CLEF 2025 – CheckThat Subtask 4b Approach of Team TUWien_AIR2025_G15

Kartik Arya
email1@example.com

Teodor Esaiasson
teodor.esaiasson@gmail.com

Nooreldin Lasheen
e12302427@student.tuwien.ac.at

Daniel Levin
e12433760@student.tuwien.ac.at

Philipp Moeßner
e12412779@student.tuwien.ac.at

Abstract

Abstract — Placeholder for the abstract. (Add a short summary of the project here later.)

1. CLEF Challenge

Social media platforms such as X.com (formally known as Twitter) are mostly used for informal and fast-paced public discussions. When scientific topics are debated on these platforms, stated claims often lack transparency regarding their sources or general evidence. Therefore, fact-checking scientific discussions on social media platforms is of major importance. Task 4 of the CheckThat-Lab at CLEF 2025 takes on this problem, specifically for the COVID-19 domain. We participated in task 4b which asks its participants to develop an information retrieval system which outputs a ranked list of scientific documents for a given tweet making a statement about supposedly conducted research. The better the tweet's content matches a document's content, the higher it shall be ranked in the output list. In order to achieve this, the organizers provide a corpus of 7718 documents which are possibly mentioned in the tweets. Added to that, 12853 tweets and the corresponding correct document mentioned by the tweet are provided as training data. 1400 tweets (+ correct document) are provided as development data to internally test and evaluate the trained approach. Finally, 1446 tweets are provided as test data without the corresponding correct documents. The ranked lists for each of the test data tweets are then submitted to the challenge, evaluated by the organizers and serve as the final result of a team on the leaderboard. The following report examines our approach for the demanded IR system.

$$rsv_q = \sum_{t \in q} \log\left(\frac{N+1}{df_t}\right) \cdot \left(\frac{(k_1+1).tf_{td}}{k_1.\left((1-b)+b.\left(\frac{L_d}{L_{avg}}\right)\right)+tf_{td}} + \delta \right)$$

Figure 1: BM25 formula [1]

2. General Approach

We split the given task into 3 sub-tasks: First (3), we build a traditional IR model, meaning an approach that is solely based on individual term frequency and (inverse) document frequency of terms, namely BM25. Secondly (4), we make use of representation learning approaches to build abstract embeddings for both documents and queries (tweets) to represent them in a more holistic and context-aware way. Finally (5), we use these embeddings as input for different neural reranking approaches, which rerank and potentially improve the BM25 preranked lists.

3. Traditional IR Model

3.1. BM25

From the organizers of the task, a BM25 model was given as a baseline. Since later reranking approaches are built on the initial documents retrieved by the BM25 model, this offered a starting point for improving the final results of the predictions.

3.2. What is BM25?

Best match 25 (BM 25) is a basic approach to information retrieval that is based on the concept of counting how rare terms are in the whole document corpus and how often the terms appear in any of the documents. To do this, an inverted index is created of all terms across all documents which contains information about the frequency of the term in a given document. Too use BM25 in a information retrieval task, a score is calculated for each term (or rather token) in a query together with a each document in the corpus. The score for all tokens in combination with that document is then summed up as the total score for that query document pair. In an intuitive sense, the score is based on the frequency of the token in a specific document multiplied by how rare the token is across all documents.

There are many variations of BM25, but the mathematical formula for the one used in this project can be seen in Figure 1.

3.3. Modifications and Contributions

3.4. Final Model and Performance

Our final BM25 pipeline combined several of the most effective strategies identified during the iterative process that was based on the findings of each member of the group. We used the BM25Plus model with synonym normalization (using the COVID-19 thesaurus [2]), careful text cleaning,

and retention of important numerical terms. Named entity recognition was omitted, as it did not provide additional benefits in this context.

This approach resulted in a substantial improvement over the original baseline. As summarized in Table 2, our best BM25 configuration achieved an MRR@5 of 61.76% on the development set, compared to 55.20% for the baseline. The improvements can be attributed to domain-specific synonym normalization, optimized preprocessing, and the selection of the most suitable BM25 variant for our data.

Our contributions lie in the systematic evaluation of preprocessing strategies, the integration of external domain resources (thesaurus [2]), and the empirical comparison of BM25 variants. These steps addressed the core challenges of vocabulary mismatch, style gaps between queries (tweets) and documents (scientific articles) and noisy input, enhancing the retrieval effectiveness of the traditional IR model.

4. Representation Learning

To not only rely on exact word / token matching like in our traditional IR model, we transition to token embeddings for the reranking approach. There are already a lot of pretrained models for embedding creation that also roughly fit our domain. Also, the challenge does not provide enough training data to train a state-of-the-art transformer model from scratch. We therefore choose a transfer learning setup and finetune a pretrained BERT instance.

4.1. BERT Finetuning

BERT model (in ColBERT)	**MRR@5 (dev)**
SciBERT[1]	68.03%
BioBERT[2]	67.56%
PubMedBERT[3]	67.05%

Table 1: Performance of different BERT versions in the ColBERT reranking architecture

We test our best performing reranking approach ColBERT (section 5) on different domain-specific pretrained BERT instances. Here, we focus on BERT models that cover the biomedical scientific domain instead of e.g. Twitter-specific BERT models since only the queries (which are tweets) would benefit from such a BERT model while a more general biomedical specific model can cover both the documents' and tweets' content. Table 1 summarizes our results and reveals that SciBERT fits the task's data best.

System	MRR@5 (dev)
ColBERT ConvKNRM MatchPyramid BERTCat Transformer Kernel	68.03%
BM25 (modified)	61.76%
BM25 (baseline)	55.20%

Table 2: Best achieved MRR@5 scores on the dev dataset for different tested reranking approaches compared to BM25 preranking

5. Neural Re-Ranking

5.1. ColBERT

We achieved the highest MRR@5 score on the development dataset with a ColBERT reranking architecture (Table 2). Its reranking mechanism is quite simple compared to other approaches (Section 5.2). Queries and documents are both passed through the same encoder resulting in token-level embeddings for each query and each document. To compute a relevance score of a document for a given query, a match matrix is built consisting of cosine similarity values for each query token – document token pair. The final score is aggregated from the matrix by simply summing up (over query dimension) the maximum values of each row (document dimension).

5.1.1 Modifications and Contributions

As mentioned in section 4 we use a SciBERT instance as encoder for both queries and documents after finetuning it to our specific domain (COVID-19 related tweets and documents). This finetuning is the only part of ColBERT training since the scoring / reranking mechanism of ColBERT does not have to be trained as it is a simple matrix aggregation rule. Still, in the context of ColBERT, the SciBERT finetuning loss function is adjusted: The training goal is to obtain a higher score for a relevant than for a non-relevant document while satisfying a predefined margin between the scores. These scores are already computed as the ColBERT-specific match matrix aggregation instead of e.g. simple cosine similarity of CLS vectors.
To adjust ColBERT to the challenge's task, we tune its hyperparameters as follows: First, we try finetuning for different amounts of epochs and find that training for 6 epochs gives the best MRR@5 score on the dev dataset (table 3). Two additional important training parameters are the number of negative document samples for a query and the margin. We perform a grid search (with a fixed number of epochs at 6 and a learning rate at 2e-5) to find the best combination of these parameters and find that a margin of 0.5 and 1 negative sample achieve the highest MRR@5 score (table 4).

[1]Huggingface: `allenai/scibert_scivocab_uncased`
[2]Huggingface: `dmis-lab/biobert-v1.1`
[3]Huggingface: `microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract`

Finetuning epochs	MRR@5 (dev)
0	56.94%
2	63.51%
6	**64.02%**
10	62.26%

Table 3: Snippet from finetuning SciBERT (ColBERT reranking) for a different amount of epochs

Margin	Negatives	MRR@5 (dev)	Last Epoch Loss
0.3	1	0.6537	8.0212
0.3	2	0.6477	16.7373
0.3	4	0.6118	29.3307
0.4	1	0.6574	12.9181
0.4	2	0.6503	21.4273
0.4	4	0.6360	27.8490
0.5	**1**	**0.6610**	**9.6190**
0.5	2	0.6393	18.1380
0.5	4	0.6259	31.5070

Table 4: Grid search for finding best margin and number of negative samples (table has cutoff at margin 0.5 since we saw MRR@5 go down again with higher margins)

Lastly, we check whether the length of the BM25 preranked lists influence the MRR@5 score and find that a length of 100 documents results in the best score (5).

Preranked List Size	MRR@5 (dev)
25	67.75%
50	68.03%
100	**68.06%**
200	67.58%

Table 5: MRR@5 on dev set for different preranked list sizes (number of documents) using ColBERT (SciBERT) with margin 0.5

5.1.2 Evaluation

Overall, the ColBERT reranking approach increased the MRR@5 (dev) score by almost 13% points compared to the BM25 baseline and around 5% points compared to our modified BM25 model. None of our other tested approaches achieves this increase. Thus, we use this ColBERT configuration (trained in 6 epochs with margin 0.5, 1 negative sample and a learning rate of 2e-5) in our ranking pipeline for the test data and our final submission to the leaderboard.

5.2. Other Tested Approaches

In addition to our primary methods (BM25 and ColBERT), we explored several alternative approaches in our experiment workflow. We began by evaluating different transformer-based models, such as SBERT, and MiniLM, to understand their baseline effectiveness. Within the MiniLM family, we compared the 12-layer (MiniLM-L12-v2) and the 6-layer distilled version (all-MiniLM-L6-v2). interestingly, the 6-layer model outperformed the larger model in terms of MRR, highlighting that smaller, distilled models can be more efficient than larger ones especially when the complexity of the dataset does not need deep contextual representation.

During input preparation, we encountered items in the collection particularly abstracts and titles that exceeded the 512-token input limit of BERT-based models, sometimes reaching up to 1372 tokens. We tried truncating these inputs, but this did not improve performance and, in some cases, led to a loss of relevant content.We also experimented with standard preprocessing techniques such as lowercasing, punctuation removal, and stopword filtering. These had varying effects across models: while Cross encoder MiniLM benefited from preprocessing, SBERT's performance declined, as an unexpected result given the importance of contextual understanding in such models. This illustrates that each model has its own way of handling input and preprocessing; some approaches enhanced MRR for certain models, while the same methods led to reduced MRR in others.

Overall, these additional experiments provided valuable insights, but none of the alternative approaches outperformed ColBERT in terms of MRR, leading us to select it as the final model.

6. Final Results

7. Discussion

- Maybe ColBERT worked best because it only needed fine-tuning (of SciBERT) and no from-scratch training like a CNN approach, etc. -¿ From-scratch training works best with a lot of training data, but the challenge organizers did not provide that.
- We could have tried another ColBERT version making it a dual encoder setup where queries are encoded by a Twitter specific BERT instance and documents by SciBERT -¿ the learning objective would have been to map the embeddings to the same latent space

References

References

[1] A. Trotman, A. Puurula, B. Burges, *Improvements to BM25 and Language Models Examined*, November 2014, https://doi.org/10.1145/2682862.2682863 (accessed May 20, 2025).

[2] Patricia Fener, *COVID-19 Thesaurus*, https://loterre.istex.fr/C0X/en/ (accessed May 20, 2025).