Team AKSH at CheckThat! 2026: Resource-Efficient Hybrid Retrieval for Multilingual Scientific Source Retrieval[,★★]

Sufyan Siddiqui[1,*,†], Faisal Alvi[1,*] and Abdul Samad[1]

[1]Department of Computer Science, Dhanani School of Science and Engineering, Habib University, Karachi, Pakistan

Abstract

CheckThat! 2026 Task 1 requires systems to identify the five most probable source papers for a scientific claim embedded in a tweet, selecting from a fixed 10,000-document collection. We present **Team AKSH** from Habib University, a system that fine-tunes multilingual E5 on pooled English, French, and German tweet-paper pairs, then fuses the resulting dense scores with BM25. A single global weight α=0.81 is selected entirely on held-out training data; the development set is left untouched throughout. On the official development set, macro MRR@5 reaches 0.5605 (EN 0.6184, FR 0.5974, DE 0.4658). We also evaluated cross-encoder reranking and multi-signal fusion pipelines; both degraded performance relative to the hybrid baseline, so we excluded them from the final submission.

Keywords

CheckThat!, source retrieval, hybrid retrieval, BM25, dense retrieval, multilingual retrieval

1. Introduction

The CLEF 2026 CheckThat! Lab [1, 2] evaluates fact-checking technologies in multilingual contexts. Within this lab, Task 1 [3, 4] concerns source retrieval for scientific web claims. Each instance is a tweet implying a scientific finding. The system must rank a shared pool of 10,000 papers and return an ordered list of the five most probable sources. Tweets and training labels are provided for English (EN), French (FR), and German (DE); the same paper collection serves all three languages.

Beyond finding a strong pipeline, this work asks a broader research question: to what extent can a simple, single-stage hybrid retriever perform well against more complex reranking models when dealing with the large difference between informal social media text and dense scientific abstracts?

The primary evaluation metric is MRR@5. If the correct paper appears at rank r between 1 and 5, it contributes $1/r$ to the score; otherwise, it contributes zero. The organisers also report accuracy at rank 1 (Acc@1) and whether the gold paper falls within the top five (Recall@5).

This paper focuses exclusively on Task 1. It outlines the experimental setup, the full training and retrieval pipeline in exact detail, the development-set results, and the experiments that failed to yield improvements.

2. Task and Objectives

2.1. What the task requires

The task overview [3] fully defines Task 1. Related work on scientific web text at CheckThat! is in the 2025 Task 4 overview [5].

For every development or test tweet:

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

[★★] Public dataset: `sschellhammer/CT26_Task1_SourceRetrievalForScientificWebClaims` on Hugging Face (splits `en`, `fr`, `de`, and `collection`).

[*]Corresponding authors.

[†]Contributing Author

ms09020@st.habib.edu.pk,sufyan4854@gmail.com (S. Siddiqui); faisal.alvi@sse.habib.edu.pk (F. Alvi); abdul.samad@sse.habib.edu.pk (A. Samad)

1. Consider all 10,000 papers in the collection (each paper has a numeric pubkey ID, title, abstract, venue, and authors).
2. Produce an ordered list of exactly five pubkey values.

Training data provides one gold pubkey per tweet. A small number of pubkeys are shared across multiple tweets. At inference there is no candidate shortlist: every paper is scored.

2.2. Metrics

Let L_i be the gold pubkey for query i and let $R_{i,1}, \ldots, R_{i,5}$ be the predicted list.

$$\text{MRR@5} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{5} \mathbf{1}[R_{i,k} = L_i] \cdot \frac{1}{k} \tag{1}$$

$$\text{Acc@1} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[R_{i,1} = L_i] \tag{2}$$

$$\text{Recall@5} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[L_i \in \{R_{i,1}, \ldots, R_{i,5}\}] \tag{3}$$

Macro MRR@5 reported in the tables is the unweighted mean of per-language MRR@5 across EN, FR, and DE.

All numbers in Section 5 use the official labelled dev splits. Dev labels were not used for training or for choosing α.

2.3. Goals

Three rigid constraints shaped the system architecture. First, the entire pipeline had to run on free GPU notebook infrastructure, entirely avoiding paid inference APIs at query time. Second, the hybrid approach needed to measurably outperform dense-only retrieval on the development set. Third, supplementary components were incorporated only if they demonstrated clear empirical benefits. Prior CheckThat! work has already demonstrated the value of combining lexical and dense signals (such as SimBa [6] for claim retrieval in 2022, ATOM [7] for scientific evidence retrieval in 2025). Reranking has also been explored: SeRRa [8] and SCIRE [9] report 2025 results with cross-encoder stages. Building on this precedent, four architectural extensions were tested: (1) cross- encoder reranking over dense top-K candidates, (2) three-model reciprocal rank fusion with BGE-M3, (3) BPE tokenisation for BM25 with per-language α, and (4) BGE-M3 fine-tuning on the task pairs. Of these, the first three were evaluated on the dev set and none consistently improved on the base hybrid across all three languages; the fourth, BGE-M3 fine-tuning, did not complete within our GPU budget. Only E5 + BM25 with a global α=0.81 was submitted.

Positioning relative to the 2025 state of the art. At the equivalent CheckThat! 2025 scientific claim source retrieval task (Task 4b), the best participating system reached a test-set MRR@5 of 0.67, and most strong systems combined dense retrieval with a cross-encoder reranking stage [5]. Our submitted system uses a fine-tuned multilingual E5 encoder but omits the reranking stage, running entirely within free-tier Kaggle GPU notebooks (NVIDIA T4, 15.7 GB VRAM). On the official test set, our system reaches a macro MRR@5 of 0.5199 (Table 6); the gap of roughly 0.15 points relative to the strongest reported 2025 system reflects primarily the absence of a fine-tuned reranker rather than a fundamental architectural divergence. Our own experiments with a zero-shot cross-encoder reranker (Section 4.6) confirm that a reranker without task-specific fine-tuning degrades results on this domain, consistent with the vocabulary gap between informal tweet text and formal scientific abstracts.

3. Data and Resources

3.1. Dataset and splits

We load data from the Hugging Face dataset `sschellhammer/CT26_Task1_SourceRetrievalForScientificWebClaims`. Per-language configs `en`, `fr`, and `de` each contain `train` and `dev` splits; the `collection` config lists all 10,000 papers. Table 1 shows tweet counts per language.

Table 1
Task 1 tweet counts per language. The paper collection has 10,000 entries shared across all languages.

Language	Train tweets	Dev tweets
English (EN)	14,977	3,905
French (FR)	2,807	702
German (DE)	1,460	386

We joined the mapping from tweet ID to `pubkey` with the collection JSON to construct complete tweet-paper text pairs. After filtering out invalid pairs, exactly 19,244 training instances remained. We discarded pairs in which the gold `pubkey` did not appear in the 10,000-paper collection, or in which either the tweet text or the paper abstract was absent or empty. The retained count matches the size of the raw training data because no such invalid pairs were found in practice; the filtering step confirmed data integrity rather than removing any examples. German contributes roughly one-tenth of the English pair count. That disparity carries through to development-set performance, where German scores remain consistently lower.

3.2. Text given to each retriever

Dense encoder (E5). Each paper is represented as:

`passage: {title}. {abstract}. Venue: {venue}`

Each tweet query as:

`query:` t

These prefixes follow the `multilingual-e5` training convention.

BM25. We use the same title, abstract, and venue text, but *without* the `passage:`/`query:` prefixes. For the tweet side, we use raw lowercase whitespace tokenisation (no stemmer, no stopword list).

3.3. Software and hardware

We used Python 3, PyTorch, Hugging Face `datasets`, `sentence-transformers` (training and dense encoding), and `rank_bm25` (BM25Okapi). We ran training and encoding on Kaggle with NVIDIA T4 or P100 GPUs.

The required packages are `sentence-transformers`, `rank-bm25`, `datasets`, `transformers`, and `accelerate`. We enabled mixed-precision training (`use_amp`) when CUDA was available.

4. Methodology

4.1. Overview

The submitted system has four steps: (1) fine-tune a bi-encoder on tweet-passage pairs, (2) embed all 10,000 collection papers once offline, (3) at query time, score every paper with dense cosine similarity and BM25 at the same time, and (4) apply per-query min-max normalisation, mix the scores with weight α, and return the top five paper IDs. Figure 1 diagrams the sequence. No query expansion or reranking step is included in the final submission.

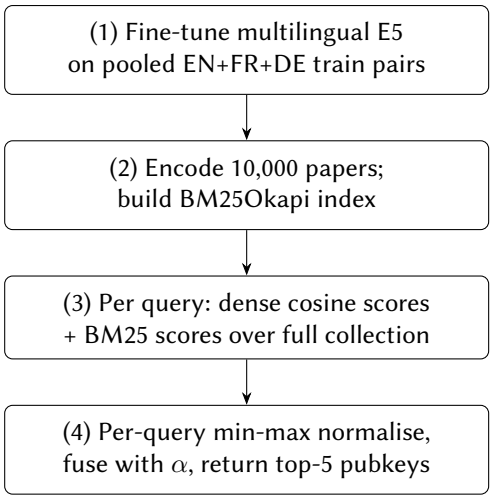

Figure 1: Submitted Task 1 pipeline. Steps (1)-(2) run once offline; step (3)-(4) run per query at inference time.

4.2. Step 1: Fine-tuning the dense retriever

Base model. We used `intfloat/multilingual-e5-large` as the base bi-encoder. We kept the smaller `intfloat/multilingual-e5-base` available as a fallback only if the large model caused out-of-memory errors during training; we used the large model throughout the submitted run.

Training pairs. One positive pair per training tweet: the `query:` tweet text paired with the `passage:` string of its gold paper. We pooled all 19,244 pairs from EN, FR, and DE train splits into a single list.

Train/holdout split. After shuffling, we used the first 80% of pairs (15,395) to train the model, and we kept the last 20% (3,849) only to tune α. Dev tweets were never in this holdout.

Loss and optimisation. We trained the model using Multiple Negatives Ranking Loss (MNRL) with in-batch negatives. We used the following training hyperparameters:

- Epochs: 3
- Batch size: 6 (submitted run; fallback policy: retry with 4 then 2 if out-of-memory on the same model)
- Maximum sequence length: 192 tokens
- Warmup steps: 10% of total training steps
- Optimiser and scheduler: `sentence-transformers` default training settings (no manual override in the code)

Note on batch size and in-batch negatives. Under MultipleNegativesRankingLoss, every other sample in the batch acts as a negative example. At batch size 6, each positive therefore sees only five in-batch negatives. Kaggle T4 memory ceilings imposed this constraint rather than our design choice. Larger batches generally provide harder and more informative negatives. They would likely improve retrieval quality, yet they remained strictly out of reach on the available hardware.

After training, we loaded the new model weights for all later encoding and retrieval steps using the same max length of 192 tokens.

4.3. Step 2: Indexing the collection

We encoded all 10,000 papers with the fine-tuned model in batches of 16, producing normalised embeddings. Dense similarity between a query and a paper is the dot product of their normalised embeddings (equivalent to cosine similarity).

We built the BM25 index once on tokenised collection passages and reused it for every query.

4.4. Step 3: Scoring and fusion

For each tweet:

1. Compute a length-10,000 dense score vector and a length-10,000 BM25 score vector.
2. Apply per-query min-max scaling to $[0, 1]$ on each vector separately: subtract the per-query minimum over all papers, divide by the range, and add 10^{-9} to the denominator to prevent division by zero.
3. Combine with a single global weight α:

$$s_{\text{hybrid}} = \alpha \cdot \hat{s}_{\text{dense}} + (1 - \alpha) \cdot \hat{s}_{\text{BM25}} \tag{4}$$

4. Sort papers by s_{hybrid} descending and return the top-five `pubkey` values.

We chose per-query min-max normalisation because of its simplicity and its natural ability to handle different score ranges across queries. We did not test rank-based alternatives like reciprocal rank scaling; this remains a task for future work.

Tuning α. We used only the 3,849 holdout pairs for weight selection. A basic search covered $\alpha \in \{0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$; the best coarse value was then refined over ± 0.04 in steps of 0.01. We applied the selected value, $\alpha = 0.81$, equally across all three languages at dev time. We did not tune for each language separately.

This mix of dense and BM25 scores matches methods set by SimBa [6] in 2022 and extended at a larger scale by ATOM [7] in 2025.

4.5. Step 4: Development evaluation

For each language, we loaded the official `dev` split and scored all 10,000 papers with the same collection index and the fixed $\alpha = 0.81$. We computed MRR@5, Acc@1, and Recall@5 against dev labels. Macro MRR@5 is the unweighted mean across the three languages.

The submission to the organisers consisted of a JSON list per language, with one entry per dev tweet, each entry containing a ranked list of five integer `pubkey` values.

4.6. Experiments not submitted

Cross-encoder reranking. We used the reranker `cross-encoder/mmarco-mMiniLMv2-L12-H384-v1`. For each tweet, we formed the top $K \in \{50, 100, 150\}$ papers from the *dense* channel (rather than the hybrid ranking) into (tweet, passage) pairs, scored them in batches of 24, and returned the top five. Table 2 shows MRR@5 for all three development languages. On every language and every K tested, scores dropped well below the hybrid baseline. We therefore kept the cross-encoder out of the submitted run.

Table 2
Cross-encoder reranking on the dev set. Candidates are drawn from dense top-K only. The submitted hybrid baseline (α=0.81) is included solely for reference. EN K=150 was not completed due to runtime constraints.

Setup	MRR@5 EN	MRR@5 FR	MRR@5 DE
Hybrid (α=0.81, submitted)	**0.6184**	**0.5974**	**0.4658**
CE rerank, K=50	0.3853	0.4124	0.3112
CE rerank, K=100	0.3632	0.3877	0.2923
CE rerank, K=150	–	0.3726	0.2725

Why the cross-encoder degraded performance. Tweets are short and informal; scientific abstracts are long, technical, and dense with domain terminology. That mismatch likely forces the model to rank candidates by general web relevance instead of scientific source match, inverting many correct dense rankings in the process. Two important variations went untested: a reranker fine-tuned specifically on

tweet-abstract pairs, and the application of reranking to hybrid rather than dense-only candidates. The finding that this specific configuration drops performance should not, therefore, be taken as conclusive proof that reranking is entirely unhelpful on this task.

BPE BM25 with per-language α. Replacing whitespace tokenisation with BPE for the BM25 index and tuning a separate α per language on the holdout lifted French MRR@5 from 0.5974 to 0.6105. English, however, fell slightly, decreasing from 0.6184 to 0.6155 (compared against the submitted hybrid in Table 5; the daggered row in Table 7 is a lower-scoring re-run and is not the comparison baseline). The simpler global whitespace BM25 worked well overall and avoided the hard work of per-language tuning, so it was kept for the submission.

Three-way reciprocal rank fusion. We added zero-shot `BAAI/bge-m3` (including both dense and sparse channels) to the fusion pool via reciprocal rank fusion (RRF). Resulting MRR@5 values were 0.576 on EN, 0.578 on FR, and 0.433 on DE, which are all below the fine-tuned E5+BM25 hybrid (0.6184, 0.5974, 0.4658). This result is unsurprising because BGE-M3 was added to the fusion without task-specific fine-tuning, while E5 had been fine-tuned on 15,395 tweet-paper pairs. However, the zero-shot BGE-M3 RRF result provides a useful reference point for untuned multilingual retrieval on this task.

Hard-negative mining, tweet paraphrase, and BGE-M3 fine-tuning. We explored mining hard negatives (papers scoring high under BM25 but low under the dense model, used as training negatives), paraphrasing tweet queries via a multilingual T5 model, and fine-tuning BGE-M3 on the task pairs. None of these produced a full set of dev predictions within the available GPU limits.

5. Results

5.1. Fusion weight selection on holdout

Table 3 reports the coarse α sweep over the 3,849-pair multilingual holdout. MRR@5 rises steadily from α=0.55 to a flat area centred near 0.80–0.85, after which it drops. The difference between the two best basic steps, namely α=0.80 (MRR@5 0.6226) and α=0.85 (0.6225), is very small. Fine-grained search between those values produced $\alpha = 0.81$ as the selected weight for all subsequent evaluations.

Table 3
Fusion weight α sweep on holdout training pairs (n=3,849, all languages mixed). Dense weight is α; BM25 weight is $(1-\alpha)$.

α	MRR@5	Acc@1	Recall@5
0.55	0.5154	0.4599	0.6051
0.60	0.5479	0.4853	0.6498
0.65	0.5772	0.5084	0.6846
0.70	0.6023	0.5360	0.7062
0.75	0.6126	0.5461	0.7158
0.80	0.6226	0.5549	0.7267
0.85	0.6225	0.5544	0.7246
0.90	0.6208	0.5521	0.7259
0.95	0.6119	0.5409	0.7233

5.2. French development set

Table 4 compares three configurations on the 702 French development tweets. The hybrid improves over dense-only on all three metrics. Cross-encoder reranking, however, drops performance at every candidate pool size; the drop grows as K increases, which makes sense as the reranker processes weaker dense candidates.

Table 4
French development set (702 queries). Rerank candidates come from dense top-K.

Setup	MRR@5	Acc@1	Recall@5
Dense only	0.5835	0.5043	0.7123
Hybrid (α=0.81)	**0.5974**	**0.5214**	**0.7208**
CE rerank, K=50	0.4124	0.3405	0.5313
CE rerank, K=100	0.3877	0.3234	0.4943
CE rerank, K=150	0.3726	0.3105	0.4772

5.3. Submitted system on all development languages

Table 5 reports the submitted system across all three languages.

Table 5
Submitted hybrid system on the official dev set (α=0.81, fine-tuned multilingual E5 + BM25).

Language	MRR@5	Acc@1	Recall@5
English	**0.6184**	0.5539	0.7163
French	0.5974	0.5214	0.7208
German	0.4658	0.3860	0.5907
Macro average	**0.5605**	–	–

5.4. Official test set results

Table 6 reports the official test set results for our submitted system (Team AKSH), as evaluated by the CheckThat! organisers.

Table 6
Official test set MRR@5 scores for the submitted system.

Language	Avg.	EN	FR	DE
MRR@5	**0.5199**	0.5724	0.5345	0.4529

5.5. Alternative configurations

Table 7 places the submitted system alongside three alternative configurations. The German column for the BPE-BM25 per-language variant was not completed within the available Kaggle GPU session time; the *n/a* entry is footnoted accordingly. The submitted hybrid German score in Table 5 therefore serves as the only available DE reference for that row.

Table 7
Alternative configurations on dev. RRF DE comes from the runs described in Section 4.6. The submitted system is shown at the bottom for direct comparison. † To recover dense-only scores not separately logged at submission, the daggered rows in this table were produced by a post-submission re-run. They therefore differ slightly from the submitted system (Table 5) and, for French dense-only, from the submission-time figure in Table 4, owing to run-to-run session variance. ‡ German BPE BM25 evaluation was not completed within the available GPU session time at submission; the standard hybrid result (Table 5) uses global α with whitespace BM25.

Setup	MRR@5 EN	MRR@5 FR	MRR@5 DE
Non-submitted experiments			
Fine-tuned E5, dense only†	0.5613	0.5899	**0.4680**
Fine-tuned E5 + BPE BM25, per-lang. α	0.6155	0.6105	*n/a*‡
BGE-M3 + E5 + BM25, 3-way RRF	0.5760	0.5780	0.4330
Post-submission re-eval baseline			
Hybrid E5+BM25, α=0.81†	**0.6150**	**0.5905**	0.4659

6. Discussion

All metric differences reported here are from a single run; no confidence intervals or significance tests were done.

Language gap. English achieves the highest dev MRR@5 (0.6184); German the lowest (0.4658).

German contributes roughly one-tenth as many training pairs as English (Table 1), and a single global α cannot compensate for that difference.

BM25 contribution across languages. On German the hybrid and dense-only scores are effectively equivalent, with the hybrid lower by a negligible 0.0021 in the post-submission re-evaluation run (see footnote †, Table 7); we therefore treat the German result as neutral rather than as evidence for or against the hybrid. On English and French the hybrid clearly improves over dense-only (using the submitted evaluation for French, Table 4), and the macro average likewise favours the hybrid, indicating that BM25 contributes complementary lexical signal overall even when the dense encoder has been fine-tuned on task data. The α flat area in Table 3 is consistent with this, suggesting that BM25 contributes positively when averaged across all three languages.

Cross-encoder failure. Across all three languages and all candidate pool sizes, Table 2 shows large MRR@5 drops relative to the hybrid baseline. The most probable cause is domain mismatch: `mmarco-mMiniLMv2-L12-H384-v1` was trained on MS MARCO web-document pairs, not on tweet-abstract relevance. Short, informal tweets and long technical abstracts make a much harder retrieval pair than the web text the model was built for. We believe that the cross-encoder may frequently rank highly technical papers with general scientific terms above the true sources, struggling to match the informal phrasing of the tweet to the dense academic text of the abstract without task-specific fine-tuning. The experiment tested a single reranker applied to dense-only candidates; conclusions about reranking in general are too early given this limited setup.

7. Limitations

Dev-tuned pipeline. All hyperparameter choices (including α) were made on held-out training data and validated on the official development set. Official test-set scores (Table 6) were released by the organisers after the submission deadline and are included here without any post-hoc tuning.

English-dominated holdout. The 20% holdout used for α selection matches the training distribution: approximately 78% English, 14% French, and 8% German pairs. The chosen α=0.81 is therefore tuned mostly on English data. Per-language weights might better serve French and German; tuning those weights on dev labels was explicitly avoided to prevent data leakage.

Small batch size for MNRL. At batch size 6, each training example has five in-batch negatives under MultipleNegativesRankingLoss, which is fewer than normal for strong dense retrieval training.

The Kaggle T4 memory ceiling imposed this constraint. Gradient accumulation was not tried as an alternative.

Cross-encoder experiment scope. Reranking candidates were drawn from the dense-only top-K. Whether the ordering of hybrid candidates changes the reranker results remains an open question.

Hardware variability. The specific GPU assigned by Kaggle varied across sessions; occasional CUDA cache clearing was needed at batch size 6 on T4 hardware, causing minor run-to-run differences during training.

8. Conclusion and Future Work

We submitted a hybrid retrieval system to CheckThat! 2026 Task 1. The system fine-tunes multilingual E5-large on 15,395 pooled tweet-paper pairs, combines the resulting dense scores with BM25Okapi via linear fusion, and selects a single global weight α=0.81 from a 3,849-pair held-out training slice. All retrieval runs are conducted over the full 10,000-paper collection on free Kaggle GPU notebooks, achieving a macro dev MRR@5 of 0.5605 (EN 0.6184, FR 0.5974, DE 0.4658).

Returning to the research question framing this work: these results show that a simple, resource-efficient bi-encoder and lexical pipeline can provide a strong baseline for multilingual scientific source retrieval. At the same time, the findings suggest that without task-specific fine-tuning on social media to academic text pairs, more complex cross-encoder models are highly vulnerable to severe performance drops. Future work will therefore focus on adapting such rerankers to bridge the vocabulary gap between informal claims and formal scientific literature.

Acknowledgments

We would like to express our gratitude to Habib University for the invaluable support, guidance, and the academic environment that made this research possible. We also thank the CheckThat! organisers for the task definition, data, and evaluation infrastructure.

Declaration on Generative AI

During the preparation of this work, the authors used **Claude** (Anthropic, https://www.anthropic.com) in order to: refine the experimental plan (including suggesting hyperparameter search strategies and ablation designs), and debug and review implementation code. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

No AI tool was used to write, generate, or substantially paraphrase any scientific content, experimental results, analysis, or conclusions in this paper. No AI tool is listed as an author or contributor.

References

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

[2] J. M. Struß, S. Schellhammer, S. Dietze, V. V., V. Setty, T. Chakraborty, P. Nakov, A. Anand, P. Chungkham, S. Hafid, D. Sahnan, K. Todorov, Overview of the CLEF-2026 CheckThat! Lab: Advancing Multilingual Fact-Checking, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer, 2026.

[3] 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: CLEF 2026 Working Notes, CLEF 2026, Jena, Germany, 2026.

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

[5] S. Hafid, J. M. Struß, S. Schellhammer, S. Dietze, et al., Overview of CheckThat! 2025 Task 4: Scientific Web Discourse, in: Working Notes of CLEF 2025, volume 4038, CEUR-WS, 2025, pp. 722–732.

[6] A. Hövelmeyer, K. Boland, S. Dietze, SimBa at CheckThat! 2022: Hybrid Similarity for Previously Fact-Checked Claims, in: Working Notes CLEF 2022, volume 3180, CEUR-WS, 2022, pp. 511–531.

[7] M. Staudinger, A. El-Ebshihy, W. Kusa, F. Piroi, A. Hanbury, ATOM at CheckThat! 2025: Retrieve the Implicit — Scientific Evidence Retrieval, in: Working Notes of CLEF 2025, volume 4038, CEUR-WS, 2025, pp. 1246–1255.

[8] G. A. Marchetti, G. Rocha, H. L. Cardoso, SeRRa at CheckThat! 2025: Sequential Re-Ranking for Source Retrieval, in: Working Notes CLEF 2025, volume 4038, CEUR-WS, 2025, pp. 1045–1055.

[9] P. M. Thapliyal, R. S. Chavan, S. Samridh, C. Zuo, R. Banerjee, SCIRE at CheckThat! 2025: Dense Retrieval and Cross-Encoder Re-Ranking, in: Working Notes CLEF 2025, volume 4038, CEUR-WS, 2025, pp. 1256–1264.