Team SeRRa at CheckThat! CLEF 2025: Sequential Re-Ranking in a Scientific Claim Source Retrieval Pipeline

Notebook for the CheckThat! Lab at CLEF 2025

Guilherme A. Marchetti[1,*], Gil Rocha[2] and Henrique Lopes Cardoso[1]

[1]LIACC, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
[2]INESC-ID, Lisboa, Portugal

Abstract

Indirect reference resolution represents a complex challenge within the field of information retrieval. To promote the advancement of new methods and technologies to tackle this class of challenges, the 8th edition of the CheckThat! Lab at the CLEF conference [1] proposed, as one of its shared tasks, the SciWeb Claim-Source Retrieval task [2], in which participants were challenged to correctly identify the research paper indirectly referenced by a tweet. This paper presents a multi-step pipeline for document retrieval based on a tweet containing an indirect mention of it. The process begins by selecting the top 200 candidate documents employing a pre-trained Sentence-BERT model for dense retrieval. These candidates are then re-ranked using a binary classification model trained with negative sampling. Finally, a third model determines the final ranking through pairwise comparisons of the top 10 re-ranked documents. This final model was trained using document pairs selected by the earlier models to ensure highly correlated documents are used for contrast with the gold reference. The combination of multiple models, trained with different negative sampling strategies, resulted in a robust retrieval quality, achieving an MRR@5 in the development dataset of 0.7024, compared to 0.5522 from the BM25 baseline. In the subtask's evaluation stage, our methodology achieved the 8th highest score, with an MRR@5 of 0.61 in the test dataset.

Keywords

Dense Retrieval, BERT Re-Ranking, Multi-Step Document Retrieval, Negative Sampling Strategies

1. Introduction

Indirect reference resolution represents a complex challenge within the field of information retrieval. In this type of retrieval, systems aim to accurately identify a specific document referenced in a free-form text, such as a tweet, that contains no explicit link to the target. To promote the advancement of new methods and technologies to tackle this class of challenges, the 8th edition of the CheckThat! Lab at the CLEF conference [1, 3] proposed, as one of its shared tasks, the SciWeb Claim-Source Retrieval task [2]. Participants were challenged to correctly identify the research paper indirectly referenced in a tweet, among a corpus containing over 8,000 documents. The tweets to be used as indirect reference were separated into 3 different splits, the *train* split with 12,853 tweets, the *dev* split with 1400 tweets, and the *test* split with 1465 tweets. The tweets in the *train* and *dev* splits also included the gold document reference.

The proposed task is particularly challenging, since the texts contained in the tweets do not follow any structure, as can be seen in the examples shown in Figure 1. In this paper, we describe our submission to the shared task. We propose a method composed of a three-step pipeline, adapting previous multi-stage approaches [4] to take advantage of new developments on transformer-based language models. This pipeline performs a sequence of re-ranking and reductions on the candidate documents set until the final result is obtained.

This sequence of document filters allows us to employ different types of models at each step. By using larger and more powerful models on a decreasing number of documents, this multi-stage approach

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
up202408600@up.pt (G. A. Marchetti); gil.rocha@inesc-id.pt (G. Rocha); hlc@fe.up.pt (H. Lopes Cardoso)
0009-0002-6616-5900 (G. A. Marchetti); 0000-0001-8252-7292 (G. Rocha); 0000-0003-1252-7515 (H. Lopes Cardoso)

Tweet Text	Reference Paper Title	Reference Paper Abstract
Peer-reviewed in the New England Journal of Medicine regarding Delta (B.1.617.2): •Pfizer is ~90% effective •AstraZeneca is ~70% effective. This falls in line with vaccine efficacy of other variants. Yes, the vaccines ARE indeed effective against Delta.	Effectiveness of Covid-19 Vaccines against the B.1.617.2 (Delta) Variant	BACKGROUND: The B.1.617.2 (delta) variant of the severe acute respiratory syndrome coronavirus 2 [...]
Published in the journal Antiviral Research, the study from Monash University showed that a single dose of Ivermectin could stop the coronavirus growing in cell culture -- effectively eradicating all genetic material of the virus within two days.	The FDA-approved drug ivermectin inhibits the replication of SARS-CoV-2 in vitro	Although several clinical trials are now underway to test possible therapies, the worldwide response to the COVID-19 outbreak has been largely limited to monitoring/containment [...]

Figure 1: Examples of the provided tweets and reference research paper [2].

provides a balance between computing effort and retrieval quality. Another advantage of employing different models at each stage is the possibility of training each one with a different strategy. We designed a negative sampling strategy that, by leveraging the representation learned by the model used in the first steps, selects particularly challenging contrastive samples to train the model used in the last step.

These more powerful models, paired with this negative sampling strategy to fine-tune them, allow us to achieve a robust document retrieval capability, surpassing the baseline, measured by the BM25 metric, by a significant margin. Besides the main task results, we also demonstrate through an ablation study that the challenging contrastive examples selected are central to the overall performance of the pipeline.

This paper is organized as follows: Section 2 presents some recent research papers related to information retrieval based on pre-trained language models; Section 3 describes our proposed methodology; Section 4 presents the results of our evaluations on the provided development dataset; and Section 5 provides our conclusions and directions for future work.

2. Related Work

Recent advances in natural language processing, particularly the development of transformer-based models [5], have significantly influenced the field of information retrieval. A growing trend involves the use of dense retrieval methods [6, 7], which rely on pre-trained language models to produce semantically rich embeddings for both documents and queries, leading to improved retrieval performance. These embeddings can be generated using either a shared model to encode both documents and queries [8] or distinct models [9], though both approaches typically employ similarity measures such as the dot product or cosine similarity to rank documents during retrieval.

In addition to single-stage methods, multi-step retrieval pipelines have become increasingly common [4]. In these pipelines, a set of documents goes through a series of ranking algorithms, reducing the total size of the set at each step. This sequential reduction of the set of candidate documents enables the use of more computationally intensive models in later stages, achieving a good compromise between complexity and execution time.

Besides the pipeline and model architecture used, the way training data is selected is of great importance to the overall performance of the retrieval methodology. This can be evidenced by the RocketQA [10] training approach. In this work, the authors focus on the problem of selecting the hard negative examples for passage retrieval. Throughout their work, the authors experiment with different data augmentation and de-noising techniques to improve the retrieval quality, ultimately arriving at a multi-stage training approach, where one model not included in the retrieval pipeline is specifically trained for selecting challenging examples to train the final model.

Our proposed approach for indirect document retrieval differs from prior work in some important ways. First, we employ a newer model for the pairwise comparison, one that has a larger context

window, as detailed in Section 3.3. This allows us to present more information for the model to analyze. The second, and more important, difference concerns the negative sampling methods used to train the models. In [4], the same sampling process is used to train the models across the stages, beginning with BM25 filtering followed by random sampling. We have devised a more robust strategy, which takes advantage of results in earlier stages to select highly similar documents. This ensures that the model is trained on highly similar documents, presenting a greater challenge and promoting the learning of complementary representations relative to the other models in the pipeline. Another advantage over other methods is that the model used to select the negative samples is included in the pipeline during inference, instead of just being used to select training samples [10]. Given that during inference the candidate document set will resemble the ones used for training, we hypothesize that our approach should display better generalization than previous works.

3. Methodology

The methodology proposed follows the already established multi-step ranking architecture [4]. The first step involves reducing the total document corpus D_0 of approximately 8000 documents to a smaller set D_1 with k_1 documents, using SentenceBert [8] for dense retrieval (Section 3.1). Next, a fine-tuned SciBert [11] model, trained for binary classification, computes a relevance score f_c used to re-rank and further reduce D_1 to a set D_2 with k_2 documents (Section 3.2). Finally, a third model [12] is responsible for the final ordering of documents (Section 3.3). This model evaluates all possible pairwise combinations of $\langle doc_i, doc_j \rangle \in D_2$, calculating an f_p score to determine the most relevant documents for the original tweet. An illustration of the full pipeline is shown in Figure 2. In the following sections, we provide detailed descriptions for each step of our proposed approach.

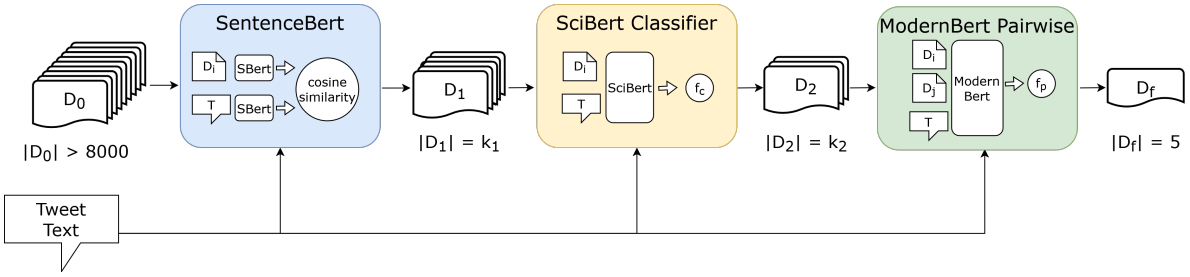


Figure 2: Overview of the full retrieval pipeline.

3.1. First Step: Pre-filter with SentenceBert

The goal of the first step is to reduce the total number of documents so that we can apply more complex ranking methods in the reduced set. Given that the goal of this first step is to filter the relevant documents, without requirements in terms of ranking the documents yet, we focus on improving the *Recall@k* at this stage (i.e., reducing the set of candidate documents, while keeping the most relevant documents). While one of the most used methods for this pre-selection of documents is the BM25 method, during our experiments, we found that a pre-trained SentenceBert (S-Bert) model [1] performed better on the provided Dev dataset, as shown in Table 1.

For filtering the documents, we employ the standard approach of dense retrieval. First, we encode all the documents in the corpus using the model. To perform this encoding, different fields from the research papers were used, namely title, author, abstract, and journal. Each field is concatenated into a single string, using the "[SEP]" token as a separator between them. This final string is then presented to the S-Bert model, which produces the n-dimensional representation for the corresponding document.

[1] multi-qa-mpnet-base-dot-v1

Table 1
Comparison between BM25 and S-Bert for the pre-selection of documents.

Ranking Method	MRR@5	Recall@100	Recall@150	Recall@200
BM25	0.5522	0.7828	0.8042	0.8121
SentenceBert (Dot Product)	**0.5402**	0.87	0.8935	0.9071
SentenceBert (Cosine Similarity)	0.5247	0.8707	0.895	**0.9107**

After computing the representations for all documents, we apply the same process to the tweet being evaluated, using the same embedding model. Each document embedding is then compared to the tweet embedding using a cosine similarity scoring function. Curiously, although the employed model was originally optimized for use with dot product scoring, cosine similarity increases the Recall@200 from 0.9071 to 0.9107, while presenting a slight decrease in MRR@5 from 0.5402 to 0.5247. Since we are more concerned with Recall at this stage, we use cosine similarity to score and rank all documents, selecting the top k_1 to include in the D_1 candidates set.

3.2. Second Step: Fine-tuned Relevance Classifier Re-ranking

In the second step, we re-rank the reduced document set D_1 using a model fine-tuned for binary classification. We present the tweet paired with information from a research paper to the model that, in turn, is tasked with classifying whether the document is relevant to the tweet. Similarly to the previous step, the input is the concatenation of different fields from the research paper, but in this case, it is also preceded by the "[CLS]" token and the tweet text. A representation of the model and inputs used is shown in Figure 3.

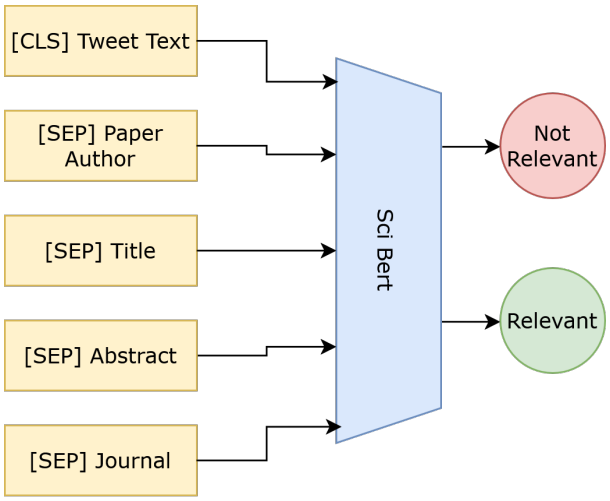


Figure 3: Using the SciBert model to score the documents.

During training, we present the model with correctly labeled positive and negative examples of tweet-document pairs. To build the set of training examples, we perform a random negative sampling of the training dataset provided for Task 4b [2] of the CheckThat! Lab [1]: for each provided tweet with the gold reference, we randomly selected 5 other documents from the corpus to use as negative cases for classification. This results in a total of over 68,000 examples. Starting from a pre-trained SciBert [11], the model was fine-tuned using this set of tweet-document pairs, using an 80/20 train-validation split, for 2 epochs, with a learning rate of 2e-5, with the AdamW optimizer.

During inference, the model computes a relevance score f_c for each document concerning a given tweet, defined as:

$$f_c(tweet, doc_i) = logit_{\text{relevant}}(tweet, doc) \quad (1)$$

where $logit_{\text{relevant}}$ is the logit value corresponding to the *relevant* class, as produced by the trained model. Using this fine-tuned model, each document in the D_1 set is evaluated against the selected tweet, and the top k_2 documents are chosen for the final ranking step.

3.3. Final Step: Pairwise Document Relevance Evaluation

In the final step, we perform pairwise comparisons between all documents $doc_i \in D_2$. Although this step also uses a binary classification model, the architecture is very different from the previous step. The input of the model is extended to include both documents being compared, using the same fields from each one. The model's task is determining which document, A (the first in the sequence) or B (the second), is more closely related to the tweet. An illustration of this model is shown in Figure 4.

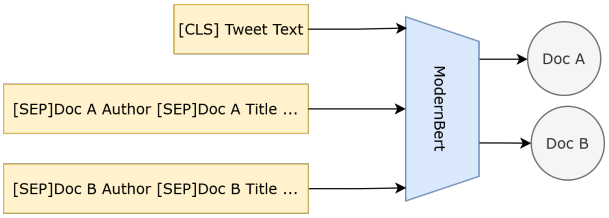


Figure 4: Comparing a pair of documents against the tweet using the ModernBert model.

The base model used for classification also had to be modified. To accommodate the larger number of tokens resulting from combining the two documents, this step employs the ModernBert [12] model, which supports a context window of up to 8192 tokens.

We use a different negative sampling strategy to build the training examples for this model, focusing on challenging examples that are closely related to the gold reference. We start by reusing the filtering and re-ranking models from earlier stages to pre-select the top 5 candidate documents for each tweet. To ensure that the model learns to distinguish the gold reference from its most similar alternatives, we retain only the tweets in which the gold reference is included in the top 5 candidate documents retrieved by the models employed in earlier stages. For each training example, we create all possible pairs between the gold document and the remaining documents. To mitigate positional bias, each gold-neighbor pair is duplicated with the order of documents reversed. This process results in 8 pairings for each of the selected sets, for a total of over 75,000 training examples. The model was trained for 2 epochs, using an 80/20 train-validation split, with a learning rate of 2e-6 and the AdamW optimizer.

To compute the final ranking, each document doc_i is compared to every other element doc_j of the document set D_2. The final score $f_p(tweet, doc_i)$ is the average value of the pairwise comparison of the documents:

$$f_p(tweet, doc_i) = \frac{\sum logit_A(tweet, doc_i, doc_j)}{|D_2|}, \forall doc_j \neq doc_i \in D_2 \tag{2}$$

where $logit_A(tweet, doc_i, doc_j)$ is the logit value corresponding to the doc_i being the more closely related document to the reference $tweet$, as produced by the trained model. To avoid extra computations, we make the simplifying assumption that:

$$logit_A(tweet, doc_i, doc_j) = logit_B(tweet, doc_j, doc_i) \tag{3}$$

That is, we assume symmetry in model predictions. This allows us to compute both scores (A and B) in a single pass, avoiding redundant evaluations of the same document pair in reverse order. This assumption should not introduce significant error in the scoring function, since this positional bias is accounted for during training.

After computing f_p for all the documents in the set, the final candidate set of documents is determined by selecting the top 5 documents sorted by their assigned score.

Table 2
Comparison between BM25 and each step of the retrieval pipeline.

Ranking Method	MRR@5 (Dev)	Recall@5 (Dev)
BM25	0.5522	0.625
Step 1	0.5247	0.6264
Step 1 and 2	0.6725	0.7771
Step 1, 2 and 3	**0.7024**	**0.78**

4. Experiments and Results

This section is organized into three parts. Section 4.1 presents the main results from the shared task, including MRR@5 scores on both the development and evaluation datasets. In Section 4.2, we analyze how varying the number of selected documents, k_1 and k_2, impacts the overall performance of the pipeline. Finally, Section 4.3 reports the results of an ablation study designed to show that the negative sampling strategy used to train the pairwise comparison model is essential for enabling it to learn complementary representations of the documents, thus improving the overall result of the pipeline.

4.1. Main task results

For the shared task, the organizers [2] provided the participants with two distinct sets of tweets: *train* and *dev*. The split contained about 14,000 and 1,400 tweets, respectively, both annotated with the correct document to be retrieved. To evaluate the performance of the proposed methodology, the models were trained exclusively with the *train* set, with the *dev* being used exclusively to evaluate the performance of the retrieval pipeline.

To make sure that every step of the pipeline contributes meaningfully to the final result, we report the MRR@5 and Recall@5 at every step of the pipeline. Table 2 presents the main results and comparisons with the baseline.

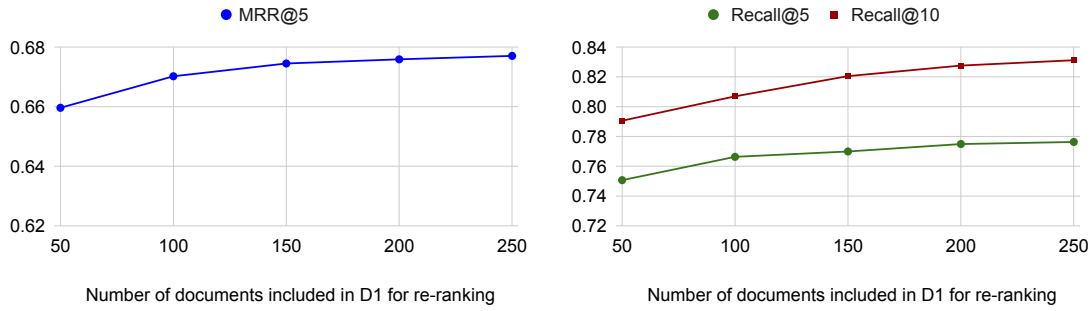
The complete pipeline achieves an MRR@5 score of 0.7024. As expected, the MRR@5 increases after each step of the pipeline, with all steps contributing meaningfully to the overall retrieval quality. It should also be noted that the drop in performance that occurs in the first step, compared to the baseline, is expected, as we prioritize recall at this step.

After determining that we have reached a good configuration for the pipeline, we performed document retrieval using the *evaluation* dataset. To ensure a fair comparison between approaches to the task, this dataset contains only the tweets to be used, without the gold reference. Since we do not have access to the gold reference, the organization of the shared task was responsible for evaluating the results, calculating only the MRR@5 of each participant. We present the results from our approach, together with the best-performing metric and baseline in Table 3.

Table 3
Results of the evaluation phase of the competition.

Ranking Method	MRR@5 (Eval)
BM25	0.43
SeRRa (ours)	0.61
Top Result	**0.68**

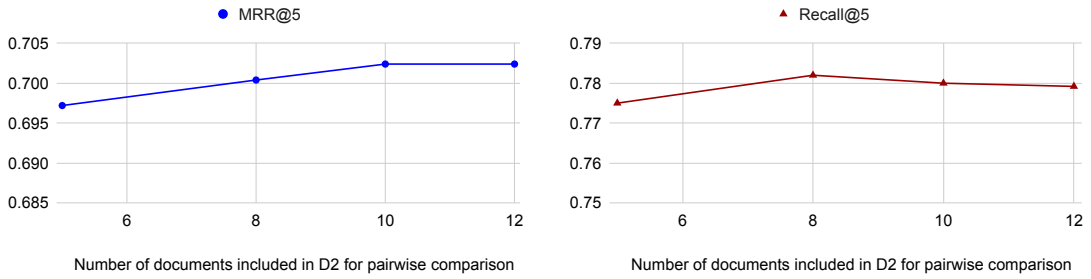
Out of 30 participants in the shared task, our approach ranked in the 8th position, surpassing the BM25 baseline by a good margin. It is interesting to note that, while both the baseline and our method had a lower performance in the *evaluation* set compared to the *dev* set, ours presented a smaller relative reduction, dropping from 0.70 to 0.61 against 0.55 to 0.43 of the BM25 baseline, which may indicate a better generalization potential.



(a) Evolution of MRR@5 metric.

(b) Evolution of Recall@k metric.

Figure 5: Re-ranking performance with different values of k_1.



(a) Evolution of MRR@5 metric.

(b) Evolution of Recall@k metric.

Figure 6: Pairwise comparison performance with different values of k_2, with fixed $k_1 = 200$

4.2. Impact of different k_i values on ranking quality

In addition to the main results, we have also evaluated the impact that different values of k_1 and k_2 have on each phase, which are reported in Figures 5 and 6, respectively. The size of the document sets D_1 and D_2 has a big impact on the performance of each phase, as a small number of documents may result in the correct reference being excluded, while a set containing a large number of documents may be too computationally demanding to compute. Besides the MRR@5 metric, we also report the Recall@k to gauge how often the re-rankers were dropping the golden document.

Although increasing the number of documents improves the overall retrieval quality, the gains suffer a diminishing return effect, i.e., after a certain threshold, adding more documents to the set will not provide any meaningful increase in quality. This is particularly relevant for the pairwise classifier (step 3), where computational costs rise sharply due to the need to evaluate all pairwise combinations. Considering this trade-off, we found that using $k_1 = 200$ and $k_2 = 10$ provides the best balance between performance and computational effort.

4.3. Ablation study

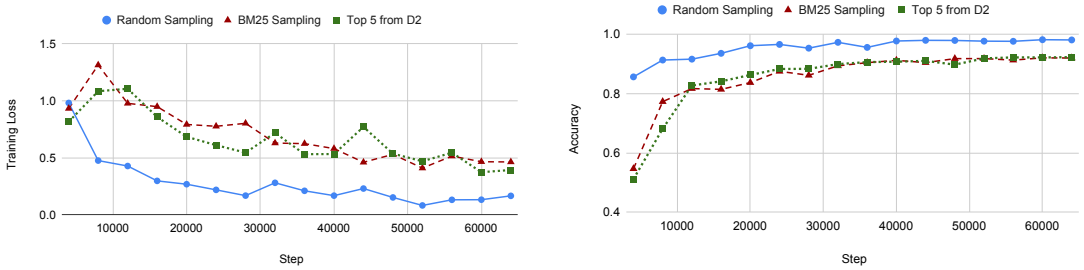
To measure the effectiveness of the negative sampling technique employed, we have performed an ablation study, where the same parameters of the pipeline are kept, but the negative sampling strategy used to select the training data of the pairwise comparison model (step 3) is varied. Three different sampling techniques were explored: random sampling, using the top 5 candidates from the BM25 ranking, and selecting the top 5 documents from the D_2 candidate set computed by the first steps of the pipeline, as detailed in Section 3.3. The performance of each model is presented in Table 4.

The results clearly show that, by taking advantage of the representations learned in previous steps (i.e., Top 5 from D_2), the pairwise comparison model learns more complementary representations, leading to better performance in the overall pipeline. The hypothesis that the learned representations

Table 4
Ablation study of different negative sampling techniques.

Sampling Method	MRR@5 (Dev)	Recall@5 (Dev)
Random	0.6043	0.7442
BM25	0.6192	0.7371
Top5 from D_2 (ours)	**0.7024**	**0.78**

are complementary is further supported by analyzing the progression of the learning loss and accuracy during model training, as shown in Figure 7.



(a) Evolution of Training Loss.

(b) Evolution of Accuracy.

Figure 7: Evolution of loss and accuracy during training of pairwise document relevance evaluation for different sampling methods.

There are a few interesting points to note in the evolution of the metrics during training. As expected, just performing a random negative sampling results in unrelated pairs of documents. This results in the model quickly learning how to differentiate between the two, denoted by how fast the accuracy converges to a high value. On the other hand, applying any filter prior to selecting the negative examples results in more difficult to learn examples, as can be seen by the similar evolution in training loss and accuracy.

Although the evolution of the non-random sampling methods is similar, there are differences between them. Examining the loss during training in Figure 7a, it is possible to see that the BM25 sampling results in a more monotonic decay, when compared to our sampling method's oscillating behavior. It is also possible to see that using our sampling method, the model has a significantly worse accuracy at the start of training, but achieves a similar accuracy to the BM25 sampling at the end of the training period.

During inference, the results are reversed, as already demonstrated in Table 4. The high accuracy achieved during training using random negative sampling provides no meaningful information for ranking the documents in D_2, negatively impacting the results of previous steps. Similarly, even though the models achieved a similar accuracy during training using either the BM25 or our sampling methods, only the model trained with the harder examples can improve the final result.

The results of these studies seem to indicate that, by selecting dissimilar examples for comparison, the model learn to differentiate them in a lexical manner. Since the first step of the retrieval pipeline already performs a more robust selection than lexical similarity, the model does not contribute meaningfully to further rank the documents. On the other hand, by selecting highly similar documents, our sampling method forces the model to identify more nuanced semantic differences between the documents. Even though it is more difficult to identify these differences, they allow the model to learn representations that are complementary to the previous steps in the pipeline, improving the overall document retrieval performance.

4.4. Qualitative Analysis

To better understand the candidate document retrieval behavior, we conducted a tweet-level evaluation of MRR@5 scores on the development dataset. Analyzing individual queries allows us to observe which

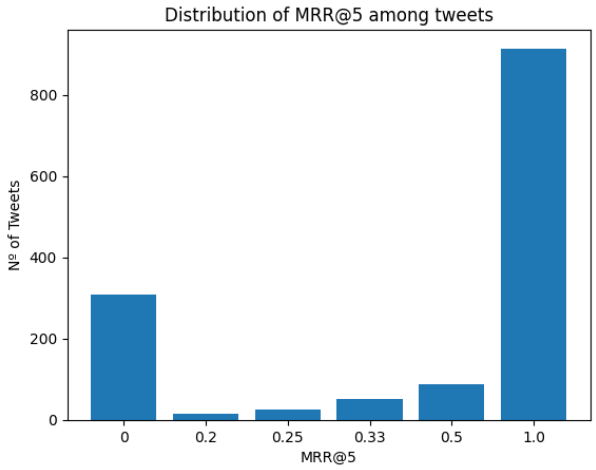


Figure 8: Distribution of MRR@5 scores over the tweets in the Development Dataset

Tweet Reference	Abstract Passages
ah good point on uk using asians. but the best cdc study found 81% protection against omicron for 2 doses mrna <180d, 57% 2 doses >180d, 90% for 3 doses. but would you expect any other v platforms to protect against a far different variant than the vaccine?	[...] after dose 2, 76% ≥ 180 days after dose 2, and 94% ≥14 days after dose 3. Estimates of VE for the same intervals after vaccination during Omicron variant predominance were 52%, 38%, and 82%, respectively. [...]
	[...] higher than that after a second dose; however, VE waned with increasing time since vaccination. During the Omicron period, VE against ED/UC visits was 87% during the first 2 months after a third dose and decreased to 66% among those vaccinated 4-5 months earlier; [...]
Antibody dependent enhancement/pathogenic priming led to all previous coronavirus spike protein vaccines failing safety testing and FDA approval. It's utterly misleading that everyone is being told these shots are safe when that is absolutely not true.	[...] antibody-dependent enhancement (ADE). Previous respiratory syncytial virus and dengue virus vaccine studies revealed human clinical safety risks related to ADE, resulting in failed vaccine trials. [...]
	Antibody dependent enhancement (ADE) of infection is a safety concern for vaccine strategies. [...] the N-terminal domain (NTD) of the SARS-CoV-2 spike protein facilitate virus infection in vitro, but not in vivo [...]

Figure 9: Examples of incorrectly retrieved documents with their original tweet reference

documents are being prioritized and identify potential areas for improvement. The distribution of MRR@5 scores across the development set is shown in Figure 8.

Interestingly, the MRR@5 distribution is heavily skewed toward the extreme values of 0 and 1, corresponding to cases where the gold reference was either missing from the candidate set or ranked as the top result, respectively. This distribution suggests that the pairwise comparator (step 3) is highly effective at identifying the most relevant document, and that its performance is primarily limited by the recall in the earlier retrieval steps.

In Figure 9, we selected some of the incorrectly retrieved tweet references to analyze in further detail. The top row contains one tweet that scored an MRR@5 of 0, accompanied by the gold reference (in green) and the top-rated candidate document (in red). The second row follows the same structure, but uses a tweet with an MRR@5 of 0.2 instead. From these examples, we note that even in the cases with the lowest scores, the documents retrieved are still correlated to the gold reference. In the first example, we can see that both documents discuss the effectiveness of the COVID-19 vaccines and how it varies according to the number of doses applied, while the second mentions the effect of antibody-dependent enhancement. This seems to indicate a cap on the possible differentiation between documents using only information contained in them. To further improve the relevance ranking, it may be necessary to

include more information about the context in which the tweet was created, so it is possible to better capture the information needs of the tweet creator.

5. Conclusion and Future Work

In this paper, we presented our approach for task 4b of the CheckThat! Lab at CLEF 2025. This task consisted of correctly identifying and retrieving research papers, using as input a free-form tweet that indirectly referenced the study. The proposed retrieval methodology consisted of a three-step filter, where at each step a different language model is used to filter and re-rank the documents, until a final ordered list of the 5 most relevant documents is computed. The proposed methodology displays a robust performance, achieving the 8th-best score in the task leaderboard, with an MRR@5 rating of 0.61 on the evaluation phase of the task.

Besides the main metric evaluation, we also demonstrated the impact that varying the amount of documents included in each filtering step has on the performance of the overall retrieval pipeline. Another contribution of our approach was presenting an effective way to incorporate the knowledge learned in previous steps to improve the performance of the final ranking, by using the retrieved documents from previous steps to train a more powerful pairwise classifier model.

In future work, we aim to explore ways to better extract the information needs conveyed in the tweet text and investigate how this extra information could be included in the pipeline to further improve the performance.

Acknowledgments

This work was financially supported by UID/00027 - Artificial Intelligence and Computer Science Laboratory (LIACC), funded by Fundação para a Ciência e a Tecnologia, I.P./ MCTES through national funds. Gil Rocha was supported by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (i.e., the Center For Responsible AI), and also by the Fundação para a Ciência e Tecnologia, specifically through the project with reference UIDB/50021/2020 (DOI: 10.54499/UIDB/50021/2020).

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Grammarly for grammar and spelling checking, and for improving writing style. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. Alam, J. M. Struß, T. Chakraborty, S. Dietze, S. Hafid, K. Korre, A. Muti, P. Nakov, F. Ruggeri, S. Schellhammer, V. Setty, M. Sundriyal, K. Todorov, V. V., The CLEF-2025 CheckThat! Lab: Subjectivity, Fact-Checking, Claim Normalization, and Retrieval, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2025, pp. 467–478. doi:10.1007/978-3-031-88720-8_68.

[2] S. Hafid, Y. S. Kartal, S. Schellhammer, K. Boland, D. Dimitrov, S. Bringay, K. Todorov, S. Dietze, Overview of the CLEF-2025 CheckThat! lab task 4 on scientific web discourse, in: [1], 2025, pp. 467–478. doi:10.1007/978-3-031-88720-8_68.

[3] F. Alam, J. M. Struß, T. Chakraborty, S. Dietze, S. Hafid, K. Korre, A. Muti, P. Nakov, F. Ruggeri, S. Schellhammer, V. Setty, M. Sundriyal, K. Todorov, V. Venktesh, Overview of the CLEF-2025 CheckThat! Lab: Subjectivity, fact-checking, claim normalization, and retrieval, in: J. Carrillo-de Albornoz, J. Gonzalo, L. Plaza, A. García Seco de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina,

G. Faggioli, N. Ferro (Eds.), Experimental IR meets multilinguality, multimodality, and interaction. Proceedings of the sixteenth international conference of the CLEF association (CLEF 2025), 2025.

[4] R. Nogueira, W. Yang, K. Cho, J. Lin, Multi-stage document ranking with BERT, CoRR abs/1910.14424 (2019). URL: http://arxiv.org/abs/1910.14424, arXiv: 1910.14424 tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 25 Feb 2025 13:21:07 +0100.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, december 4-9, 2017, long beach, CA, USA, 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html, tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Thu, 21 Jan 2021 15:15:21 +0100.

[6] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, E. Grave, Unsupervised Dense Information Retrieval with Contrastive Learning, Trans. Mach. Learn. Res. 2022 (2022). URL: https://openreview.net/forum?id=jKN1pXi7b0.

[7] W. X. Zhao, J. Liu, R. Ren, J.-R. Wen, Dense Text Retrieval Based on Pretrained Language Models: A Survey, ACM Trans. Inf. Syst. 42 (2024) 89:1–89:60. URL: https://dl.acm.org/doi/10.1145/3637870. doi:10.1145/3637870.

[8] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410/. doi:10.18653/v1/D19-1410.

[9] 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: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), 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.

[10] Y. Qu, Y. Ding, J. Liu, K. Liu, R. Ren, W. X. Zhao, D. Dong, H. Wu, H. Wang, RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 5835–5847. URL: https://aclanthology.org/2021.naacl-main.466/. doi:10.18653/v1/2021.naacl-main.466.

[11] I. Beltagy, K. Lo, A. Cohan, SciBERT: A Pretrained Language Model for Scientific Text, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. URL: https://aclanthology.org/D19-1371/. doi:10.18653/v1/D19-1371.

[12] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, I. Poli, Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, CoRR abs/2412.13663 (2024). URL: https://doi.org/10.48550/arXiv.2412.13663. doi:10.48550/ARXIV.2412.13663, arXiv: 2412.13663 tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Thu, 23 Jan 2025 22:31:11 +0100.