse models based on the Probabilistic Relevance Framework (PRF) [4] screens candidate papers from social media posts. We tested and compared various combinations of configuration parameters, and also gauged the usage of a Large Language Model in this inital phase to generate queries from the social media posts. The batch of candidate results is then eventually re-ranked through a cross-encoder model. Our contributions are summarized as follows:

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21–24, 2026, Jena, Germany

greta.brocco@studenti.unipd.it (G. Brocco); daniele.daccordo@studenti.unipd.it (D. Daccordo); vansang.ho@studenti.unipd.it (V. S. Ho); davide.pinzan.1@studenti.unipd.it (D. Pinzan); aurora.sancetta@studenti.unipd.it (A. Sancetta); nicola.ferro@unipd.it (N. Ferro)

https://www.dei.unipd.it/~ferro/ (N. Ferro)

0000-0001-9219-6239 (N. Ferro)

- We evaluate the performance for source retrieval of a system that initially ranks documents based on sparse (PRF) methodologies, and produces a final ranking with a dense (cross-encoder) model applied to the preliminary results.
- We assess the application of LLMs to synthesize social media posts into queries containing fewer terms.

The paper is organized as follows: Section 2 introduces related works; Section 3 describes our approach; Section 4 explains our experimental setup; Section 5 discusses our main findings; finally, Section 6 draws some conclusions and outlooks for future work.

2. Related Work

Source retrieval is a specific field of Information Retrieval (IR). As such, many IR techniques have been applied in order to have reliable source retrieval from claims.

IR models are typically referred to as either *sparse* or *dense*. Sparse approaches leverage sparse vector representations of documents and queries, and the retrieval approach is purely lexical, focusing on exact matches between terms. Among these, Okapi BM25 [5] is based on the concept of probabilistic relevance and is one of the most prominent approaches. Dense approaches are based on dense vector representations, namely word and sentence embeddings, and their retrieval strategy is semantic. These approaches typically make use of encoder-only transformer-based architectures (most notably BERT [6]) to compute embeddings, and are ulteriorly divided into dual encoders and cross-encoders: the former compute separate embeddings for documents and queries, while the latter compute a single representation for the interaction between query and document. These approaches have been applied in the field of source retrieval as well. In [7], a system combining sparse retrieval based on BM25 and dense retrieval is presented. In [8], both sparse and dense methodologies are applied for initial candidate retrieval and are finally ranked with an ulterior dense approach. Efforts have also been made to explore the potentialities of LLMs in IR. In [9], Large Language Models are used to address the problem of stylistic differences between the language used in social media posts and in scientific documents.

3. Methodology

The system was incrementally developed to accomodate different configurations and components schematized as in figure 1. Overall, the two main phases are a sparse baseline that performs lexical matching with BM25 and a reranking phase using a cross-encoder model.

We will be referring hereafter to the social media posts as "topics", following the TREC nomenclature, as they are the surrogate for the information need in this specific case.

3.1. Retrieval pipeline

Preprocessing Preprocessing was applied to collection documents and topics in the sparse phase of the pipeline on account of the lexical nature of sparse retrieval methods. Apart from classical processes, such as stemming and stopword filtering, some considerations arise from the nature of the dataset entries. First of all, a token is considered to be exclusively anything that is followed and/or preceded by a whitespace and any standalone word (i.e. in case only one word is present) to preserve terms specific to scientific domains. For example, the word "a(h1n1)pdm09", a proper term of the medical domain referring to a specific virus [10], would be split around the parentheses according to Unicode Text Segmentation [11]. Trailing and leading special characters typical of social media discourse (such as hashtags) have been removed, as well as user handles (tag/mentions) and email addresses (found in the *Authors* field of collection documents).

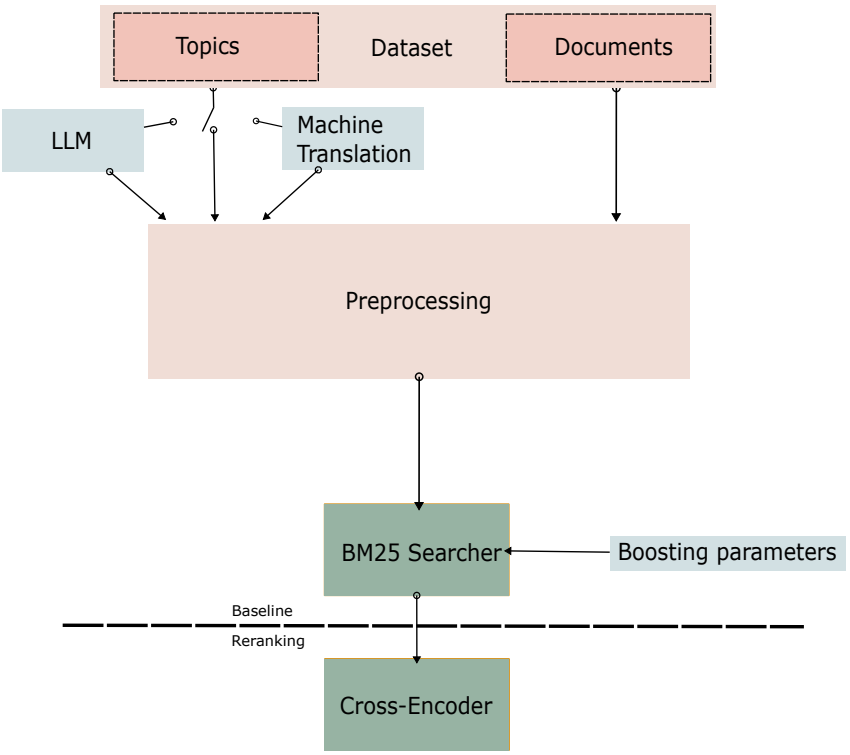

Figure 1: Overall system scheme

Reranking with a dense model The reranking phase makes use of a cross-encoder model in a pointwise manner. The cross-encoder associates the queries to each document that has been retrieved by the corresponding query with the sparse retriever, and for each of these pairs computes a new score and produces a final ranking.

Query generation We gauged the application of an LLM to generate queries that contain fewer terms than the original social media posts. The goal was to bridge the gap between the informal register used in social media and formal academic terminology by extracting a small set of important keywords from the original text, and expanding them with synonyms and other related terms. Prompt engineering has been applied to achieve compliant results (see Section 3.2).

Handling CLIR We propose a simple approach to CLIR that leverages machine translation to handle the German and French documents and topics, alongside the English ones.

The main difficulty in matching topics in German and French with the corresponding paper using lexical matching is that the vast majority of documents in the collection is written in English.

The approach we adopted was to simply employ a seq2seq[1] model to translate topics in German and French to English, and then matching the translated topics with the documents.
This has the obvious disadvantage that if social media posts implicitly reference papers in a language different from English, these would most likely not be retrieved. On the other hand, this allows for the retrieval of references to English papers, and results (see Section 5) show that performances are comparable to running the system on topics that are natively in English.

3.2. Implementation

The baseline of the IR system was implemented using the *Apache Lucene*[2] library in Java. This baseline

[1] "seq2seq" refers to models that convert input sequences into output sequences, useful for tasks such as machine translation [12]. Here, we are referring to an "encoder-decoder" model (see Section 3.2).
[2] https://lucene.apache.org/

retrieves an initial batch of documents through Okapi BM25 similarity.

Searching Strategies We evaluated two different searching strategies to retrieve the initial pool of candidate documents with the baseline pipeline.

The first approach presented Boolean OR logic behaviour operating over multiple document fields. This approach retrieves the results that contain at least one of the query tokens in the selected document fields. In particular, experiments may involve either two fields (*title* and *abstract*) or four fields (including *authors* and *venue*). We also distinguished the importance of the different fields by applying weights to them. We refer to this as *query boosting* or *boosting*. The corresponding weights were determined empirically and are the following:

- Title: 1.0
- Abstract: 1.1
- Authors: 0.33
- Venue: 0.44

This choice was motivated by an increase in performance, as shown in section 5, when giving more importance to the title and abstract than to the authors and venue.

The second approach considered a searching strategy that leverages the concepts of *term frequency* and *inverse document frequency* (as in TFIDF [13]), to extract the most prominents terms and retrieve documents similar to a given one [14]. The rationale behind this choice was given by the nature of topics, which are long enough to permit this approach. We will be referring to this as *MoreLikeThis* or *MLT* from the name used in the Apache Lucene library.

Fine tuning The cross-encoder used for reranking has been fine-tuned using the training and development datasets for Task 1 of CheckThat! 2026. The base model is `ms-marco-MiniLM-L6-v2` and can be found at: https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2. The training and validation datasets were composed by associating the relevant document instances to the corresponding entries in, respectively, the train and dev split of the CheckThat! Task 1 English dataset. The loss function is a two phase cached learning technique for contrastive learning. The model is fed with pairs of topics (anchor) and their relevant documents (positive instances). For any given anchor, all the other positive samples within the batch are treated as negative samples [15, 16]. To optimize memory, the gradient results are saved in a cache to be reused during the encoder's training [17]. As hyperparameters, the number of epochs has been set to 3 and the learning rate has been set to 2e-5.

LLM We implemented the query generation component with a resource-efficient, lower-parameter (1B) model, `Llama-3.2-1B-Instruct`. The model can be found at https://huggingface.co/unsloth/Llama-3.2-1B-Instruct. The submitted prompt, shown below, serves as the "instruction set" for the model:

- Role: System
- Content: "You are a professional search query optimizer. Your goal is to turn a search intent into 5-8 unique keywords separated by spaces. No numbers, no bullets, no repetition."

Example for the model:

- Role: User
- Content: "Extract keywords for: A recent study shows that transmission of COVID is higher in closed spaces rather than the outside."
- Role: Assistant
- Content: "covid transmission higher interior household inside outside"

Actual data:

- Role: User
- Content: f"Extract keywords for: {topic_text}"

Machine translation Machine translation from German and French to English was carried out using respectively the Opus MT German–English and French–English models by the Helsinki-NLP Research Group of the University of Helsinki. The models can be found at the following links:

- https://huggingface.co/Helsinki-NLP/opus-mt-de-en
- https://huggingface.co/Helsinki-NLP/opus-mt-fr-en

4. Experimental Setup

The repository with the codebase of this project can be found at:
https://bitbucket.org/upd-dei-stud-prj/seupd2526-sagasu.

Fine tuning of the cross-encoder model used for reranking was performed using the training and development splits of the dataset provided by CheckThat! for the Task 1 of 2026. Evaluation of system performances is done over the development and test splits of the same dataset.

Experiments, including inference with the reranker was run on a device with the following specifications:

- CPU: Intel(R) Core(TM) i5-8600K @ 3.6 GHz
- RAM: 16 GB DDR4
- GPU: NVIDIA GeForce RTX 3050 8 GB VRAM

Fine tuning of the cross-encoder model, machine translation of social media posts and generation of query using LLMs was performed by hosting notebooks on Google Colab (https://colab.research.google.com).

Performance of runs was evaluated using MRR@5 score.

5. Results

In the following, the best run results are presented together with the corresponding parameter configurations. In particular, Section 5.1 presents the preliminary analysis we conducted on the dev set to determined which runs should be submitted as official runs, while Section 5.2 presents the results for the official runs on the test set.

5.1. Preliminary analysis on the dev set

The first two runs were executed without any stemming techniques using the boolean searcher: the first without a stoplist, the second one using it; both results are presented in table 1.

Table 1
MRR@5 performance of system runs, with and without using a stoplist, on the English dev set.

Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
en	BOOLEAN	None	No	2	0.48268
en	BOOLEAN	None	Yes	2	0.48737

Simply looking at the MRR@5, a slight increase in performance can be noticed using a stoplist.

There appears to be no significant difference between the four stemming techniques (scores reported in table 2) used but, looking at the raw MRR@5, the Krovetz stemmer perfomed slightly better than the others on the English language. The experiment using the MoreLikeThis searching strategy led to a noticeable decrease in the MRR@5 (see table 3).

Table 4 shows the performance when searching over 4 fields and compares it with the approach that employs the LLM generated queries. While the run without the generated queries resulted in slightly worse performances simply looking at raw MRR@5, the LLM approach achieved significantly

Table 2
MRR@5 performance of system runs, over 4 different stemmers, on the English dev set.

Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
en	BOOLEAN	Krovetz	Yes	2	0.48865
en	BOOLEAN	Porter	Yes	2	0.48622
en	BOOLEAN	Snowball	Yes	2	0.48659
en	BOOLEAN	EnglishMinimal	Yes	2	0.48675

Table 3
MRR@5 performance of system runs using the MoreLikeThis based searcher, on the English dev set.

Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
en	MLT	Krovetz	Yes	2	0.43624

worse performance, being outperformed by every approach that does not employ the LLM generation component. Both this attempt and the MLT one are based on the idea of reducing the amount of words for the topics and extracting only the most important terms: the decrease in performance when applying these two approaches may be linked to this concept, meaning that the whole topics may better identify the source documents.

Table 4
MRR@5 performance of system runs with and without topics generated by an LLM, on the English dev set.

Language	Searcher	Stemmer	Stoplist	Fields	Generated	MRR@5
en	BOOLEAN	Krovetz	Yes	4	No	0.48399
en	BOOLEAN	Krovetz	Yes	4	Yes	0.37267

We then show the results when applying field boosting, reported in table 5.

Table 5
MRR@5 performance of system runs with field boosting, on the English dev set.

Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
en	BOOLEAN	Krovetz	Yes	4	0.49857
en	BOOLEAN	Snowball	Yes	4	0.49528

The query boosting approach leads to the best results achieved with the sparse baseline. We then applied the reranking with a cross-encoder to these runs leading to the best results overall. Similar results were found for the other two languages, with the only difference being that we did not attempt to use generated queries in German and French since the performances in English were subpar. Both results can be seen in table 6 and table 7. Performance after the application of the dense reranker is summarized in table 8.

Table 6
MRR@5 performance of system runs on the French dev set.

Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
fr	BOOLEAN	None	No	2	0.42728
fr	BOOLEAN	None	Yes	2	0.43678
fr	BOOLEAN	Krovetz	Yes	2	0.44440
fr	BOOLEAN	Porter	Yes	2	0.45427
fr	BOOLEAN	Snowball	Yes	2	0.45309
fr	BOOLEAN	EnglishMinimal	Yes	2	0.44596
fr	BOOLEAN	Krovetz	Yes	4	0.43979
fr	BOOLEAN	MLT	Yes	2	0.40399
The following system runs employed boosting techniques					
fr	BOOLEAN	Krovetz	Yes	4	0.46821
fr	BOOLEAN	Snowball	Yes	4	0.47616

Table 7
MRR@5 performance of system runs on the German dev set.

Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
de	BOOLEAN	None	No	2	0.31576
de	BOOLEAN	None	Yes	2	0.31865
de	BOOLEAN	Krovetz	Yes	2	0.31092
de	BOOLEAN	Porter	Yes	2	0.33428
de	BOOLEAN	Snowball	Yes	2	0.32893
de	BOOLEAN	EnglishMinimal	Yes	2	0.31334
de	BOOLEAN	Krovetz	Yes	4	0.31261
de	BOOLEAN	MLT	Yes	2	0.30307
The following system runs employed boosting techniques					
de	BOOLEAN	Krovetz	Yes	4	0.33506
de	BOOLEAN	Snowball	Yes	4	0.34149

Table 8
MRR@5 performance of system reranked boosted runs on all the dev sets.

Language	Searcher	Stemmer	Stoplist	Fields	Boosted	Reranked	MRR@5
en	BOOLEAN	Krovetz	Yes	4	Yes	Yes	0.57778(*)
fr	BOOLEAN	Snowball	Yes	4	Yes	Yes	0.55505
de	BOOLEAN	Snowball	Yes	4	Yes	Yes	0.41636

[*]Note that the English dev set was used as validation set for the fine tuning of the reranker and as such the result may be skewed.

5.2. Evaluation on the test set

The performance of the systems over the test set is summarized in tables 9, 10, 11 (note that we added an additional reranked run per language for comparison purposes). We performed two-way ANOVA with Tukey's HSD test for multiple pair-wise comparisons: the results can be found in figures 2, 3, 4. ANOVA shows that possibly no significant difference is present whether employing a stoplist or not, as well as between systems using different kinds of stemming. The run using the generated queries appears to be significantly worse than any other run; on the contrary, the reranking with the cross-encoder model yields performances that are significantly better than any other result achieved by the sparse baseline (also in this case, no significant difference is seen for reranked runs that use different stemming techniques, as is the case for normal runs).

In the end the runs submitted for evaluation were:

- English: `B-K-s-4f-b-re`, MRR@5 0.5606
- German: `B-S-s-4f-t-b-re`, MRR@5 0.5309
- French: `B-S-s-4f-t-b-re`, MRR@5 0.4379

6. Conclusions and Future Work

In the paper, we have validated hybrid approaches for implicit source retrieval. Our final approach ranks 24th out of 37 participants for Task 1 of the 2026 edition of CheckThat! Lab at CLEF. We also explored the possibility of integrating an LLM for query generation. By evaluating the runs under different parameter configurations, using both searchers, the best MRR@5 values reached for the test set were:

- 0.5606 for English language, using Krovetz stemmer;
- 0.5348 for French language, using Krovetz stemmer;
- 0.4443 for German language, using Krovetz stemmer.

The MRR@5 values obtained for the runs submitted for evaluation, prior to the release of the labels for the test set, were:

- 0.5606 for English language, using Krovetz stemmer;
- 0.5309 for French language, using Snowball stemmer;
- 0.4379 for German language, using Snowball stemmer.

All these results were obtained by considering four document fields, applying query boosting weights and performing re-ranking.
For future work, an option would be to evaluate the performance of a completely dense pipeline, using dual encoders to retrieve the initial batch of documents. Other improvements would be with respect to CLIR: in order not to lose references to document that are not natively English, a method that aggregates results from multiple languages can be implemented. Alternatively, machine translation can be applied on the collection documents as well.

Declaration on Generative AI

During the preparation of this work, Google Translate has been used to translate some passages from Italian to English. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

Table 9
MRR@5 performance of system runs on the English test set.

Name	Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
B-N-n-2f	en	BOOLEAN	None	No	2	0.47686
B-N-s-2f	en	BOOLEAN	None	Yes	2	0.48086
B-K-s-2f	en	BOOLEAN	Krovetz	Yes	2	0.48451
B-P-s-2f	en	BOOLEAN	Porter	Yes	2	0.48200
B-S-s-2f	en	BOOLEAN	Snowball	Yes	2	0.48293
B-EM-s-2f	en	BOOLEAN	EnglishMinimal	Yes	2	0.48308
B-K-s-4f	en	BOOLEAN	Krovetz	Yes	4	0.46260
MLT-K-s-2f	en	MLT	Krovetz	Yes	2	0.44516
The following system runs employed boosting techniques						
B-K-s-4f-b	en	BOOLEAN	Krovetz	Yes	4	0.48799
B-S-s-4f-b	en	BOOLEAN	Snowball	Yes	4	0.48650
The following system run employed query generation techniques						
B-K-s-4f-gen	en	BOOLEAN	Krovetz	Yes	4	0.31972
The following system runs employed boosting and reranking techniques						
B-K-s-4f-b-re	en	BOOLEAN	Krovetz	Yes	4	0.56055
B-S-s-4f-b-re	en	BOOLEAN	Snowball	Yes	4	0.55870

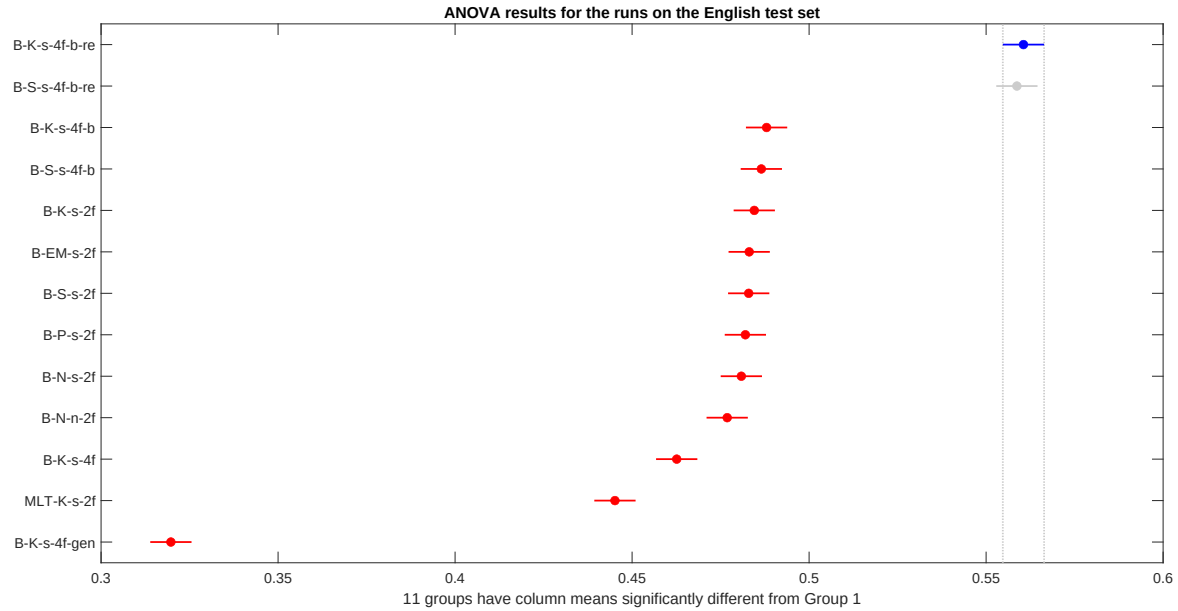

Figure 2: ANOVA results for the runs on the English test set

Table 10
MRR@5 performance of system runs on the French test set.

Name	Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
B-N-n-2f-t	fr	BOOLEAN	None	No	2	0.42490
B-N-s-2f-t	fr	BOOLEAN	None	Yes	2	0.43060
B-K-s-2f-t	fr	BOOLEAN	Krovetz	Yes	2	0.44087
B-P-s-2f-t	fr	BOOLEAN	Porter	Yes	2	0.43803
B-S-s-2f-t	fr	BOOLEAN	Snowball	Yes	2	0.43903
B-EM-s-2f-t	fr	BOOLEAN	EnglishMinimal	Yes	2	0.43667
B-K-s-4f-t	fr	BOOLEAN	Krovetz	Yes	4	0.43664
MLT-K-s-2f-t	fr	MLT	Krovetz	Yes	2	0.40933
The following system runs employed boosting techniques						
B-K-s-4f-t-b	fr	BOOLEAN	Krovetz	Yes	4	0.45736
B-S-s-4f-t-b	fr	BOOLEAN	Snowball	Yes	4	0.45240
The following system runs employed boosting and reranking techniques						
B-K-s-4f-t-b-re	fr	BOOLEAN	Krovetz	Yes	4	0.53486
B-S-s-4f-t-b-re	fr	BOOLEAN	Snowball	Yes	4	0.53088

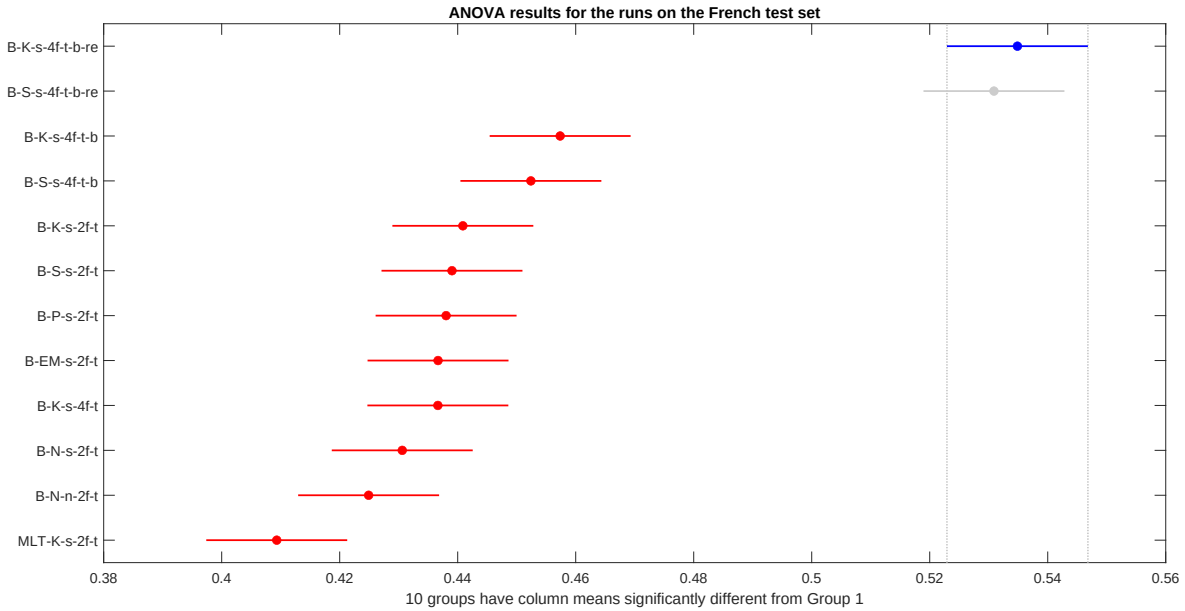

Figure 3: ANOVA results for the runs on the French test set

Table 11
MRR@5 performance of system runs on the German test set.

Name	Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
B-N-n-2f-t	de	BOOLEAN	None	No	2	0.29471
B-N-s-2f-t	de	BOOLEAN	None	Yes	2	0.30557
B-K-s-2f-t	de	BOOLEAN	Krovetz	Yes	2	0.32388
B-P-s-2f-t	de	BOOLEAN	Porter	Yes	2	0.32384
B-S-s-2f-t	de	BOOLEAN	Snowball	Yes	2	0.32616
B-EM-s-2f-t	de	BOOLEAN	EnglishMinimal	Yes	2	0.31130
B-K-s-4f-t	de	BOOLEAN	Krovetz	Yes	4	0.32112
MLT-K-s-2f-t	de	MLT	Krovetz	Yes	2	0.30105

The following system runs employed boosting techniques

Name	Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
B-K-s-4f-t-b	de	BOOLEAN	Krovetz	Yes	4	0.34098
B-S-s-4f-t-b	de	BOOLEAN	Snowball	Yes	4	0.34254

The following system runs employed boosting and reranking techniques

Name	Language	Searcher	Stemmer	Stoplist	Fields	MRR@5
B-K-s-4f-t-b-re	de	BOOLEAN	Krovetz	Yes	4	0.44431
B-S-s-4f-t-b-re	de	BOOLEAN	Snowball	Yes	4	0.43788

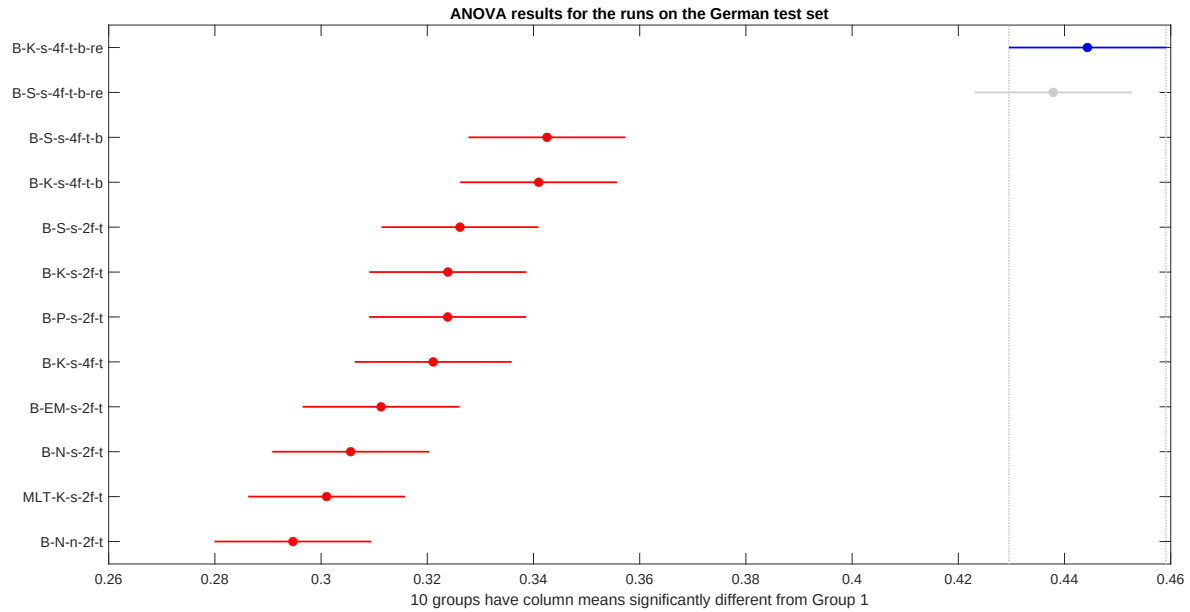

Figure 4: ANOVA results for the runs on the German test set

References

[1] 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: [2], 2026.

[2] 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.

[3] CheckThat!, CheckThat! — checkthat.gitlab.io, https://checkthat.gitlab.io/clef2026/task1/, 2026.

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

[5] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, Okapi at TREC–3, in: D. K. Harman (Ed.), The Third Text REtrieval Conference (TREC-3), National Institute of Standards and Technology (NIST), Special Publication 500-225, Washington, USA, 1995, pp. 109–126. URL: https://www.researchgate.net/publication/221037764_Okapi_at_TREC-3.

[6] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[7] P. J. Sager, A. Kamaraj, B. F. Grewe, T. Stadelmann, Deep Retrieval at CheckThat! 2025: Identifying Scientific Papers from Implicit Social Media Mentions via Hybrid Retrieval and Re-Ranking, in: [18], 2025.

[8] H. Liu, A. Soroush, J. G. Nestor, E. Park, B. Idnay, Y. Fang, J. Pan, S. Liao, M. Bernard, Y. Peng, C. Weng, Retrieval augmented scientific claim verification, JAMIA Open 7 (2024) ooae021. URL: https://doi.org/10.1093/jamiaopen/ooae021. doi:10.1093/jamiaopen/ooae021.

[9] T. Schreieder, M. Färber, Claim2Source at CheckThat! 2025: Zero-Shot Style Transfer for Scientific Claim-Source Retrieval, in: [18], 2025.

[10] WHO, Standardization of terminology of the pandemic a(h1n1)2009 virus, https://cdn.who.int/media/docs/default-source/influenza/global-influenza-surveillance-and-response-system/nomenclature/standardization-of-terminology-of-the-pandemic.pdf?sfvrsn=667f8727_8, 2011.

[11] Unicode, UAX #29: Unicode Text Segmentation — unicode.org, https://unicode.org/reports/tr29/, 2025.

[12] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, CoRR abs/1409.3215 (2014). URL: http://arxiv.org/abs/1409.3215. arXiv:1409.3215.

[13] K. Spärck Jones, A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation 28 (1972) 11–21.

[14] A. S. Foundation, MoreLikeThis (Lucene 10.4.0 queries API) — lucene.apache.org, https://lucene.apache.org/core/10_4_0/queries/org/apache/lucene/queries/mlt/{M}ore{L}ike{T}his.html, 2026.

[15] M. L. Henderson, R. Al-Rfou, B. Strope, Y. Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, R. Kurzweil, Efficient natural language response suggestion for smart reply, CoRR abs/1705.00652 (2017). URL: http://arxiv.org/abs/1705.00652. arXiv:1705.00652.

[16] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, CoRR abs/1807.03748 (2018). URL: http://arxiv.org/abs/1807.03748. arXiv:1807.03748.

[17] L. Gao, Y. Zhang, Scaling deep contrastive learning batch size with almost constant peak memory usage, CoRR abs/2101.06983 (2021). URL: https://arxiv.org/abs/2101.06983. arXiv:2101.06983.

[18] G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, CLEF 2025, Madrid, Spain, 2025.

A. Note on query generation with LLMs

Query generation led to generally poor results as shown in section 5. This can partially be attributed to the low number of parameters of the model that has been employed. In fact, some of the results produced are bad quality, presenting hallucinations and frequent repetition of terms. These are nonetheless a minority, and most generated queries are good quality, as they only a few keywords and expanding

them with synonyms and related terms. Whether performance is degraded or improved varies greatly from query to query. Table 12 shows examples of queries of both categories and their effect on retrieval.

Table 12
Generated queries examples with the position at which the relevant document was retrieved

Bad quality result

Original topic & generated query	**Position**
Original topic	0
Generated query: virus spread science research study scientific findings reputable sources reputable sources reputable sources reputable sources reputable sources reputable sources reputable sources reputable sources reputable sources reputable sources reputable sources	0

Good quality results

Original topic	1
Generated query: COVID-19 pandemic twisted misinformation government corruption medical institutions media bias international organizations	18
Original topic	23
Generated query: hospital-acquired infections uk study vaccines reduce hospital infections spread patient-to-patient healthcare workers	12

In general, performance improvement with generated queries happens more rarely then degradation, which clearly leads to yielding worse results overall. As shown in table 13, generated queries increase the amount of instances in which the relevant document is not retrieved at all, and performance improvement only happens for a small portion of instances.

Table 13
Comparison of effects of applying query generation.

	dev	test
Total number of topics	3906	6077
Number of relevant documents not found with generated queries	1503	2488
Number of relevant documents not found with original topics	1209	1950
Number of instances in which performance is improved by query generation	534	764