ipezoTU at SemEval-2025 Task 7:
Hybrid Ensemble Retrieval for Multilingual Fact-Checking

Iva Pezo
iva.pezo0801@gmail.com
and **Allan Hanbury** and **Moritz Staudinger**
TU Wien, Data Science Research Unit
firstname.lastname@tuwien.ac.at

Abstract

Fact-check retrieval plays a crucial role in combating misinformation by ensuring that claims are accurately matched with relevant fact-checks. In this work, we present a hybrid retrieval pipeline that integrates lexical and semantic retrieval models, leveraging their complementary strengths. We evaluate different retrieval and reranking strategies, demonstrating that hybrid ensembling consistently outperforms individual models, while reranking provides only marginal improvements.

1 Introduction

Social media has transformed the way information is shared by enabling instant, unfiltered access to news and various perspectives. Unlike traditional media, it allows anyone to publish content without verification, making it more difficult to distinguish between true and misleading claims (Hale et al., 2024). The traditional approach of manual fact-checking has proved its reliability in providing high-quality results, however, it lacks the scalability to address the volume and speed of online information spread. The amount of data is rapidly growing as the same misinformation is often spread across different platforms while slightly altered in format, detail, length, or even language. It is often the case that the users are unaware of the veracity of the claims, especially in non-English contexts where fact-checking resources may be limited or less accessible (Balalau et al., 2024; Kazemi et al., 2021b). This highlights the need to develop and automate fact-checking systems to help maintain the accuracy and reliability of information shared online (Panchendrarajan and Zubiaga, 2024).

SemEval-2025 Shared Task 7 (Peng et al., 2025) tackles the challenge of multilingual and crosslingual Previously Fact-Checked Claim Retrieval (PFCR) (Pikuliak et al., 2023), a critical task in combating misinformation across languages. The task is divided into two subtasks: monolingual and crosslingual retrieval. In the monolingual subtask, the search space is restricted to fact-checked claims in the same language as the query claim. In contrast, the crosslingual subtask allows retrieval across multiple languages, enabling corresponding fact-checks in any language to be retrieved for a given query. The monolingual subtask includes data for Arabic, English, French, German, Malay, Portuguese, Spanish, and Thai, with Polish and Turkish added to the test set.

This paper explores the effectiveness of hybrid retrieval architectures for monolingual and crosslingual PFCR. We focus on zero-shot retrieval (Shen et al., 2024; Thakur et al., 2021), avoiding finetuning to ensure general applicability across diverse topics, languages, and platforms. By leveraging pre-trained models, our approach maintains competitive performance with minimal resource demands, demonstrating their effectiveness in multilingual settings without task-specific adaptations.

In both subtasks, we achieved our best results with a retriever ensembler, ranking 8th out of 28 teams in the monolingual and 12th out of 29 teams in the crosslingual task with over 177 participants and 1400 submissions.

The remainder of this work is structured as follows: we introduce the task and give a fact-checking pipeline overview in Sec. 2. Sec. 3 describes the modules of our system, while Sec. 4 presents key experiments and evaluation results that guided our design choices. Lastly, Sec. 5 summarizes our findings and outlines directions for future work.

2 Background

A survey on monolingual, multilingual, and crosslingual research (Panchendrarajan and Zubiaga, 2024) outlines the key components of an automated fact-checking pipeline: claim detection, claim prioritization, retrieval of evidence, veracity prediction, and explanation generation (Nakov et al.,

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1159–1167
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

2021; Balalau et al., 2024). In addition to these core steps, there is an additional component responsible for retrieving previously fact-checked claims. This component identifies claims that have already been verified, linking them to existing fact-checks. Aligning similar claims across languages can improve fact-checking efficiency and combat misinformation more effectively. This task is commonly referred to as verified claim retrieval (Barrón-Cedeño et al., 2020) or claim matching (Kazemi et al., 2021a), though it is also known as PFCR (Pikuliak et al., 2023) or fact-checked claim detection (Shaar et al., 2020). In this work, we focus on this retrieval component.

One could argue that the limitation of this task is the assumption of the existence of a fact-checked article for a given claim. However, it is important to remember that PFCR is an additional component that aims to create a shortcut in the fact-checking pipeline in case the same, perhaps reformulated, claim reappears online after it has been verified, reducing redundancy and improving response time in tackling misinformation.

MultiClaim dataset (Peng et al., 2025) includes 205,751 fact-checks in 39 languages and 28,092 social media posts in 27 languages. The dataset contains 31,305 verified post-fact-check pairs, with 4,212 being crosslingual. More details on the used dataset are given in Appendix A.

3 System Overview

This section presents the system architecture shown in Figure 1, outlining the key components of the pipeline: (1) Data preprocessing module, (2) Retrieval-ensemble module, (3) Reranking-ensemble module, (4) Evaluation module. Given a collection of fact-checks, our system retrieves the top k relevant fact-checks for any given claim.

3.1 Data Preprocesing Module

The data preprocessing module prepares claims and fact-checks for downstream retrieval and reranking. For lexical models, preprocessing focuses on cleaning and normalizing the text, while for semantic models, the text is enriched with additional contextual descriptions.

3.2 Retrieval-Ensemble Module

The retrieval-ensemble module returns the top k relevant fact-checks for a given claim as an ensemble of lexical and semantic retrievers. Each retriever independently ranks fact-checks based on their similarity to the claim, selecting the most relevant ones from the pool of verified claims. We compare a range of pre-trained retrieval models and select those with the best average performance across languages. We avoid language-specific adaptations and adopt a zero-shot retrieval setup (Shen et al., 2024; Thakur et al., 2021), avoiding model fine-tuning. This approach ensures robustness and applicability across diverse topics, languages, and platforms while minimizing resource demands.

The ensembler balances the strengths of sparse lexical and dense semantic retrievers, ensuring that the lexical model provides high-precision results for explicit term matches while semantic models capture implicit relationships and conceptual similarities.

3.3 Reranking-Ensemble Module

The reranking-ensemble module refines the top candidates obtained from the previous module using a set of cross-encoder rerankers. Each reranker returns its top candidates, which are then aggregated by a final ensembler into the final top 10 results.

Cross-encoders jointly encode claim–fact-check pairs, enabling fine-grained relevance scoring by capturing context-sensitive semantic interactions between the claim and the fact-check. While computationally more intensive, their use is justified at this stage due to a smaller pool of candidates, allowing for higher ranking precision without compromising efficiency.

3.4 Evaluation Module

We use the Success@k (S@k) metric for evaluation, which measures the proportion of claims for which at least one relevant fact-check appears within the top-k retrieved results.

4 Experiments and Results

4.1 Preprocessing Module

Preprocessing for Lexical Models. We evaluated BM25 retrieval using S@10 scores on both original-language and English-translated text without preprocessing. The translated version outperformed the original (0.5569 vs. 0.5171), likely due to greater linguistic consistency with fact-check sources. To further improve performance, we developed a preprocessing pipeline including URL and HTML entity removal, stop-word and punctuation

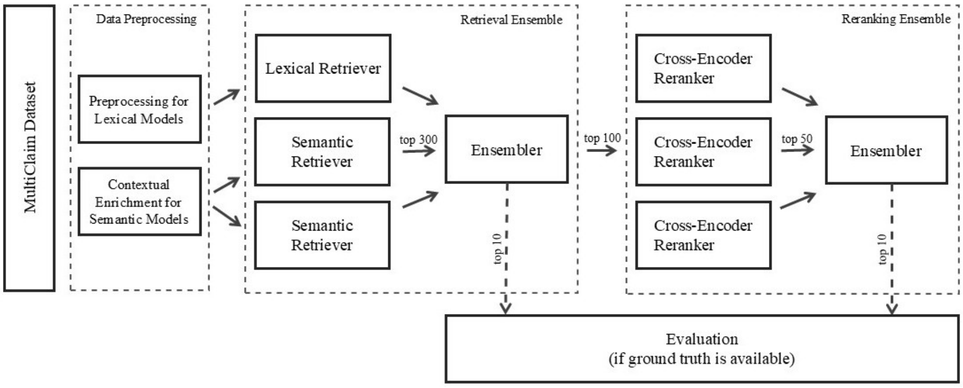

Figure 1: System architecture overview

filtering, Unicode and case normalization, character repetition reduction, whitespace standardization, emoji and date normalization, and lemmatization (Appendix C). Applying our full preprocessing pipeline led to a substantial improvement: S@10 increased to 0.7851 on original text and 0.7967 on translations.

Contextual Enrichment for Semantic Models. We optimized the input formatting, finding that explicitly defining components of fact-checks and posts significantly improved retrieval accuracy, enhancing the model's ability to understand contextual relationships between elements.

The best-performing format was:

> The following claim was posted: *OCR+text*, posted on *date*. The content is labeled as: *verdict*.
>
> This is a fact-checked claim: *claim*, with the title *title* posted on *date*.

Additionally, we evaluated simpler formats, such as concatenating components without descriptions and prefixing them with their names only, but both approaches led to lower retrieval accuracy, likely due to the lack of contextual guidance. We compare the retrieval performance of semantic models with the different input formats using S@100 in Table 1.

Model/Setting	No Descriptions	Prefixing	Contextual Guidance
E5	0.9440	0.9555	0.9586
BGE	0.9471	0.9531	0.9537

Table 1: Average S@100 scores for E5 and BGE models using different input formats and English-translated text

4.2 Retrieval-Ensemble Module

In the retrieval-ensemble module, the top 300 candidate fact-checks are retrieved using each retriever model and then aggregated into the top 100 candidates using an ensembler.

4.2.1 Retrievers

For evaluation of retrievers, we report S@100 scores ($k = 100$) rather than S@10, as the retrieval stage is responsible for producing a larger candidate set of fact-checks, which is later refined. Thus, performance on a broader set is more indicative of retrieval effectiveness at this stage.

Lexical models. BM25 is a widely used baseline in modern information retrieval (IR) research (Barrón-Cedeño et al., 2020; Nakov et al., 2022; Shaar et al., 2020; Aarab et al., 2024), making it a natural choice for our lexical retrieval component.

Model	Avg S@100	Model size (params)
Multilingual-E5-Large-Instruct	0.9330	560M
BGE-Multilingual-Gemma2	0.9293	9.24B
NV-Embed-v2	0.9201	7.85B
GTR-T5-Large	0.9019	1.24B
BGE-M3	0.8731	568M
MiniLM-L6-v2	0.7947	22.7M
stella_en_1.5B_v5	0.5288	1.54B
XLM-RoBERTa-Large	0.1467	561M

Table 2: Retriever model comparison on S@100 using original languages on the monolingual data

Semantic models. To identify the most effective retrievers in the zero-shot setting, we evaluated several pre-trained models using S@k scores on the training set. The results, shown in Table 2, informed our selection.

Among the evaluated models, `multilingual-E5-Large-Instruct`[1] (E5)

[1] `https://huggingface.co/intfloat/multilingual-e5-large-instruct`

was the top performer, achieving the highest average S@100 score of 0.9330. Despite its modest size (560M parameters), E5 outperformed the performance of much larger models, making it an efficient and effective choice. `BGE-Multilingual-Gemma2`[2] (BGE) followed closely with an average S@100 score of 0.9293. However, it comes at a significantly higher computational cost, with 9.24B parameters — over 16 times the size of E5. We included BGE as a complementary bi-encoder due to its strong performance. However, its latency was notably higher: E5 averaged 0.08 seconds per claim, while BGE required 0.39 seconds.

These findings highlight that larger models do not guarantee better retrieval. Instead, architecture design, training objectives, and multilingual optimization play a more critical role. For real-world deployments, especially in latency-sensitive or resource-constrained settings, mid-sized models like E5 provide an effective balance between retrieval performance and computational efficiency.

4.2.2 Ensembler

Aggregation Function. To combine outputs from multiple retrievers, we evaluated several aggregation strategies: majority voting, exponential decay weighting, and reciprocal rank fusion (RRF). Across both monolingual and crosslingual settings, RRF delivered the best retrieval performance.

In the monolingual setting, RRF achieved an S@100 score of 0.9720, outperforming exponential decay weighting (0.9674) and majority voting (0.9649). Similarly, in the crosslingual setting, RRF led with an S@100 of 0.8967, compared to 0.8897 for exponential decay weighting and 0.8813 for majority voting. Based on these results, we adopted RRF as the aggregation strategy in our final ensemble.

RRF (Cormack et al., 2009) assigns a score to each document d based on the reciprocal value of its rank between different retrievers:

$$R_{\text{score}}(d) = \sum_{r \in R} \frac{1}{k + \text{rank}_r(d)} \tag{1}$$

where R is the set of retrievers, $\text{rank}_r(d)$ is the rank of the document d assigned by the retriever r, and k is a constant added to prevent division by zero.

Retrieval Set Size. We evaluated the ensembler's S@k performance across retrieval set sizes (k = 50, 100, 200, 300, 400) in both monolingual and crosslingual settings to determine its optimal value. Performance improved as k increased, plateauing around k=300. In the monolingual setting, S@k increased from 0.9693 at k=50 to 0.9720 at k=300, with no further improvement beyond that. Similarly, in the crosslingual setting, scores increased from 0.8914 to 0.8970. We selected k=300 as an effective balance between retrieval quality and computational efficiency. The ensembler's robustness at higher k values can be attributed to the RRF aggregation method, which ensures that highly ranked fact-checks remain prioritized while lower-ranked ones have minimal impact.

Ensemble Weighting. We explored ensemble weighting strategies to optimize retrieval performance. Our findings show that assigning a lower weight of 0.5 to the lexical BM25 and a weight of 1.0 to the semantic E5 and BGE improves retrieval effectiveness. This reflects the stronger contribution of semantic retrieval in capturing the relationships between queries and fact-checks, whereas BM25, though effective for keyword matching, benefits more as a complementary component rather than a dominant factor. Experiment details are given in Appendix D.

4.2.3 Retrieval-Ensemble Module Performance

Table 3 compares the performance of retrievers and ensemble configurations. The results show that dense retrievers (E5, BGE) consistently outperform the lexical BM25 in all languages, demonstrating the effectiveness of semantic models. However, the ensemble methods that combine BM25 with dense retrievers show further performance gains, achieving the highest average S@100 scores of 0.9703.

Additionally, we assess the module's effectiveness as a standalone component instead of as an intermediate step in the pipeline using S@10. Table 5 presents a performance comparison between our retrieval-ensemble module and the best-performing baseline model (GTR-T5-Large) from the Multiclaim dataset paper (Pikuliak et al., 2023). Our retrieval-ensemble consistently outperforms the baseline across all languages, improving the average S@10 from 0.82 to 0.9237.

[2] https://huggingface.co/BAAI/bge-multilingual-gemma2

Model	k	ARA	DEU	ENG	FRA	MSA	POR	SPA	THA	AVG
E5	300	0.9691	0.9672	0.9580	0.9758	0.9844	0.9763	0.9719	1.0000	0.9752
BM25	300	0.9451	0.8946	0.8768	0.9345	0.9142	0.9289	0.9334	0.9886	0.9270
BGE	300	0.9657	0.9742	0.9481	0.9776	0.9649	0.9534	0.9627	1.0000	0.9683
E5 + BM25	100	0.9657	0.9344	0.9386	0.9677	0.9766	0.9575	0.9600	0.9924	0.9616
BGE + BM25	100	0.9537	0.9625	0.9358	0.9686	0.9571	0.9551	0.9558	0.9962	0.9606
E5 + BGE	100	0.9674	0.9672	0.9485	0.9722	0.9766	0.9583	0.9634	0.9962	0.9687
E5 + BM25 + BGE	100	0.9640	0.9720	0.9500	0.9713	0.9805	0.9633	0.9651	0.9962	0.9703

Table 3: Retrieval performance (S@k) using original-language text across models and ensembles on the training set

4.3 Reranking-Ensemble Module

Each reranker in the reranking ensemble module processes the top 100 fact-checks (obtained from the retrieval-ensemble module) and returns its top 50 candidates, which are then aggregated by a final ensembler into the top 10 results.

4.3.1 Rerankers

To select the rerankers for our pipeline, we prioritized models that demonstrated strong zero-shot performance while remaining computationally feasible. We used the MTEB[3] (Massive Text Embedding Benchmark) as a starting reference point and evaluated its leading reranker models.

QWEN (`gte-Qwen2-7B-instruct`[4]), NV (`NV-Embed-v2`[5]), and GRITLM (`GritLM-7B`[6]) were chosen for their strong reranking performance and compatibility with our resource constraints. Their average per-claim latencies were 0.40s (QWEN), 1.23s (GRITLM), and 1.28s (NV). In contrast, the retrievers, E5 (0.08s) and BGE (0.39s), are significantly faster, highlighting the importance of narrowing down the candidate set size during first-stage retrieval to keep reranking computationally feasible, especially in latency-critical or large-scale applications.

Evaluation Setups. We evaluated rerankers in three setups to analyze the impact of language representation and instruction translation: (1) using the original-language text with English task instructions, (2) using English-translated text and instructions, and (3) using the original language text with the task instructions translated into that language. While instruction translation in setup (3) improved performance in some cases, such as with Malay (MSA), where linguistic alignment aided retrieval, other languages, like Thai (THA), experienced performance drops. This suggests that instruction translation is not always beneficial and depends on both the complexity of the language and translation quality. The model performances under the three setups are compared in Table 4.

4.3.2 Reranking-Ensemble Module Performance

Table 4 presents the S@50 and S@10 scores of rerankers and ensembles across languages. GRITLM achieves the highest average S@50 score (0.9541), followed by NV (0.9512) and QWEN (0.9494). The original-language text paired with English task instructions (setup 1) consistently achieved the highest average scores across all three models and was therefore selected for use in the ensemble configurations. Performance on English-translated version (setup 2) shows mixed results across languages. Arabic (ARA) and Thai (THA) benefit the most, suggesting that translation into English can normalize morphologically rich languages, improving retrieval for models trained on English-heavy corpora. However, for others, as French (FRA) and Portuguese (POR), translation yields inconsistent results.

Our full ensemble strategy (QWEN + GRITLM + NV) achieves the highest average S@10 score (0.9202), outperforming pairwise ensembles and the baseline (0.9202 vs. 0.82).

4.4 Effectiveness of Reranking

Despite the expected advantages of reranking, the results in Table 5 indicate that the reranking-ensemble pipeline delivers only marginal improvements in Arabic, English, and French, failing to consistently outperform the retrieval-ensemble approach. Our results demonstrate that hybrid retrieval strategies — combining lexical and dense models — are both more effective and computa-

[3] `https://huggingface.co/spaces/mteb/leaderboard`

[4] `https://huggingface.co/Alibaba-NLP/gte-Qwen2-7B-instruct`

[5] `https://huggingface.co/nvidia/NV-Embed-v2`

[6] `https://huggingface.co/GritLM/GritLM-7B`

Model	k	ARA	DEU	ENG	FRA	MSA	POR	SPA	THA	AVG
NV (1)	50	0.9503	0.9438	0.9386	0.9596	0.9591	0.9379	0.9472	0.9734	0.9512
NV (2)	50	0.9588	0.9297	0.9386	0.9596	0.9493	0.9297	0.9444	0.9886	0.9498
NV (3)	50	0.9451	0.9438	0.9386	0.9614	0.9669	0.9395	0.9317	0.9544	0.9477
GRITLM (1)	50	0.9468	0.9485	0.9394	0.9650	0.9630	0.9379	0.9548	0.9772	0.9541
GRITLM (2)	50	0.9571	0.9204	0.9394	0.9659	0.9571	0.9453	0.9517	0.9886	0.9532
GRITLM (3)	50	0.9485	0.9321	0.9394	0.9650	0.9649	0.9404	0.9527	0.9316	0.9468
QWEN (1)	50	0.9451	0.9485	0.9235	0.9534	0.9630	0.9436	0.9448	0.9734	0.9494
QWEN (2)	50	0.9503	0.9110	0.9235	0.9453	0.9376	0.9093	0.9247	0.9810	0.9353
QWEN (3)	50	0.9430	0.9321	0.9235	0.9459	0.9643	0.9411	0.9421	0.9710	0.9454
Baseline Model (GTR-T5-Large)	10	0.86	0.69	0.77	0.86	0.82	0.80	0.84	0.90	0.82
QWEN + GRITLM	10	0.9177	0.8478	0.8792	0.9318	**0.9181**	0.9101	0.9196	0.9316	0.9070
QWEN + NV	10	0.9262	0.8501	0.8803	0.9309	0.9045	0.9003	0.9058	0.9316	0.9037
NV + GRITLM	10	0.9091	0.8618	**0.8942**	0.9363	0.9103	0.9028	0.9247	0.9087	0.9060
QWEN + GRITLM + NV	10	**0.9297**	**0.8735**	0.8938	**0.9444**	**0.9181**	**0.9142**	**0.9261**	**0.9620**	**0.9202**

Table 4: Reranking performance (S@k) on the training set. (1), (2) and (3) refer to the evaluation setups explained in 4.3.1. The best S@10 scores per language are in bold.

Model	ARA	DEU	ENG	FRA	MSA	POR	SPA	THA	AVG
Baseline Model (GTR-T5-Large)	0.86	0.69	0.77	0.86	0.82	0.80	0.84	0.90	0.82
Retrieval-Ensemble (E5 + BM25 + BGE)	0.9280	0.8923	0.8784	0.9408	0.9279	0.9158	0.9296	0.9772	**0.9237**
Reranking-Ensemble (QWEN + GRITLM + NV)	0.9297	0.8735	0.8938	0.9444	0.9181	0.9142	0.9261	0.9620	0.9202

Table 5: Comparison of retrieval-ensemble and reranking-ensemble approaches with the baseline using S@10

tionally efficient than stacking increasingly complex neural architectures. The limited improvements from reranking suggest that retrieval bottlenecks cannot always be resolved through additional processing, reinforcing the importance of well-designed ensembling over the reliance on increasingly complex models. This highlights that retrieval performance can be optimized efficiently without excessive computational overhead.

4.5 Test Set Performance

For the monolingual submission, we selected the best-performing setup per language, choosing between retrieval-ensemble and reranking-ensemble configurations. We used the retrieval-ensemble for Arabic, Malay, German, Thai and Turkish, and the full reranking-ensemble pipeline for English, French, Spanish, Portuguese and Polish. For crosslingual retrieval, we used the retrieval-ensemble setup. On the test set, our approach outperformed the organizer's baseline in both monolingual (S@10: 0.93 vs. 0.84) and crosslingual retrieval (S@10: 0.75 vs. 0.59). The top leaderboard model achieved 0.96 and 0.86, respectively.

4.6 Error Case Analysis

We analyzed retrieval errors in two scenarios: when individual retrievers failed but the ensembler succeeded, and when the ensembler was unsuccessful as well. In the first case, retrievers often identified relevant fact-checks but ranked them too low, favoring semantically related yet incorrect ones. The ensembler overcame this by integrating lexical and semantic cues, demonstrating the strengths of hybrid retrieval. In contrast, failure of ensembler typically involved vague or context-lacking claims, where implicit references made correct retrieval difficult. Overall, retrieval errors arise from limitations in ranking semantically relevant content and handling ambiguity. While ensembling improves robustness, performance remains sensitive to the clarity and specificity of claim formulation.

5 Conclusion

In this work, we show that reranking provides only marginal improvements over hybrid ensembling, while ensembling offers a balance between accuracy and efficiency. By combining strategic ensemble design and zero-shot retrieval, the retrieval-ensemble provides a scalable and effective solution for multilingual fact-checking. Its simplicity leaves room for further enhancements, such as k-shot retrieval or fine-tuning with fact-check data. Future work could explore improving retrieval robustness for ambiguous claims, alternative architectures, and adapting retrieval strategies based on language or claim complexity.

References

Abdelkrim Aarab, Ahmed Oussous, and Mohammed Saddoune. 2024. Optimizing arabic information retrieval: A comprehensive evaluation of preprocessing techniques. In *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 1–4.

Oana Balalau, Pablo Bertaud-Velten, Younes El Fraihi, Garima Gaur, Oana Goga, Samuel Guimaraes, Ioana Manolescu, and Brahim Saadi. 2024. FactCheckBureau: Build Your Own Fact-Check Analysis Pipeline. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, pages 5185–5189, New York, NY, USA. Association for Computing Machinery.

Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. 2020. Overview of checkthat! 2020: Automatic identification and verification of claims in social media. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings*, page 215–236, Berlin, Heidelberg. Springer-Verlag.

Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. 2020. Overview of CheckThat! 2020: Automatic Identification and Verification of Claims in Social Media. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 215–236, Cham. Springer International Publishing.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Scott A. Hale, Adriano Belisario, Ahmed Mostafa, and Chico Camargo. 2024. Analyzing Misinformation Claims During the 2022 Brazilian General Election on WhatsApp, Twitter, and Kwai. ArXiv:2401.02395.

Ashkan Kazemi, Kiran Garimella, Devin Gaffney, and Scott Hale. 2021a. Claim Matching Beyond English to Scale Global Fact-Checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4504–4517, Online. Association for Computational Linguistics.

Ashkan Kazemi, Zehua Li, Verónica Pérez-Rosas, and Rada Mihalcea. 2021b. Extractive and Abstractive Explanations for Fact-Checking and Evaluation of News. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 45–50, Online. Association for Computational Linguistics.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated Fact-Checking for Assisting Human Fact-Checkers. ArXiv:2103.07769.

Preslav Nakov, Giovanni Da San Martino, Firoj Alam, Shaden Shaar, Hamdy Mubarak, and Nikolay Babulkov. 2022. Overview of the clef-2022 checkthat! lab task 2 on detecting previously fact-checked claims. In *Conference and Labs of the Evaluation Forum*.

Rrubaa Panchendrarajan and Arkaitz Zubiaga. 2024. Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research. *Natural Language Processing Journal*, 7:100066.

Qiwei Peng, Robert Moro, Michal Gregor, Ivan Srba, Simon Ostermann, Marian Simko, Juraj Podroužek, Matúš Mesarčík, Jaroslav Kopčan, and Anders Søgaard. 2025. Semeval-2025 task 7: Multilingual and crosslingual fact-checked claim retrieval. In *Proceedings of the 19th International Workshop on Semantic Evaluation*, SemEval 2025, Vienna, Austria.

Matúš Pikuliak, Ivan Srba, Robert Moro, Timo Hromadka, Timotej Smoleň, Martin Melišek, Ivan Vykopal, Jakub Simko, Juraj Podroužek, and Maria Bielikova. 2023. Multilingual Previously Fact-Checked Claim Retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16477–16500, Singapore. Association for Computational Linguistics.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a Known Lie: Detecting Previously Fact-Checked Claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. 2024. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15933–15946, Bangkok, Thailand. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *CoRR*, abs/2104.08663.

A MultiClaim Dataset

MultiClaim dataset (Peng et al., 2025) was created to overcome the lack of data for crosslingual and non-English PFCR. This task was previously done mostly in English while many other languages or even language families were not considered. The new dataset contains 205,751 fact-checks in 39 languages and 28,092 social media posts in 27 languages. With the help of professional fact-checkers, 31,305 pairs of posts and corresponding fact-checks were gathered, out of which 4,212 pairs are crosslingual, meaning that the language of the post and the fact-check is different. The organizers provided datasets for the training, development and testing stages. Three datasets are provided for each stage; fact checks, social media posts, and mappings between them. Each post is paired with at least one fact-check. The use of any external data apart from the Shared Task Dataset to prepare the submission was not allowed, but using pre-trained language models and data augmentation of the Shared Task Dataset was.

B Implementation Details

We used the Massive Text Embedding Benchmark (MTEB)[7] as a starting reference point for the choice of retriever and reranker models. We implemented a BM25 model with BM25Okapi[8]. For cross-encoder implementation, we used Sentence-Transformer[9] library, and for bi-encoders AutoModel from transformers library[10].

Due to the large sizes of the models, we used mixed precision to improve efficiency, reduce memory footprint, and accelerate computation.

Inference was conducted on NVIDIA GeForce GTX 1080 Ti or NVIDIA TITAN RTX GPUs.

C Multilingual Preprocessing for Lexical Models

The pipeline standardizes case and whitespace, removes URLs, HTML entities, punctuation, digits, and normalizes Unicode to ASCII for consistency. While we experimented with transcribing emojis into text, removing them consistently improved retrieval. Repeated characters are collapsed, stopwords are eliminated using language-specific resources, and dates are converted to a standardized (month day, year e.g. February 12, 2025) format. The most impactful step was ensuring the correct matching of inflected words through lemmatization. We used WordNetLemmatizer[11] for English and Simplemma[12] for other languages. We also evaluated stemming, but it reduced retrieval performance likely due to overly aggressive reductions that produced non-standard word forms which no longer matched fact-checked claims, weakening lexical alignment. While digit removal had a minimal effect, date normalization improved retrieval. We evaluated grammar and spell correction; however, reduced performance due to overcorrection altering key terms.

D Ensemble Weighting

BM25	E5	BGE	S@100
1.0	1.0	1.0	0.8947
0.5	1.0	1.0	0.8967
0.25	1.0	1.0	0.8940
0.5	2.0	1.0	0.8856
0.5	1.0	2.0	0.8947

Table 6: Comparison of ensemble weighting schemes on S@100 performance in the crosslingual setting

The results in Table 7 show that reducing BM25's weight while maintaining higher weights for semantic models in the monolingual setting leads to improved retrieval effectiveness. Specifically, assigning a weight of 0.5 to BM25 and 1.0 to both E5 and BGE achieves the highest average S@100 score of 0.9720, slightly outperforming the equal-weighted (1.0, 1.0, 1.0) ensemble, which scored 0.9718. Further reducing BM25's weight to 0.25 (0.25 BM25 + 1.0 E5 + 1.0 BGE) resulted in a slight performance drop with S@100 of 0.9711, indicating that BM25 still provides useful lexical matching and should not be completely minimized. Interestingly, increasing the weight of E5 to 2.0 (0.5 BM25 + 2.0 E5 + 1.0 BGE) or BGE to 2.0 (0.5 BM25 + 1.0 E5 + 2.0 BGE) led to slightly lower performance (0.9701 and 0.9694, respectively), suggesting that the ensemble benefits from the strengths of each semantic model, rather than favouring one over the other.

Table 6 compares ensemble weighting schemes on S@100 performance in the crosslingual setting. Consistent with monolingual results, reducing

[7] https://huggingface.co/spaces/mteb/leaderboard

[8] https://pypi.org/project/rank-bm25/

[9] https://sbert.net/

[10] https://huggingface.co/transformers

[11] https://www.nltk.org/_modules/nltk/stem/wordnet.html

[12] https://github.com/adbar/simplemma

BM25's weight while maintaining higher weights for semantic models improved performance.

E BM25 Hyperparameters

To approximate the optimal hyperparameters b and k_1 for the BM25 model, we conduct a grid search. The optimal value for b is 0.85 and for k_1 it is 1.5 which means a higher normalization of the field length and slower term frequency saturation in comparison to the default settings.

BM25	**E5**	**BGE**	**ARA**	**DEU**	**ENG**	**FRA**	**MSA**	**POR**	**SPA**	**THA**	**AVG**
1.0	1.0	1.0	0.9726	0.9742	0.9497	0.9731	0.9805	0.9632	0.9651	0.9962	0.9718
0.5	1.0	1.0	0.9708	0.9742	0.9509	0.9722	0.9825	0.9632	0.9662	0.9962	**0.9720**
0.25	1.0	1.0	0.9691	0.9742	0.9513	0.9722	0.9805	0.9608	0.9641	0.9962	0.9711
0.5	2.0	1.0	0.9657	0.9649	0.9521	0.9722	0.9825	0.9616	0.9655	0.9962	0.9701
0.5	1.0	2.0	0.9657	0.9719	0.9477	0.9749	0.9766	0.9592	0.9631	0.9962	0.9694

Table 7: Ensembler performance comparison on S@100 based on different ensemble weighting schemes in the monolingual setting