DS@GT at CheckThat! 2025: Exploring Retrieval and Reranking Pipelines for Scientific Claim Source Retrieval on Social Media Discourse

Notebook for the CheckThat! Lab at CLEF 2025

Jeanette Schofield[1], Shuyu Tian[1], Hoang Thanh Thanh Truong[1] and Maximilian Heil[1,*]

[1]Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332

Abstract

Social media users often make scientific claims without citing where these claims come from, generating a need to verify these claims. This paper details work done by the DS@GT team for CLEF 2025 CheckThat! Lab Task 4b Scientific Claim Source Retrieval which seeks to find relevant scientific papers based on implicit references in tweets. Our team explored 6 different data augmentation techniques, 7 different retrieval and reranking pipelines, and finetuned a bi-encoder. Achieving an MRR@5 of 0.58, our team ranked 16th out of 30 teams for the CLEF 2025 CheckThat! Lab Task 4b, and improvement of 0.15 over the BM25 baseline of 0.43. Our code is available on Github at https://github.com/dsgt-arc/checkthat-2025-swd/tree/main/subtask-4b.

Keywords

cite-worthiness, science-related discourse, social media, data augmentation, retrieval and reranking, CEUR-WS

1. Introduction

The spread of health misinformation on social media has increasingly become a problem in recent years creating the need to substantiate claims made by users on social media [1]. CheckThat! Task 4b, Scientific Claim Source Retrieval, attempts to solve this problem by asking users to retrieve the most relevant scientific articles from a collection set that support user claims on social media. This is a hard challenge to solve as the language used on social media often differs greatly from that used in the scientific articles that make up the collect set [2].

The DS@GT team explored finetuning a bi-encoder, 6 different data augmentation techniques, and 7 different retrieval and reranking pipelines. Achieving an MRR@5 of 0.58, our team ranked 16th out of 30 teams for the CLEF 2025 CheckThat! Lab Task 4b, an improvement of 0.15 over the BM25 baseline of 0.43.

2. Related Work

First offered in 2018, CheckThat! 2025 is the eighth offering of the CheckThat! Lab which focuses on technology that helps automate the journalistic verification process [3, 4]. Task 4b, Scientific Claim Source Retrieval, predicts which scientific articles are most relevant to user discourse related to COVID-19 on Twitter. Prior CheckThat! labs have offered COVID-related tasks. The goal of Task 1, Identifying Relevant Claims in Tweets, in 2022 was to predict which COVID-19 related tweets in a dataset were worth fact-checking [5, 6]. Task 4, Detecting hero, villain, and victim from memes, from CheckThat! 2024 asked participates to identify the role of entities within memes, many of which were related to COVID-19 [7].

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

jschofield8@gatech.edu (J. Schofield); stian40@gatech.edu (S. Tian); htruong47@gatech.edu (H. T. T. Truong); mheil7@gatech.edu (M. Heil)

0009-0000-0669-8962 (J. Schofield); 0000-0001-6444-651X (S. Tian); 0009-0007-4130-3349 (H. T. T. Truong); 0009-0002-6459-6459 (M. Heil)

Before a source can be retrieved to support a claim, one first needs to determine if a claim is being made. Cite-worthiness refers to the problem of determining if there are missing references to scientific results in text. Designed for cite-worthiness detection in scientific text, the CiteWorth dataset [8] contains 1.1M English-language sentences, with 375K sentences designed as cite-worthy. Models trained using the CiteWorth dataset were found to perform poorly when tasks with scientific citations in social media discourse [2]. Consisting of 1,261 tweets and a subset of the SciTweets dataset, SCiteTweets is dataset created for detecting citations in social media discourse [9].

3. Data

Task 4b organizers provided three datasets: two sets of query tweets, one for training and one for development, and the CORD-19 paper collection. The training (train) dataset contains 12,853 tweets, while the development (dev) set includes 1,400 tweets. Both datasets share the same data structure with the following three columns:

1. post_id - A unique identifier for the tweet
2. tweet_text - The textual content of the tweet
3. cord_uid - A unique identifier corresponding to a paper in the CORD-19 collection

The CORD-19 paper collection consists of 7,718 academic papers with 17 columns. Some of the key columns are:

1. cord_uid - A unique identifier for the paper for linking with tweet queries in the train and dev set
2. title - The title of the paper
3. abstract - The abstract of the paper
4. authors - A list of the paper's authors
5. journal - The journal where the paper was published
6. publish_time - The publication date

Word clouds in Figures 1 and 2 show that words such as "patients", "COVID-19", "infection", and "risk" frequently appear in the set of query tweets and the CORD-19 paper collection. Words such as "study" and "vaccine" appear often in the tweets dataset, but are not as prevalent in paper abstracts. Terms like "disease" and "pandemic" appear more often in the CORD-19 paper abstracts than in the query tweets.

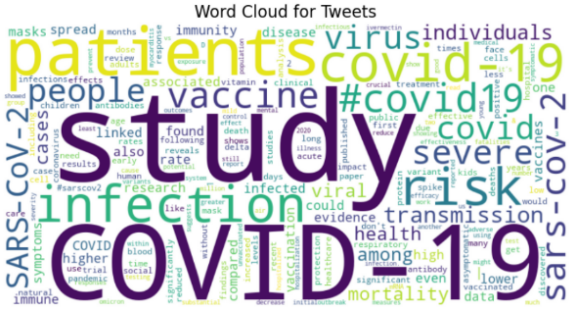

Figure 1: Word Cloud of Tweets in dataset

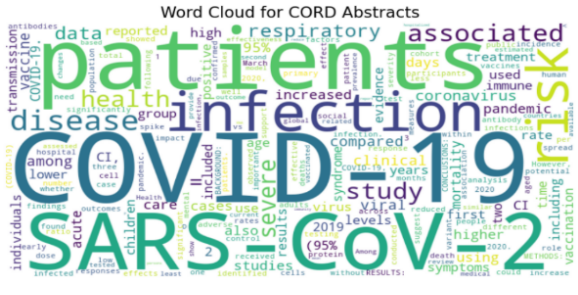

Figure 2: Word Cloud of Abstracts in dataset

4. Methodology

For this task, we explored multiple retrieval and reranking pipelines, experimenting with BM25 and bi-encoders for retrieval and exploring 7 different models for reranking. Multiple data augmentation experiments were performed during the retrieval stage to see if modifying the tweet data would improve results.

4.1. Mean Reciprocal Rank (MRR)

The Mean Reciprocal Rank at k (MRR@k) was the official metric for the CheckThat! 2025 Task 4b [3]. For a set of N queries, the MRR is calculated as:

$$MRR@k = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i}$$

where $rank_i$ is the position of the first correct and relevant document in the top-k retrieved list for the i-th query. If the relevant document does not appear within the top-k results, the reciprocal rank for that query is treated as zero.

4.2. Retrieval and Reranking Pipeline: First Attempt

Our first attempt at a retrieval and reranking pipeline used BM25 for retrieval, building on the organizers' baseline. BM25, or Best Match 25, ranks documents based on how many query terms appear in each document [10]. BM25 is often used to return a subset of documents from a large corpus due to it's speed. After the initial subset of documents are retrieved, more computationally expensive reranking algorithms are applied to improve upon the initial results. In our first pipeline, a cross-encoder trained on the MS Marco dataset (cross-encoder/ms-marco-MiniLM-L6-v2) was used for reranking [11, 12].

In all pipelines created, 100 documents were retrieved. Prior to retrieval, tweets and documents were converted to lowercase, tokenization was applied by splitting sentences up into words (i.e. on spaces), and stop words were removed. Data from the tweet_text column was used for the query set. For the CORD-19 paper collecition, we combined the title of the paper with the abstract.

4.3. Exploring Data Augmentation

After setting up an initial retrieval and reranking pipeline, data augmentation techniques were explored. For Task 4b, the goal was to find the journal article that best matched a user tweet from Twitter. Language used in social media discourse tends to be very different than that used in academic writing. Social media discourse often takes a more informal tone whereas academic writing uses formal language.

With this in mind, we explored various means of augmenting the tweet_text in the query dataset: rewriting tweets using formal (i.e. scientific) language, concatenating the original tweet text with the rewritten tweets, and replacing the original tweet with science-related keywords used in the tweet.

Two experiments replaced the original tweet text with rewritten text. The first experiment, "Replace w/ Formal Rewritten", used the prompt "Rewrite the input using formal language" to generate the rewritten tweets. Upon noticing that some of the tweets in the query set were not written in English, a second experiment was performed. "Replace w/ English Formal Rewritten" used the prompt "Rewrite the input using formal English language."

Three data augmentation experiments involved combining the original tweet text with the rewritten text. "Concat w/ Formal" combined the original tweet with tweets were rewritten using "formal language" prompt. "Concat w/ English formal" combined the original tweet with tweets rewritten using the "formal English language" prompt. "Concat w/ All" combined the original tweet, the tweet rewritten using "formal language", and the tweet rewritten using "formal English language."

The experiment "Replace w/ Keywords" used the prompt "Return a list of only science-related keywords in the tweet." The motivation was to see whether or not our pipeline would improve by using only science-related keywords to retrieve relevant documents.

Table 1 provides an example of the augmented data for each experiment.

In all data augmentation experiments, BM25 was used for retrieval and cross-encoder/ms-marco-MiniLM-L6-v2 was used for reranking. gpt-4o was used in all experiments to rewrite tweet data.

Table 1
Data Augmentation Experiments and Generated Tweet Text.
Data generated from the tweet with a post_id of 3491 in the query set

Data Augmentation Experiment	Tweet Text
Original Dataset	Bile salts in gut and liver pathophysiology
Replace w/ Formal Rewritten	Bile salts in the pathophysiology of the gastrointestinal tract and hepatic systems.
Replace w/ English Formal Rewritten	The role of bile salts in the pathophysiology of the gastrointestinal tract and liver.
Concat w/ Formal	Bile salts in gut and liver pathophysiology Bile salts in the pathophysiology of the gastrointestinal tract and hepatic systems.
Concat w/ English Formal	Bile salts in gut and liver pathophysiology The role of bile salts in the pathophysiology of the gastrointestinal tract and liver.
Concat w/ All (Formal & English Formal)	Bile salts in gut and liver pathophysiology The role of bile salts in the pathophysiology of the gastrointestinal tract and liver. Bile salts in the pathophysiology of the gastrointestinal tract and hepatic systems.
Replace w/ Keywords	Bile, salts, gut, liver, pathophysiology

4.4. Experimenting with Reranking Models

After our data augmentation techniques, we changed the retrieval stage to use BM25-PyTorch instead of BM25. This was done to make use of GPU and run the pipeline faster. With BM25-PyTorch as our retrieval model, we explored 7 different reranking models to see how they performed.

Three of these reranking models were trained on the MS Marco dataset: cross-encoder/ms-marco-MiniLM-L-6-v2, tomaarsen/reranker-msmarco-ModernBERT-base-lambdaloss, and tomaarsen/reranker-msmarco-MiniLM-L12-H384-uncased-lambdaloss [11, 12, 13, 14]. MS Marco is a widely used question answer dataset featuring 100,000 Bing questions with huamn generated answers.

In an effort to explore models trained on different datasets, we tested our pipeline on 3 models trained on the GooAQ dataset which contains 5 million Google queries and 3 million answers: akr2002/reranker-ModernBERT-base-gooaq-bce, tomaarsen/reranker-ModernBERT-large-gooaq-bce, and tomaarsen/reranker-NeoBERT-gooaq-bce [15, 16, 17, 18].

Our last experiment for exploring retrieval and reranking pipelines used Google's T5 (Text-to-Text Transfer Transformer) model which is trained on the C4 dataset which was developed by Google and Meta via scraping the web [19, 20]. We used AnswerDotAI's Reranker library to implement this model [21, 22].

4.5. Using Bi-encoders for Retrieval

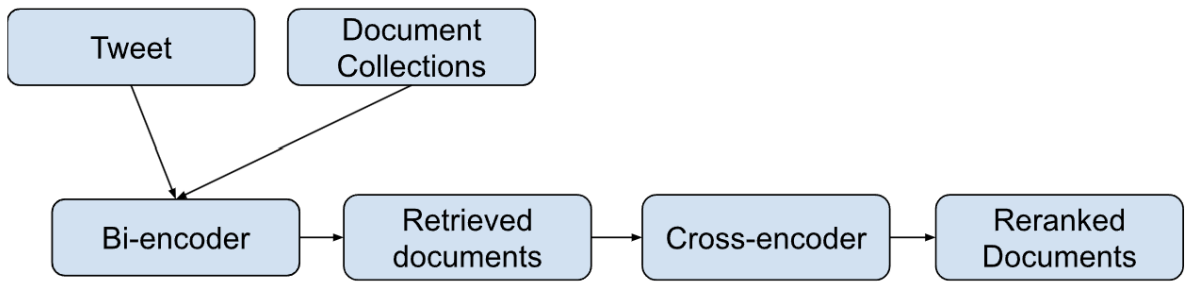

Figure 3: Two-stage document retrieval pipeline.

In this section, we focus on the bi-encoder and cross-encoder approaches using bi-encoder/msmarco-distilbert-base-v4 and cross-encoder/ms-marco-MiniLM-L-6-v2, respectively [23, 12]. Both are provided

by the Sentence-Transformers framework [24]. We aim to leverage the speed of bi-encoders to narrow down a smaller pool of document candidates and then use the precision of cross-encoders to enhance document retrieval accuracy (see Figure 3).

In the first stage, the pipeline loads a data set of query tweets and a corpus of scientific papers comprising titles and abstracts. These inputs are processed by the bi-encoder model (msmarco-distilbert-base-v4) to generate dense vector embeddings. Then, we perform a semantic search to retrieve the top 100 most similar documents for each tweet based on cosine similarity.

Since the cross-encoder is computationally expensive, this initial retrieval step helps reduce the relevant scientific documents to save resources. In the second stage, these top 100 documents are reranked using the cross-encoder (ms-marco-MiniLM-L-6-v2) to improve retrieval accuracy.

We evaluate both stages using MRR@k to allow a direct comparison between the initial retrieval and reranking performance.

4.6. Bi-encoder Finetuning

Building on the strong performance of the bi-encoder and cross-encoder pipeline, we finetuned the bi-encoder to further improve retrieval quality.

To expand the training data, we sample sentence-pairs to generate a silver dataset using contextual text augmentation, following the strategy proposed in Augmented SBERT [25]. We load tweets paired with its corresponding scientific document, which includes the document's title and abstract.

Next, we process the input data using a BERT-based augmenter (ContextualWordEmbsAug with bert-base-uncased) to insert contextually appropriate words. These augmented tweet–document pairs are saved as silver examples to diversify data and improve the generalization of the model.

After augmenting data, we train our bi-encoder (msmarco-distilbert-base-v4) through two phases. We firstly train only on the train data set and then train on the silver dataset, measured via MRR@100 on the dev set.

After data augmentation, we finetune our bi-encoder model (msmarco-distilbert-base-v4) in two phases. In the first phase, we train the model exclusively on the original training dataset. In the second phase, we continue training on the silver dataset generated through augmentation.

Both models are trained and evaluated using the same procedure. We employ the SentenceTransformerTrainer interface with a contrastive cosine similarity loss, and evaluate model performance using Mean Reciprocal Rank at 100 (MRR@100) on the dev set. The best models with highest MRR@100 are saved at each phase.

Together, this two-phase training strategy enables the model to first learn from gold-labeled data and then improve generalization through silver training examples generated via contextual augmentation. This progressive finetuning pipeline is designed to enhance retrieval performance while maintaining robustness on unseen tweet–document pairs.

5. Results

For the final submission for Task 4b, our team used BM25-Pytorch for retrieval and T5 for reranking. No data augmentation was applied. This pipeline achieved an MRR@5 of 0.58 which improved over the baseline of 0.43 by 0.15. Our team ranked 16 out of 20 teams. The leaders, SourceSniffers, achieved an MRR@5 of 0.68.

5.1. Data Augmentation

As shown in table 2, data augmentation experiments that replaced the text of the original tweet performed worse than the baseline on reranking tasks, with the MRR@5 decreasing by 0.06 or more. Experiments that combined the original tweet with additional data showed a slight improvement over the baseline with an increase in MRR@5 of roughly 0.03 after retrieval. After reranking, the experiment "Concat w/ Formal" and "Concat w/ All" improved over the baseline by 0.01.

Table 2
Data Augmentation Results
BM25 was used for retrieval and the cross-encoder ms-marco-MiniLM-L-6-v2 was used for reranking

Data Augmentation Step	MRR@5 after Reranking	MRR@5 after Retrieval
None	0.5521	0.6028
Replace w/ Formal Rewritten	0.4915	0.5183
Replace w/ English Formal Rewritten	0.5112	0.5366
Concat w/ Formal	0.5823	0.6106
Concat w/ English Formal	0.5859	0.6092
Concat w/ All (Formal & English Formal)	0.5812	0.5618
Replace w/ Keywords	0.4280	-

5.2. Reranking Models

Table 3 shows the results for experimenting with various reranking models. In all experiments, BM25-Pytorch was used the as the model for retrieval. Of the 7 models used for reranking, only the cross-encoder ms-marco-MiniLM-L-6-v2 model and the T5 model improved upon the initial retrieval results. The 3 models trained on the MS Marco dataset performed significantly better than those trained on the GooAQ dataset.

Table 3
Experimenting with Reranking Models Results
BM25-Pytorch was used for retrieval on all experiments.
*Submitted to CLEF for Task 4b

Reranking Model	MRR@5 After Retrieval	MRR@5 After Reranking
ms-marco-MiniLM-L-6-v2	0.6300	0.6474
reranker-msmarco-ModernBERT-base-lambdaloss	0.6300	0.6219
reranker-msmarco-MiniLM-L12-H384-uncased-lambdaloss	0.6300	0.4194
reranker-ModernBERT-base-gooaq-bce	0.6300	0.2437
reranker-ModernBERT-large-gooaq-bce	0.6300	0.5471
reranker-NeoBERT-gooaq-bce	0.6300	0.0563
T5*	0.6300	0.6590

5.3. Bi-encoder Finetuning

Results for bi-encoder finetuning can be found in table 4. Without finetuning, the MRR@5 using a bi-encoder for retrieval was 0.428. Unfortunately, all 3 finetuning experiments performed worse than the bi-encoder without any finetuning. The gold dataset with hard negatives from silver data achieved an MRR@5 of 0.407 which is 0.011 less than the regular bi-encoder.

Table 4
Finetuning Bi-encoder Results
Cross-encoder model ms-marco-MiniLM-L-6-v2 was used for reranking on all experiments

Finetuned On	MRR@5 After Retrieval	MRR@5 After Reranking
No Finetuning	0.428	0.612
Gold dataset	0.178	0.445
Gold dataset + silver augmented data	0.343	0.568
Gold dataset + hard negatives from silver data	0.407	0.616

6. Discussion

6.1. Data Augmentation

For data augmentation, all 3 experiments that replaced the original tweet text with rewritten text caused the MRR@5 to decrease. This suggests that the original tweets contained context that was lost by rewriting the tweet. This appears to be especially true for the "Replace w/ Keywords" experiment that removed all words from the tweet that were not related to science in some way.

The increase in the MRR@5 after reranking and retrieval for experiments where the original tweet was combined with the rewritten tweet suggests that multiple representations of the same tweet may improve performance.

Finally, it is worth noting that our data augmentation experiments were only applied to the query dataset. The data for the CORD-19 paper collection as not manipulated in any way. For some experiments, this probably affected results. For example, the "Replace w/ Keywords" likely would have performed better if the prompt "Return a list of only science-related keywords in the tweet" was applied to both the query and CORD-19 paper collection.

6.2. Reranking Models

As noted in the results section, 3 the models trained on the MS Marco dataset performed significantly better than those trained on the GooAQ dataset. The MS Marco dataset was generated from Bing queries with human-generated answers. The GooAQ dataset was curated using queries from Google with answers drawn from Google's responses to questions. Both datasets are generated from search queries, but the way the answers were generated differs greatly. One possible explanation is that the human-generated answers to the MS Marco dataset more closely reflect the language used in the abstracts for the CORD-19 paper collection. Another explanation might stem from differences in how users ask questions on Bing and Google. Perhaps users on Bing tend to be more verbose or search for different criteria than Google users.

6.3. Bi-encoder Finetuning

finetuning the bi-encoder decreased retrieval performance on all experiments. However, the "Gold dataset + hard negatives from silver data" was very close to the baseline, suggesting that there is the potential for improvement with further exploration.

7. Future Work

Task 4b leaders SourceSniffers achieved an MRR@5 of 0.68. The problem of finding relevant scientific claims for social media discourse is still unsolved. Future work should prioritize improving upon the work CheckThat! Task 4b participants explored this year.

7.1. Data Augmentation

As noted earlier, data augmentation techniques were only applied to query tweets. To see if data augmentation yields helpful results, these techniques should be applied to both the query dataset and the CORD-19 paper collection.

For the CORD-19 paper collection, it would be interesting to see if summarizing abstracts affects model performance. The abstracts in the CORD-19 paper collection were anywhere from 2 words to 1800 words. If social media users are more likely to reference mains ideas from the papers in their tweets, summarizing abstracts might lead to improved retrieval results.

In our experiments, only the title and abstract from the CORD-19 paper collection were used to when looking for relevant articles. It would be interesting to see if including additional features like the journal that the article appeared in and the names of authors would improve performance.

7.2. Bi-encoder Finetuning

The bi-encoder performed worse after finetuning, suggesting that the silver dataset may introduce noise that hinders the model's ability to retrieve relevant documents. Future work should focus on filtering the silver data before finetuning. We could explore a pre-trained cross-encoder to score the augmented tweet–document pairs and retain only high-confidence examples. The results highlight that more data may not lead to better performance, especially when additional examples are not meaningfully contributed to the learning signal.

7.3. Other Avenues to Explore

If retrieval results were poor, reranking could only do so much. Future work should prioritize techniques that improve retrieval. One potential avenue for this is to combine multiple approaches: augmenting data, finetuning a bi-encoder for retrieval, and finding the best model for reranking.

8. Conclusion

This paper demonstrates the work performed by DS@GT for CheckThat! 2025 Task 4b, Scientific Claim Source Retrieval. We explored data augmentation techniques, reranking and retrieval pipelines, and finetuning bi-encoders. We found that combining original tweets with rewritten tweets that reflect the language of the document collection may lead to an improvement in retrieval. Using BM25-Pytorch for retrieval and T5 for reranking, we achieved an MRR@5 of 0.58 on the evaluation set, and improvement of 0.15 over the baseline of 0.43. Future work should prioritize improving the retrieval stage of the retrieval and reranking pipeline. Our code is available on Github at https://github.com/dsgt-arc/checkthat-2025-swd/tree/main/subtask-4b

Acknowledgments

Thank you to the everyone in the DS@GT CLEF team for their support. Special thanks to Anthony Miyaguchi and Murilo Gustenelli for their many hours of work organizing and supporting the DS@GT CLEF team. This paper would not have happened without you!

Thank you to Partnership for an Advanced Computing Environment (PACE) [26] at the Georgia Institute of Technology, Atlanta, Georgia, USA for allowing us to use their resources to perform this research.

Declaration on Generative AI

Generative AI was used in the preparation of this work, specifically to format bibliography references and for formatting tables and figures. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] D. Kbaier, A. Kane, M. McJury, I. Kenny, Prevalence of health misinformation on social media—challenges and mitigation before, during, and beyond the covid-19 pandemic: Scoping literature review, Journal of Medical Internet Research 26 (2024) e38786. URL: https://www.jmir.org/2024/1/e38786. doi:10.2196/38786.

[2] S. Hafid, W. Ammar, S. Bringay, K. Todorov, Cite-worthiness detection on social media: A preliminary study, Natural Scientific Language Processing and Research Knowledge Graphs 19–30 (2024).

[3] 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: [4], 2025.
[4] G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 2025.
[5] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, R. Míguez, T. Caselli, M. Kutlu, W. Zaghouani, C. Li, S. Shaar, H. Mubarak, A. Nikolov, Y. S. Kartal, Overview of the CLEF-2022 CheckThat! lab task 1 on identifying relevant claims in tweets, in: [6], 2022.
[6] G. Faggioli, N. Ferro, A. Hanburg, M. Potthast (Eds.), Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, CLEF 2022, Bologna, Italy, 2022.
[7] G. Faggioli, N. Ferro, P. Galuščáková, A. G. Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CLEF 2024, Grenoble, France, 2024.
[8] D. Wright, I. Augenstein, Citeworth: Cite-worthiness detection for improved scientific document understanding (2021).
[9] S. Hafid, S. Schellhammer, S. Bringay, K. Todorov, S. Dietze, Scitweets - a dataset and annotation framework for detecting scientific online discourse, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 3988–3992. URL: https://doi.org/10.1145/3511808.3557693. doi:10.1145/3511808.3557693.
[10] Wikipedia contributors, Okapi bm25 – Wikipedia, the free encyclopedia, 2025. URL: https://en.wikipedia.org/wiki/Okapi_BM25, accessed: 2025-07-07.
[11] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, Ms marco: A human generated machine reading comprehension dataset, CoRR abs/1611.09268 (2016). URL: http://arxiv.org/abs/1611.09268.
[12] C. T. Hugging Face, cross-encoder/ms-marco-MiniLM-L6-v2, https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2, 2025. Model card; accessed: 2025-07-07.
[13] T. Arsen, the Hugging Face Community, reranker-msmarco-modernbert-base-lambdaloss, Hugging Face model hub, 2025. URL: https://huggingface.co/tomaarsen/reranker-msmarco-ModernBERT-base-lambdaloss.
[14] T. Arsen, the Hugging Face Community, reranker-msmarco-minilm-l12-h384-uncased-lambdaloss, Hugging Face model hub, 2025. URL: https://huggingface.co/tomaarsen/reranker-msmarco-MiniLM-L12-H384-uncased-lambdaloss.
[15] D. Khashabi, A. Ng, T. Khot, A. Sabharwal, H. Hajishirzi, C. Callison-Burch, Gooaq: Open question answering with diverse answer types, CoRR abs/2104.08727 (2021). URL: https://arxiv.org/abs/2104.08727. arXiv:2104.08727.
[16] akr2002, the Hugging Face Community, reranker-modernbert-base-gooaq-bce, Hugging Face model hub, 2025. URL: https://huggingface.co/akr2002/reranker-ModernBERT-base-gooaq-bce.
[17] T. Arsen, the Hugging Face Community, reranker-modernbert-large-gooaq-bce, Hugging Face model hub, 2025. URL: https://huggingface.co/tomaarsen/reranker-ModernBERT-large-gooaq-bce.
[18] T. Arsen, the Hugging Face Community, reranker-neobert-gooaq-bce, Hugging Face model hub, 2025. URL: https://huggingface.co/tomaarsen/reranker-NeoBERT-gooaq-bce.
[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. URL: http://jmlr.org/papers/v21/20-074.html.
[20] AIAAIC Repository, C4 (colossal clean crawled corpus) dataset, AI, Algorithmic & Automation Incidents and Issues (AIAAIC), 2023. URL: https://www.aiaaic.org/aiaaic-repository/ai-algorithmic-and-automation-incidents/c4-dataset, last updated Oct 2024.
[21] B. Clavié, the AnswerDotAI team, rerankers: A lightweight python library to unify ranking methods, GitHub repository, 2024. URL: https://github.com/AnswerDotAI/rerankers, version 0.6.0.
[22] B. Clavié, rerankers: A lightweight python library to unify ranking methods, arXiv preprint abs/2408.17344 (2024). URL: https://arxiv.org/abs/2408.17344, arXiv:2408.17344.
[23] Sentence-Transformers Team, msmarco-distilbert-base-v4, Hugging Face model hub, 2025. URL: https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v4.

[24] 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: http://arxiv.org/abs/1908.10084.

[25] N. Thakur, N. Reimers, J. Daxenberger, I. Gurevych, Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks, arXiv preprint arXiv:2010.08240 (2020). URL: https://arxiv.org/abs/2010.08240.

[26] PACE, Partnership for an Advanced Computing Environment (PACE), 2017. URL: http://www.pace.gatech.edu.