Top Ten From Lakhs: A Transformer-based Retrieval System for Identifying Previously Fact-Checked Claims Across Multiple Languages

Srijani Debnath[⋆], **Pritam Pal**[◇], and **Dipankar Das**[◇]
[⋆]Government College of Engineering and Leather Technology, Kolkata, India
[◇]Jadavpur University, Kolkata, India
{srijanidebnath2005, pritampal522, dipankar.dipnil2005}@gmail.com

Abstract

The efficient identification of previously fact-checked claims across multiple languages is a challenging task. It can be time-consuming for professional fact-checkers even within a single language. It becomes much more difficult to perform manually when the claim and the fact-check may be in different languages. 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 two transformer-based fact-checked claim retrieval frameworks that share a common preprocessing pipeline but differ in their underlying encoder implementations: **TIDE**, a TensorFlow-based custom dual encoder applied to english-translated data, and **PTEX**, a PyTorch-based encoder operating on both english-translated and original-language inputs, and introduces a lightweight post-processing technique based on a textual feature: **Keyword Overlap Count** applied via reranking on top of the transformer-based frameworks. Training and evaluation on a large multilingual corpus show that the fine-tuned E5-Large-v2 model in the PTEX framework yields the best monolingual track performance, achieving an average Success@10 score of 0.8846 and the same framework model with post-processing technique achieves an average Success@10 score of 0.7393 which is the best performance in crosslingual track.

1 Introduction

The rise of user-generated content on social media presents major challenges for fact-checkers, especially when claims and fact-checks span multiple languages. Automating verified claim retrieval can streamline verification, reduce manual effort, and speed up responses to misinformation. This paper aims to develop and evaluate multilingual transformer-based claim retrieval frameworks using dual-encoder architectures, fine-tuning strategies, and lightweight post-processing reranking. The main contributions of this paper are:

- We developed **TIDE**, a TensorFlow-based dual encoder framework that used E5-Large-v2 (Wang et al., 2022) and GTR-T5-Large (Ni et al., 2021c), fine-tuned on english-translated data.
- We developed **PTEX**, a PyTorch-based encoder framework integrating English-only encoders (GTE-Large (Li et al., 2023), GTR-T5-Large, MiniLM-L12-v2 (Wang and Liu, 2020), E5-Large-v2) and multilingual encoders (Multilingual-E5-large (Gao and Callison-Burch, 2023), Multilingual-E5-base (Gao and Callison-Burch, 2023), paraphrase-xlm-r-multilingual-v1 (Reimers and Gurevych, 2019), paraphrase-multilingual-MiniLM-L12-v2 (Reimers and Others, 2020), paraphrase-multilingual-mpnet-base-v2 (Song and Others, 2020)), to process english-translated and original-language texts.
- We introduced a **Keyword Overlap Count** textaul feature that compared named entities and nouns between a post and candidate claims.
- Finally, we applied a lightweight post-processing reranking step that linearly combined transformer-based similarity scores with the Keyword Overlap Count feature, enhancing retrieval accuracy with minimal computation and no model retraining due to its lightweight nature.

2 Related Work

Early fact-checking systems used keyword matching and classical IR methods like BM25 and TF–IDF, which lacked semantic and cross-lingual understanding. Neural IR models (e.g., DSSM (Huang et al., 2013), DRMM (Guo et al., 2016)) introduced learned interactions but struggled with long texts. Transformer models (e.g., BERT (Devlin et al., 2019), SBERT (Reimers and Gurevych, 2019)) improved contextual understanding, while dual encoders like GTR-T5 (Ni et al., 2021b) and E5 (Wang et al., 2022) enabled efficient dense retrieval via ANN. These advances led us to

Proceedings of Recent Advances in Natural Language Processing,pages 268–274
Varna, Sep 8–10, 2025
https://doi.org/10.26615/978-954-452-098-4-033

adopt E5-Large-v2 and GTR-T5-Large in TensorFlow (TIDE) and PyTorch (PTEX) frameworks. Language-agnostic models like LaBSE (Feng et al., 2020) generalize across languages but lag behind English-specialized encoders on translated data. Compact multilingual models (e.g., paraphrase-XLM, multilingual-MiniLM) narrow this gap but still trail fine-tuned English models. Extending prior work, we evaluated English-only and multilingual encoders on both english-translated and original-language claims, finding English text encoders (mainly E5-Large-v2) to be the most effective for cross-lingual claim retrieval.

3 Dataset

All textual analysis and experiments were based on the *SemEval 2025 Shared Task 7: Multilingual and Crosslingual Fact-Checked Claim Retrieval* (Peng et al., 2025)[1]. The training and development dataset included 24,431 multilingual social media posts, 153,743 fact-checked claims, and 25,743 post-to-fact-check pairs, with each post linked to at least one claim. Posts were divided into monolingual (18,907) and cross-lingual (5,524) evaluation tracks. The monolingual track covered eight languages: French (fra), Spanish (spa), English (eng), Portuguese (por), Thai (tha), German (deu), Modern Standard Arabic (msa), and Arabic (ara). The test set comprised 272,447 fact checks in ten languages, the above eight plus Polish (pol) and Turkish (tur), and 8,276 posts (4,000 used for cross-lingual and the rest for monolingual evaluation). Figure 1 shows the data distribution: left for training and development, right for test.

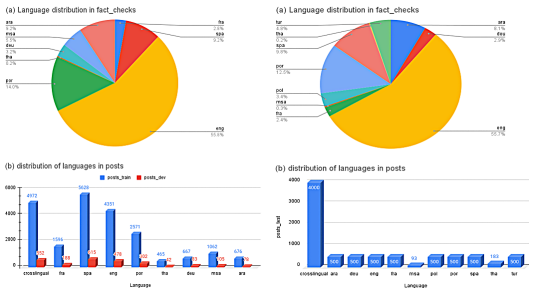

Figure 1: Training, development & test data distribution

4 Methodology

This section discusses the methodologies used. For a given post P, our main objective is to retrieve top 10 of it's most relevant fact-checked claims.

[1] https://disai.eu/semeval-2025/

4.1 Text Preprocessing

Few pre-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. 5) Removal of emojis using the `emoji` package[2]. 6) Elimination of special characters, extra spaces, and brackets via regular expressions.

4.2 Textual Analysis and Feature Extraction

Keyword Overlap Count: We extracted salient lexical items: proper nouns, common nouns, and named entities, from each social media post and fact-check claim using the spaCy toolkit (Honnibal and Montani, 2017). For example, from the post "Climate change summit in Paris kicks off today," extracted keywords include "climate," "change", "summit", and "Paris." The overlap between the two keyword sets is quantified to measure surface-level semantic alignment. This integer feature complements dense-vector similarity by highlighting direct lexical matches. Let K_{post} denote the set of keywords from the post and K_{fc} the set from a candidate claim. We define the *Keyword Overlap Count* feature, denoted f_{kw}, as the cardinality of their intersection:

$$f_{\text{kw}} = \left| K_{\text{post}} \cap K_{\text{fc}} \right|.$$

4.3 Framework Development

The following subsections detail each framework's unique models, architecture, and fine-tuning, all built on a common dual encoder setup and unified retrieval process but each framework leverages its respective platform and model nuances.

4.3.1 TIDE: TensorFlow Inference Dual Encoder

Overview and Models Used: TIDE, built in TensorFlow, used a dual encoder setup with two encoders initialized from a shared pre-trained model to separately represent social media posts (queries) and fact-check claims (passages). We implemented the system using pre-trained models: E5-Large-v2 via `TFBertModel` and GTR-T5-Large via `TFT5EncoderModel`. E5-Large-v2 generates 1024-dimensional embeddings through weakly-supervised contrastive pre-training, performing well on BEIR(Thakur et al., 2021) and MS-MARCO(Craswell et al., 2021). GTR-T5-Large,

[2] https://pypi.org/project/emoji/

pre-trained on large-scale QA tasks and fine-tuned on MS-MARCO, offers strong zero-shot generalization across domains (Ni et al., 2021a).

Framework Description: Instead of using an out-of-the-box wrapper, we implemented a custom `E5DualEncoder` class extending `tf.keras.Model`, as shown in Figure 2. It contained two encoder instances (for query and passage), both initialized from a shared base model. During the forward pass for a query Q and a passage P, it accepted batched token-ID tensors, generated pooled (CLS) embeddings (Devlin et al., 2019) using the shared encoder E_{TF} and pooler, applied L2 normalization, and computed cosine similarity via dot product.

$$\mathbf{q} = \text{pooler}\big(E_{\text{TF}}(Q)\big), \quad \mathbf{p} = \text{pooler}\big(E_{\text{TF}}(P)\big).$$

The Euclidean (L2) norm was used to normalize vectors (Reimers and Gurevych, 2019):

$$\mathbf{q}' = \frac{\mathbf{q}}{\|\mathbf{q}\|}, \quad \mathbf{p}' = \frac{\mathbf{p}}{\|\mathbf{p}\|}.$$

Cosine similarity S was computed as dot product of normalized embeddings (Mikolov et al., 2013):

$$S = \mathbf{q}' \cdot \mathbf{p}'.$$

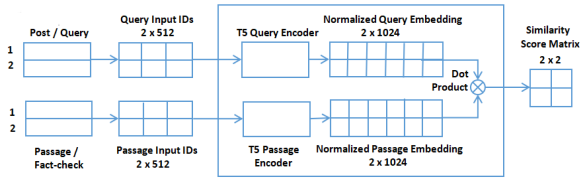

Figure 2: TIDE framework with GTR-T5-Large model

4.3.2 PTEX: PyTorch Text Encoder

Overview and Models Used: PTEX was implemented in PyTorch using the SentenceTransformer library (Reimers and Gurevych, 2019) and unified both English-only and multilingual text encoders within a single retrieval pipeline. It employed GTE-Large, pretrained on extensive relevance pairs to ensure robust semantic matching; GTR-T5-Large, which combined C4 pretraining with MS-MARCO fine-tuning for strong zero-shot performance; MiniLM-L12-v2, a compact, distilled encoder optimized for rapid inference; and E5-Large-v2, delivering high-quality 1024-dimensional embeddings via weakly-supervised contrastive pretraining. To support direct cross-lingual matching without translation overhead, PTEX integrated Multilingual-E5-large and Multilingual-E5-base, both trained on the CCAligned corpus, along with paraphrase-xlm-r-multilingual-v1, paraphrase-multilingual-MiniLM-L12-v2, and paraphrase-multilingual-mpnet-base-v2, chosen for their proven cross-lingual semantic similarity capabilities and efficient model sizes.

Framework Description: For a query Q and a passage P, each encoder E_i produced raw embeddings:

$$\mathbf{q}_i = E_i(Q), \quad \mathbf{p}_i = E_i(P).$$

Each vector was then normalized by its Euclidean norm to unit length (Reimers and Gurevych, 2019):

$$\mathbf{q}_i' = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|}, \quad \mathbf{p}_i' = \frac{\mathbf{p}_i}{\|\mathbf{p}_i\|}.$$

Final ensemble embeddings were computed as weighted sums of normalized encoder outputs:

$$\mathbf{q}_{\text{ens}} = \textstyle\sum_{i=1}^{N} \alpha_i \, \mathbf{q}_i', \quad \mathbf{p}_{\text{ens}} = \textstyle\sum_{i=1}^{N} \alpha_i \, \mathbf{p}_i'.$$

Finally, the retrieval score was calculated as the dot product of these unit-length ensemble vectors, equivalent to cosine similarity:

$$S(Q, P) = \mathbf{q}_{\text{ens}} \cdot \mathbf{p}_{\text{ens}}.$$

4.4 Training

TIDE Framework: The proposed TIDE framework models were fine-tuned using a contrastive loss function with a fixed margin to maximize similarity for positive pairs and suppress it for negatives (Hadsell et al., 2006). The models were fine-tuned using a learning rate of 1×10^{-5} with a batch size of 2 due to resource constraints and optimized using Adam (Kingma and Ba, 2015). Contrastive loss encouraged high similarity S_i for positives ($y_i = 1$) and penalized negatives ($y_i = 0$) using margin m:

$$L = y_i(1 - S_i)^2 + (1 - y_i) \cdot \max(0, S_i - m)^2$$

Contrastive accuracy was measured by thresholding similarity using τ (e.g., τ = 0.5):

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

PTEX Framework: The proposed PTEX models were fine-tuned using MultipleNegativesRankingLoss (Xiong et al., 2021), leveraging in-batch negatives to maximize positive pair similarity while minimizing negative ones. The training setup consisted of a batch size of 4 and one epoch, with warmup steps set to 10% of total steps. The optimizer used was Adam with learning rate scheduling to ensure stable training. The strategy has been applied for training on both english-translated and original-language datasets.

4.5 Retrieval Process

After obtaining normalized embeddings for a query and all passages in the corpus, each framework computed a similarity score vector:

$$\text{Scores}_P = [S_1, S_2, \ldots, S_n].$$

These scores are sorted in descending order, and the top-K passages are retrieved:

$$\text{Top-K} = \text{argsort}(-\text{Scores}_P)[:K].$$

4.6 Post-processing

Once each post–claim pair has been assigned a dense-vector similarity score s_{dl} which is the predicted output from the deep learning frameworks mentioned in Section 4.3, we incorporate the Keyword Overlap Count f_{kw} into a final reranking score. Let α represent the weight on the dense-vector score and β the weight on the keyword feature. Based on our experiments, we have set

$$\alpha = 0.95, \quad \beta = 0.05.$$

Then the combined reranking score is computed as

$$\text{score}_{\text{final}} = \alpha\, s_{\text{dl}} \;+\; \beta\, f_{\text{kw}}.$$

Here, s_{dl} denotes the cosine similarity from the transformer model, and f_{kw} is the Keyword Overlap Count defined in Section 4.2. The $\text{score}_{\text{final}}$ is the final score that is used to rerank fact checks for that post, and the top-K fact-checks per post are retrieved as mentioned in Section 4.5 for evaluation. This linear combination allows the reranker to favor claims that not only are close in semantic embedding space but also share explicit lexical content with the post. We opted for this feature-based post-processing technique due to its lightweight nature, requiring minimal additional computation and no model retraining. While more advanced retrieval techniques exist, they often involve higher computational cost and architectural complexity, making them less suitable for efficient post-processing in low-resource or real-time settings.

5 Evaluation

All the proposed frameworks were evaluated on the test datasets provided by the organizers of "SemEval-2025 Task 7" using the Success@10 metric by retrieving the top 10 fact-checks from the corpus. The Success@10 metric can be defined as:

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

For the reranking-based post-processing, the top 100 fact-checks for each post were first retrieved using the dense similarity scores from the dual encoder model. These were then reranked using the final score after reranking defined in Section 4.6, which combines semantic similarity with keyword overlap. The top 10 reranked fact-checks per post were finally used to compute Success@10.

6 Result

We evaluated our retrieval frameworks across four key dimensions: training platform (TensorFlow vs. PyTorch), the impact of fine-tuning, encoder type (English-only vs. multilingual), and data representation (english-translated vs. original-language). Table 1 presents the average Success@10 results of the frameworks without post-processing. Figure 4 shows salient monolingual and crosslingual comparisons.

Our TensorFlow-based **TIDE** framework using E5-Large-v2 and GTR-T5-Large achieved average crosslingual Success@10 of 0.525 (0.580 and 0.470, respectively). In contrast, the PyTorch-based **PTEX** framework attained an average crosslingual Success@10 of 0.688, a 31.05% relative improvement, highlighting the benefits of PyTorch's optimization strategies and model interoperability for large-scale, crosslingual retrieval tasks. Here, it can be inferred that the TIDE framework based model's performance was unsatisfactory in crosslingual and monolingual retrieval which is likely due to less mature optimization and integration of multilingual embeddings compared to PTEX.

Within the PyTorch-based PTEX framework, fine-tuning proved to be highly effective. Models without fine-tuning (GTE-Large, GTR-T5-Large, MiniLM-L12-v2, and E5-Large-v2) achieved an average crosslingual Success@10 of 0.662, while their fine-tuned counterparts achieved 0.706—a 6.65% relative gain. In particular, GTE-Large improved from 0.633 to 0.701 (+10.7%), GTR-T5-Large from 0.699 to 0.729 (+4.3%), MiniLM-L12-v2 from 0.602 to 0.664 (+10.3%), and E5-Large-v2 from 0.685 to 0.730 (+6.6%) (see Figure 4). Monolingual tracks exhibited similar trends, with the fine-tuned E5-Large-v2 attaining a Success@10 of 0.885 versus 0.856 before fine-tuning (Figure 4).

We further explored the impact of data representation by comparing english-translated and original-language claims in the PTEX framework. Crosslingual retrieval on the english-translated dataset

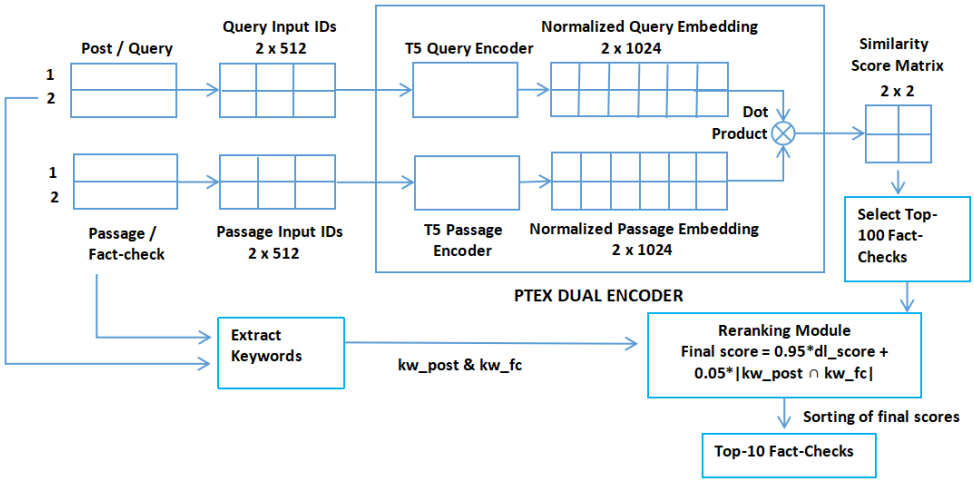

Figure 3: Proposed PTEX framework with GTR-T5-Large model and post-processing

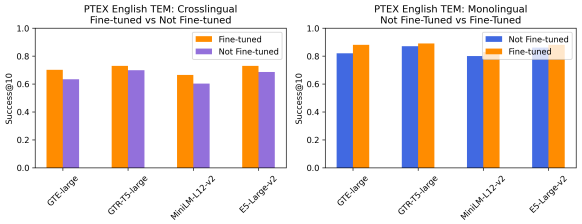

Figure 4: PTEX model performance (Crosslingual & Monolingual): Not Fine-tuned vs. Fine-tuned

achieved a slightly higher Success@10 (0.712 on eng vs. 0.702 on og). Thus, English translations may better leverage models pretrained on English corpora, despite possible translation noise.

To evaluate PTEX's multilingual ability, we compared fine-tuned multilingual encoders on english-translated (eng) vs. original-language (og) data across crosslingual and monolingual tracks. In the crosslingual track, performance dropped slightly for Multilingual-E5-large (0.7123 to 0.702) and Multilingual-E5-base (0.6625 to 0.6275). Similar declines were seen for paraphrase-multilingual-mpnet-base-v2 (7.2%), paraphrase-multilingual-MiniLM-L12-v2 (11%) and xlm-r-multilingual-v1 (9.4%), indicating challenges in directly handling non-English inputs. Conversely, in the monolingual track, some models improved on original-language data: Multilingual-E5-large (0.8539 to 0.8756) and Multilingual-E5-base (0.8212 to 0.8238), though others like paraphrase-multilingual-mpnet-base-v2, paraphrase-multilingual-MiniLM-L12-v2 and xlm-r-multilingual-v1 saw slight drops. These results underscore the generalization strength of multilingual models but also reveal their inconsistencies across languages, with English-only encoders (e.g., E5-Large-v2 at 0.730) still outperforming them in crosslingual settings.

Results after post-processing reranking: To enhance crosslingual retrieval, we applied a post-processing reranking stage using Keyword Overlap Count feature (Section 4.2). This was selectively applied to some of the fine-tuned PTEX models on english-translated data in crosslingual track. Figure 3 illustrates the PTEX framework using GTR-T5-large with post-processing reranking technique, where `kw_post` and `kw_fc` denote keywords from the post and candidate claim respectively, and `dl_score` is the predicted deep learning framework output. Table 2 shows Success@10 scores before & after reranking on english-translated data.

The incorporation of the Keyword Overlap Count reranker feature led to consistent gains across all evaluated models on the crosslingual track. E5-Large-v2 observed a 0.009 absolute improvement (+1.23%), GTR-T5-Large gained 0.003 (+0.41%), GTE-Large achieved a notable 0.013 increase (+1.85%), and Multilingual-E5-Large improved by 0.008 (+1.12%). These results showed that combining dense embedding similarity with lexical overlap helped reduce semantic drift, especially when retrieving from the large, multilingual corpus of 272K claims. We have applied reranking to the crosslingual track because its larger, heterogeneous candidate set demanded more robust disambiguation, whereas the relatively constrained, monolingual corpus already yielded high performance without reranking.

Framework	Model used	Finetuned	Dataset	Track	
				Mono	Cross
TIDE	*English Text Encoders*				
	E5-Large-v2	✓	eng	0.73	0.580
	GTR-T5-Large	✓	eng	0.62	0.470
PTEX	*English Text Encoders*				
	GTE-Large	✗	eng	0.820	0.633
	GTR-T5-Large	✗	eng	0.860	0.699
	MiniLM-L12-v2	✗	eng	0.802	0.602
	E5-Large-v2	✗	eng	0.856	0.685
	GTE-Large	✓	eng	0.872	0.701
	GTR-T5-Large	✓	eng	0.882	0.729
	MiniLM-L12-v2	✓	eng	0.831	0.664
	E5-Large-v2	✓	eng	0.885	0.730
PTEX	*Multilingual Text Encoders*				
	Multilingual-E5-large	✓	eng	0.8539	0.7123
			og	0.8756	0.702
	Multilingual-E5-base	✓	eng	0.8212	0.6625
			og	0.8238	0.6275
	paraphrase-xlm-r-multilingual-v1	✓	eng	0.7653	0.5738
			og	0.7449	0.5010
	paraphrase-multilingual-MiniLM-L12-v2	✓	eng	0.7672	0.5680
			og	0.7265	0.4580
	paraphrase-multilingual-mpnet-base-v2	✓	eng	0.7667	0.5573
			og	0.7209	0.4630

Table 1: Results for Fact-Checked Claim Retrieval without post-processing (Success@10).

Note: The dataset was split into english-translated (eng) and original-language (og) versions. English Text Encoders (e.g., TIDE, PTEX) were evaluated on the eng split, while Multilingual Encoders (e.g., PTEX) were evaluated on both eng and og splits.

Model	No Reranking	After Reranking
E5-Large-v2	0.730	0.739
GTR-T5-Large	0.729	0.732
GTE-Large	0.701	0.714
Multilingual-E5-Large	0.712	0.720

Table 2: Average Success@10 scores Before and After Post-processing Reranking in crosslingual track

7 Conclusion

In this study, we demonstrated that both the choice of training platform and the application of targeted fine-tuning were critical for advancing multilingual fact-checked claim retrieval. The TensorFlow-based TIDE framework delivered modest crosslingual performance (average Success@10 of 0.525), whereas our PyTorch-based PTEX implementation achieved substantially higher accuracy, with an average crosslingual Success@10 of 0.688, a 31.05% relative improvement. Within PTEX, fine-tuning consistently boosted retrieval effectiveness, and the additional post-processing reranker, which combined dense embedding similarity with a lightweight Keyword Overlap Count feature, further elevated the crosslingual Success@10 of the leading E5-Large-v2 model from 0.730 to 0.739. Overall, the fine-tuned E5-Large-v2 encoder in PTEX emerged as the best monolingual system (Success@10 of 0.885) and, when augmented by reranking, as the top crosslingual system (Success@10 of 0.739). This demonstrates that a fine-tuned English-only text encoder is optimal for monolingual tasks, while adding a simple lexical-overlap reranker proves most effective for crosslingual retrieval. The reranking step also improves multilingual encoders, narrowing their gap with English-specialized models, and consistently boosts crosslingual accuracy with minimal cost, highlighting the value of combining semantic and lexical signals in a scalable PyTorch pipeline.

8 Limitations

Our analysis had limitations: the English-based post-processing reranker was inapplicable to non-English inputs; we fine-tuned selected components rather than adopting full end-to-end training; and residual translation noise or OCR errors, especially in low-resource languages, may have impacted retrieval. Future work will explore multilingual reranking using named entities and nouns in non-English languages to evaluate it's cross-lingual retrieval accuracy, alongside joint embedding-lexical optimization and lightweight model compression.

Acknowledgement

This work was supported by Defence Research and Development Organisation (DRDO), New Delhi, under the project "Claim Detection and Verification using Deep NLP: an Indian perspective".

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. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Yu Feng, Baosong Qin, Lin Zhang, Di Jin, Nenghai Yu, and Siliang Zhao. 2020. Language-agnostic bert sentence embedding. *ACL*.

Tianlang Gao and Chris Callison-Burch. 2023. E5: Dense retrieval with expert examples in multilingual contexts. *ArXiv*.

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.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. In *Proceedings of the 1st Workshop on Open-DOORS in Computational Linguistics*, Sydney, Australia.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alexis 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. ACM.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

X Li, Y Zhang, and Others. 2023. Towards large-scale pretrained text encoders for information retrieval. In *CIKM*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

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. In *Proceedings of EMNLP*, pages 669–689.

Jifan Ni, Jing Li, Mauricio González, and et al. 2021b. A generalist framework for information extraction and retrieval. In *Findings of EMNLP*, pages 1234–1249. ACL.

Junnan Ni, Bowen Yao, and Others. 2021c. Large dual-encoder models for generalizable retrieval. *Transactions of ...*

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*. Association for Computational Linguistics.

Nils Reimers and Others. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *ACL*.

Kaitao Song and Others. 2020. Mpnet: Masked and permuted pre-training for language understanding. *NeurIPS*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models.

Wei Wang and Haixun Liu. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *ArXiv*.

Zhuyun Wang, Wei Ma, and Others. 2022. Text embedding models (e5): A next step for dense retrieval. In *Proceedings of ...*

Faxian Xiong, Mandar Thakur, Jimmy Lin, et al. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of SIGIR 2021*.