JU_NLP at SemEval-2025 Task 7: Leveraging Transformer-Based Models for Multilingual & Crosslingual Fact-Checked Claim Retrieval

Atanu Nayak[⋆], **Srijani Debnath**[◇], **Arpan Majumdar**[†], **Pritam Pal**[⋆], and **Dipankar Das**[⋆]
[⋆]Jadavpur University, Kolkata, India
[◇]Government College of Engineering and Leather Technology, Kolkata, India
[†]University of Kalyani, Kalyani, Nadia, India
{nayak.primary, srijanidebnath2005, arpanmajumdar952, pritampal522, dipankar.dipnil2005}@gmail.com

Abstract

Fact-checkers are often hampered by the sheer amount of online content that needs to be fact-checked. NLP can help them by retrieving already existing fact-checks relevant to the content being investigated. This paper presents a systematic approach for the retrieval of top-k relevant fact-checks for a given post in a monolingual and cross-lingual setup using transformer-based pre-trained models fine-tuned with a dual encoder architecture. By training and evaluating the shared task test dataset, our proposed best-performing framework achieved an average success@10 score of 0.79 and 0.62 for the retrieval of 10 fact-checks from the fact-check corpus against a post in monolingual and crosslingual track respectively.

1 Introduction

The rapid proliferation of misinformation across social media platforms has made manual fact-checking an increasingly daunting task. Automated retrieval systems, powered by Natural Language Processing (NLP) techniques, offer a scalable solution by identifying and presenting previously verified fact-checks relevant to new claims. In this work, we present a robust fact-check retrieval framework that leverages transformer-based dual encoder architectures, fine-tuned separately for monolingual and cross-lingual settings.

Our framework involves three state-of-the-art pre-trained models: GTR-T5 (Ni et al., 2021a) for both monolingual and coss-lingual fact-check retrieval, E5-Large-v2 (Wang et al., 2022) and MiniLM (Wang et al., 2020) for cross-lingual retrieval. Evaluated on the *SemEval 2025 Shared Task 7: Multilingual and Crosslingual Fact-Checked Claim Retrieval* dataset, our proposed system achieves a Success@10 score of 0.79 on the test set with GTR-T5 in the monolingual track. For the cross-lingual track, GTR-T5 and E5-Large-v2 achieved Success@10 scores of 0.62 and 0.58 on the test set, respectively. In addition, a MiniLM-based framework was also developed as a lightweight alternative that converts posts and fact-checks into normalized vector embeddings using MiniLM-L12-v2, which are then indexed with FAISS for rapid retrieval.

The proposed retrieval system not only addresses the challenge of the vast online misinformation but also provides a scalable solution that can be adapted to diverse multilingual environments.

2 Related Work

Early fact-checking retrieval systems relied on keyword-based and traditional IR methods, which lacked semantic understanding and multilingual support. Neural IR models like DRMM (Guo et al., 2016) and MatchPyramid (Pang et al., 2016) improved performance but struggled with scalability and cross-lingual generalization.

Transformer-based models such as BERT (Devlin et al., 2019) and Sentence-BERT (Reimers and Gurevych, 2019) enabled dense vector representations and improved semantic retrieval. Dual encoder models like GTR-T5 (Ni et al., 2021b) and E5 (Wang et al., 2022) further enhanced efficiency by allowing independent query-document encoding, making them suitable for large-scale applications.

Earlier approaches such as DSSM (Huang et al., 2013) and KNRM (Xiong et al., 2017) demonstrated the potential of deep learning for retrieval tasks but were often limited by shallow interaction architectures and difficulties in handling long input sequences. These models, while offering initial improvements over traditional IR, did not fully capture the complex semantic relationships necessary for effective claim retrieval, especially in multilingual contexts.

Multilingual models like **LaBSE** further pushed the boundaries of multilingual semantic retrieval.

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 2084–2089

3 Data

All textual analysis and experiments were performed using data from the *SemEval 2025 Shared Task 7: Multilingual and Crosslingual Fact-Checked Claim Retrieval* (Peng et al., 2025). During the training and development phase, the dataset consisted of approximately 24,431 social media posts in multiple languages, 153,743 fact-checked claims, and 25,743 post-to-fact-check pairs where each post was linked to at least one fact-check claim. The 24,431 posts were further divided into cross-lingual and monolingual tracks where 18,907 posts were used for monolingual evaluation and 5,524 posts were used for cross-lingual evaluation.

Furthermore, in the monolingual track, the 18,907 posts and 153,743 fact checks were distributed into eight different languages: French (fra), Spanish (spa), English (eng), Portuguese (por), Thai (that), Deutsch (due), Modern Standard Arabic (msa) and Arabic (ara). The distribution of monolingual and crosslingual data are provided in Figure 1 for both training and development sets.

For the testing dataset, there was a total of 272,447 fact checks distributed over 10 languages (8 languages were the same as training and development sets, 2 extra languages were added: Polish or pol and Turkish or tur) and 8,276 posts. Among 8,276 posts, 4000 posts were for cross-lingual tracks and the remaining posts were for monolingual tracks. The overall data distribution for test data is provided in Figure 2

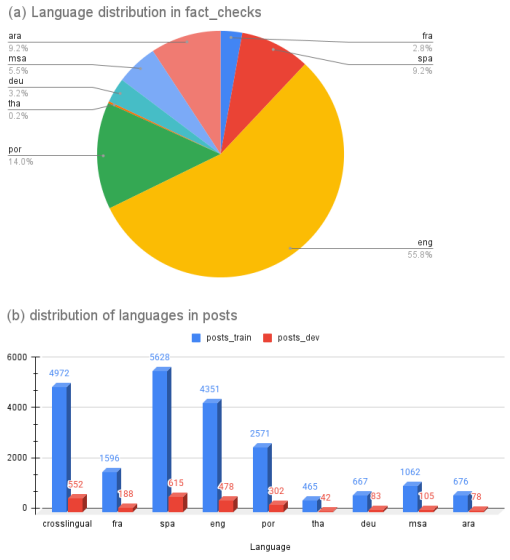

Figure 1: Distribution of training and development data

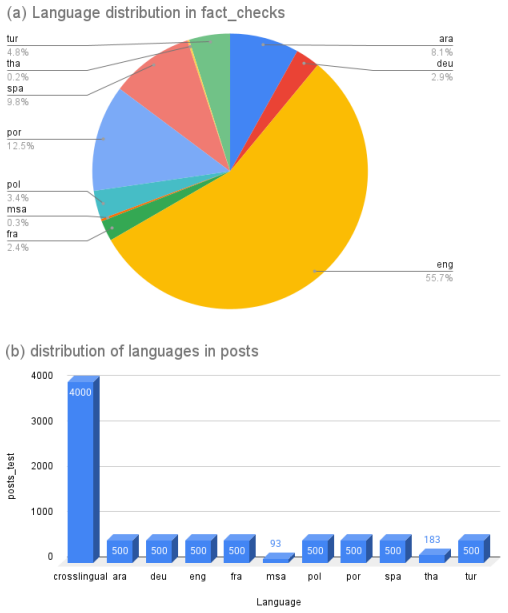

Figure 2: Distribution of test data

4 Methodology

This section briefly discusses the methodologies used to develop our proposed frameworks.

4.1 Text Preprocessing

Before diving into the actual system development and training, a few post-processing steps were applied such as 1) Removal of escape characters (e.g., `\n`, `\t`), 2) Decoding of Unicode characters, 3) OCR and post text were concatenated, 4) Tokenization of texts into tokens etc.

4.2 Framework Development

This section outlines the development of the proposed framework for the cross-lingual and monolingual relevant fact-check retrieval system.

We selected GTR-T5, E5-Large-v2, and MiniLM based on extensive evaluation across multilingual retrieval benchmarks. GTR-T5 is pretrained on large-scale multilingual corpora and fine-tuned for dense retrieval using contrastive learning, resulting in strong performance in zero-shot and multilingual retrieval tasks. E5-Large-v2, on the other hand, is optimized for retrieval-specific objectives such as passage ranking, supports multiple languages with task-specific embeddings, and remains efficient and scalable for large datasets. MiniLM was chosen for its lightweight and fast architecture, making it ideal for real-time applications while providing a good trade-off between

speed and retrieval performance.

The benchmark dataset allowed us to evaluate all models under controlled and consistent conditions, ensuring fair comparison across multilingual pairs. GTR-T5, E5-Large-v2, and MiniLM emerged as top performers based on retrieval accuracy, latency, and generalization to low-resource settings. These models demonstrated robustness across diverse language directions, including low-resource to high-resource queries and vice versa. In contrast, models that were not selected showed inferior performance on key metrics such as MRR, Recall@k, and precision, further justifying their exclusion from final deployment.

4.2.1 Dual Encoder with E5-Large-v2 and GTR-T5

The framework leverages a dual-encoder architecture used for independent fine-tuning of two transformer-based pre-trained models: E5-Large-v2 and GTR-T5. E5-Large-v2 was pre-trained on large-scale retrieval tasks and fine-tuned on datasets such as MS MARCO (Craswell et al., 2021) and various multilingual benchmarks, rendering them highly suitable for cross-lingual retrieval. In contrast, GTR-T5 is specifically designed for dense retrieval tasks and has been pre-trained on extensive monolingual datasets, making it highly effective for monolingual fact-check retrieval. Accordingly, E5-Large-v2 was employed for the cross-lingual track while GTR-T5 was employed for both the monolingual and cross-lingual tracks.

In this dual-encoder framework, separate encoders process both the query (i.e. posts) and passage (i.e. fact-check) and then yield dense vector representations. The dot product similarity scores matrix between these representations quantifies the relevance of the passage to the query. The overall model flow diagram is provided in Figure 3

Framework Description: The dual encoder architecture for the mentioned two models consists of two independent encoders—one for the query (i.e., post) and one for the passage (i.e., fact-check). For the E5-Large-v2, the encoders are implemented using `TFBertModel` to tokenize input queries and passages. GTR-T5 uses `TFT5EncoderModel` for both encoders, enabling it to process input sequences efficiently.

Let the input query/post be Q (e.g., "Is climate change real?") and passage/fact-check be P (e.g., "Scientific consensus states climate change is happening."), which are tokenized into input IDs using

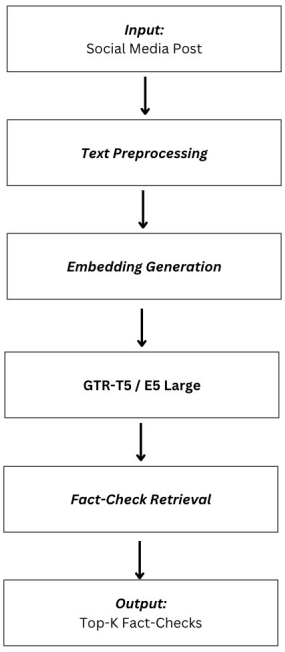

Figure 3: Flow diagram of E5-Large and GTR-T5 based frameworks

the common tokenizer. Input IDs of each input Q and P are passed through the query encoder and passage encoder to generate Query Embedding (E_Q) and Passage Embedding (E_P), which are dense vector representations. The embeddings are then normalized:

$$E_Q = \frac{E_Q}{\|E_Q\|}, \quad E_P = \frac{E_P}{\|E_P\|} \tag{1}$$

The model computes the cosine similarity to find similarity scores between all query and passage embeddings in the batch:

$$S_{ij} = E_{Q_i} \cdot E_{P_j} \tag{2}$$

where S_{ij} is the similarity score between the i^{th} query and the j^{th} passage. This results in a similarity score matrix $S \in \mathbb{R}^{n \times n}$ for a batch of size n.

4.2.2 A lightweight framework using MiniLM

This approach was built around three key components: how we represent the data, how we retrieve relevant fact-checks, and how we fine-tune our model to improve accuracy. The overall flow-diagram for the MiniLM-based framework is provided in Figure 4.

Data Representation: To compare social media posts with fact-checked claims, we first converted them into vector embeddings using the all-

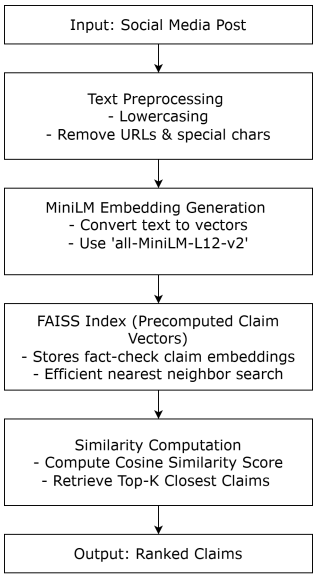

Figure 4: Flow diagram MiniLM-based framework.

MiniLM-L12-v2 (Wang et al., 2020) model. This helped us to capture similarities in meaning, even across different languages, so that we can match posts with the most relevant fact-checks.

Framework Description: To find the right fact-check efficiently, we used FAISS (Facebook AI Similarity Search) (Douze et al., 2024) a fast and scalable tool for searching through large datasets. The retrieval process works as follows: We generated MiniLM embeddings for both social media posts and fact-check claims. To improve accuracy, we normalized these embeddings, ensuring that they have the same scale before comparison. We then used the FAISS indexing system to quickly find and retrieve the most relevant fact checks based on similarity.

4.3 Training

During training, the contrastive loss ensured that the model maximized the similarity for positive pairs and minimized it for negative pairs. For each query-passage pair in the batch:

$$y_i = \begin{cases} 1, & \text{if positive pair} \\ 0, & \text{if negative pair} \end{cases} \tag{3}$$

A margin m is used to separate positive and negative pairs (e.g., $m = 0.2$):

$$L = y_i(1 - S_i)^2 + (1 - y_i)\max(0, S_i - m)^2 \tag{4}$$

The contrastive accuracy measured how well the model classified positive and negative pairs using a threshold τ (e.g., $\tau = 0.5$):

$$\hat{y}_i = \begin{cases} 1, & \text{if } S_i \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

4.4 Retrieval of Top-K Relevant Fact-Checks

For a given query or post Q, the similarity scores were computed for all passages (or fact-checks) in the corpus:

$$Scores_P = [S_1, S_2, \ldots, S_n] \tag{6}$$

The passages were then ranked by their similarity scores, and the top-K fact-checks with the highest scores were retrieved for the given post:

$$Top\text{-}K = \text{argsort}(-Scores_P)[:K] \tag{7}$$

In our experiments, we chose $K = 10$. This means we retrieved the top 10 fact-checks with the highest scores.

4.5 Fine Tuning

The proposed GTR-T5 framework was fine-tuned with a batch size of 16 and a learning rate of 3e-5 with PyTorch as the deep learning framework. The model was trained for 5 epochs, and early stopping was applied to prevent overfitting. The contrastive loss function with a margin of 0.2 was used to optimize the model and the Adam (Kingma and Ba, 2017) optimizer was used for gradient updates.

The proposed E5-Large-v2 framework was trained with TensorFlow as the deep learning framework and fine-tuned for 1 epoch with a learning rate of 1e-5. No layer of the model was frozen during training. The optimizer was chosen as Adam and the batch size was taken as 2. The loss function used was contrastive loss with a margin of 0.2 and the optimizer was chosen as Adam. The accuracy was calculated during fine-tuning of the model with the help of the contrastive accuracy function.

E5-Large-v2 showed early saturation in validation metrics, and training beyond one epoch led to performance degradation due to overfitting; hence, it was fine-tuned for only one epoch. In contrast, GTR-T5-Large, with its larger architecture and slower learning dynamics, required fine-tuning for five epochs to achieve convergence.

The miniLM model was trained with a batch size of 32 and a learning rate of 2e-5 with PyTorch as the deep learning framework and the 'MultipleNegativesRankingLoss' loss function was used. To

test the impact of extended fine-tuning, we experimented with varying training times, using runs that lasted 3 and 10 epochs. However, the best result was produced at epoch 10 and reported in Table 1 in Section 5. To prevent overfitting, we applied dropout layers and weight decay in the respective model.

All the above models were trained and evaluated on the Kaggle platform using the NVIDIA Tesla T4 GPU.

5 Result

All the proposed frameworks were evaluated on the development and testing datasets using the Success@10 metric by retrieving the top 10 fact-checks from the corpus. The success@10 metric can be defined as:

$$\text{Success@10} = \begin{cases} 1, & \text{at least one fact-chek in top 10,} \\ 0, & \text{otherwise.} \end{cases}$$

The overall results are provided in Table 1 where we can see the GTR-T5-based framework provides the best performance in both monolingual and cross-lingual tracks for both development and test datasets. The E5-Large-v2 and MiniLM models didn't perform well and we can see a performance downgrade of 6.45% and 22.58% in the test data for the mentioned models respectively compared to the GTR-T5 model.

Track	Model	Dev	Test
Monolingual	GTR-T5	0.77	0.79
Crosslingual	GTR-T5	0.59	0.62
	E5-Large-v2	0.58	0.58
	MiniLM	0.51	0.48

Table 1: Success@10 results for monolingual and crosslingual retrieval in development and test phases

6 Conclusion

In this article, we proposed GTR-T5-based monolingual and cross-lingual frameworks and E5-Large-v2 and MiniLM-based cross-lingual frameworks only for fact-checked claim retrieval from social media posts. Our experiments show that the GTR-T5 model works well for both monolingual and cross-lingual settings with success@10 scores of 0.79 and 0.62 respectively in the test dataset. These results underscore the robustness of the proposed models in their respective tasks. However, further optimization is needed to improve recall for lower-ranked fact-checks and enhance cross-lingual retrieval performance. Future work will explore incorporating language mapping using multilingual transformer-based embeddings (TEMs) and employing advanced fine-tuning techniques to further improve performance. Also, we will experiment with the E5-large-v2 and MiniLM models for monolingual settings in our future work.

7 Limitations

Although the models perform well in retrieving relevant fact-checks, several limitations remain for monolingual and cross-lingual frameworks.

In the monolingual setting, while the proposed framework achieved a Success@10 of 0.79 in the test phase, there is still room for improvement in retrieving lower-ranked fact-checks. Additionally, the model's performance on low-resource languages within the same language family remains suboptimal.

In the case of the cross-lingual framework, a Success@10 of 0.62 was achieved in the test phase using GTR-T5, but the result was not impressive in the E5-Large-v2 and MiniLM-based models. One possible reason behind the batch size being restricted to 2 in the E5-Large-v2 model is that it may downgrade performance. In our future work, we will use higher batch sizes to determine whether the performance improves. Also, there is some scope for more hyperparameter tuning in the MiniLM model to improve performance, which we'll try in our future work.

References

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Ms marco: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1566–1576, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64. ACM.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alejandro Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 2333–2338.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021a. Large dual encoders are generalizable retrievers.

Jianmo Ni, Wen-tau Yih, and Nick Craswell. 2021b. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2793–2799.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–64.