Semeval Task 7: Multilingual Fact-Checked Claim Retrieval

Contributors:
Ishika Kulkarni, Kiara LaRocca, Shriya Sravani Y
ikulkar1@ucsc.edu, klarocca@ucsc.edu, sy4@ucsc.edu

Abstract

This report focuses on the SemEval-2025 Shared Task 7 and addresses the challenge of retrieving fact-checked claims in a multilingual context. The task aims to develop systems capable of retrieving the most relevant fact-checks for social media posts in various languages, assisting global efforts to combat misinformation. The evaluation is based on the systems' ability to rank fact-checked claims using similarity scores, treating the task as a binary classification problem. Additionally, this report highlights the complexities involved in integrating Machine Translation (MT) and Information Retrieval (IR) for Multilingual Information Retrieval (MLIR). Ultimately, the goal is to enhance scalable and efficient MLIR systems that can operate effectively in diverse linguistic contexts. The main models used to implement this task are Logistic Regression, BiLSTM with attention, and RoBERTa. The limitations of the dataset may affect the performance of each model, and solutions are addressed in this paper. Future experimentations include Jina-ColBERT-v2 and RAGatouille, which are designed to alleviate the biggest issues of current transformer models using Late Interaction Architecture.

1 Introduction

Searching the internet, asking a smart speaker a question, and suggesting related products to a user are all extremely common instances of information retrieval (IR). This ability to quickly and accurately retrieve relevant information to a query is very valuable. The easiest IR models to implement are classical models, like vector models or probabilistic models (1). While these tasks are usually done in English, a pressing concern in NLP is the ability to perform the same tasks across multiple languages. There are less benchmarks available for multilingual information retrieval methods, but this field is currently expanding and advancing. For our task, we will be using a multilingual approach to retrieve the top-K relevant fact-checked claims for a given social media post. For our class, NLP 243, we attempted a multilingual task, but if the project is revised for SemEval 2025, we would reorganize our approach for mono- or crosslingual fact-checked claim retrieval.

2 Related Works

Our work was highly influenced by several different research papers related to MLIR. For our purposes specifically, we looked into creating a bi-LSTM, as the model made for fake news detection performed admirably, with an accuracy of 84% and an F1-macro score of 62.0 (10). Since we had classroom experience with these models, we found this to be a good place to start. While researching contextualized embeddings more thoroughly, we debated using the different BERT models written about in one paper (2). BERT models in general are great because they can be finetuned to specific tasks. The paper discusses the results and shortcomings of each model when it comes to multilinguality, and even though XLM-RoBERTa is a monolingual model, it significantly outperformed multilingual models and a randomized baseline. XLM-Roberta outperformed mBERT on cross-lingual classification by up to 23% on low-resource languages, and outperformed previous state of the art models by 5.1% average accuracy and 2.42% average F-1 Score (16). The paper *Ensemble Language Models for Multilingual Sentiment* also concluded that XLM-RoBERTa outperformed mBERT (4).

Future implementations could use Jina-ColBERT-v2, RAGatouille, and Matryoshka learned embeddings. Jina-ColBERT-v2 is specically designed to use XLM-RoBERTa as the backbone, but aims to reduce the large memory/computational constraints that come with such transformer models (14). Ragatouille was designed to work with ColBERT models to streamline tasks (3). Matryoshka embeddings

were presented as a good resource for scalable and hierarchical multilingual embeddings optimized for crosslingual tasks (9). Another study showed that ColBERT-X models with Multilingual Translate-Distill (MTD) were more effective than using previously proposed training techniques (15).

3 Our Task and Motivation

This task aims to evaluate the effectiveness of systems in retrieving relevant fact-checks for multilingual social media content, emphasizing the need for scalable, efficient models that can operate across diverse linguistic contexts and assist fact-checkers worldwide.

Our task, specifically, is to retrieve the top-k most relevant fact-check claims for each social media post in a multilingual setting. Given a post in one language and fact-checked claims in potentially different languages, the system calculates similarity scores to rank the most relevant claims. Although it seems like the task is the information retrieval itself, it is actually a binary classification task to determine whether a given fact-checked claim is relevant to the post, regardless of the fact-check verdict.

Our motivation is to help improve MLIR in general. Notably, the integration of Machine Translation (MT) and Information Retrieval (IR) for Multilingual Information Retrieval (MLIR) is not straightforward due to unique constraints. While MT engines can assist in translating queries and documents, MLIR often requires specialized approaches, such as IR-based indexing techniques, to bridge language gaps without explicit translation. Additionally, MLIR must accommodate the distinct nature of queries (disconnected words) and documents (coherent text), which highlights the need for a deeper integration of coupling IR and MT engines (7).

4 Proposed Approach

Our proposed approach uses the XLM-RoBERTa model, for its success in multilingual contexts. Our approach consists of the following steps:

1. Generate word embeddings for `ocr` and `claim` columns of our dataset.

2. Compute the Cosine Similarity between the two columns.

3. Use that value as input into our classifier which performs binary classification (where relevant claims receive the label 1).

4. For each post ID, compile a list of fact check IDs where the label $= 1$. For example, for the `post_id` 33, the results would be something like [33: 654, 8729, 1632] when k=3.

5. Using Top-K Accuracy, rank the result and retrieve the top-K matches. Output these to an output file.

6. Compare the output file to the ground truths.

5 The Dataset & Evaluation Metrics

5.1 The Dataset

Our dataset is the most extensive of its kind, encompassing 28,092 social media posts in 27 languages, 205,751 fact-checks in 39 languages, and 31,305 connections between these posts and the fact-checks. We will be using three separate files for our input, so we joined the datasets together and kept only relevant column. Our dataset was missing the 'post' column in *posts.csv*, and emails to the organizers were not answered in time. We used 'ocr' as our post column, which introduces a few potential issues/limitations. The first is that 'posts' would have only been in one language, but 'ocr' includes two. For Eng-Eng pairs, the text and translation are the same, thereby increasing our corpus size, but not the vocabulary size. Since not all posts include images, our dataset was originally about a quarter of the intended size. We left the text and translation in 'ocr' to help augment our data to be roughly the same size. In doing this, we effectively oversampled our majority class. In the future, we would just use the MultiClaim Dataset (17), as it is the dataset our own is based off of. Figure 1 in the Appendix is an example of two instances in our dataset. The first is Eng-Eng, and both parts of the tuple in 'ocr' are the exact same.

Our final dataset CSV includes the following columns:

- post_id: contains the id of the social media post
- fact_check_id: contains the id of the fact-checked claim
- ocr: this is the text and translation related to the image attached to the post

- claim: this is the claim and translated text
- label: if the (post_id, fact_check_id) appears in the *pairs.csv* file, it is a relevant claim and the label == 1.
- features: ocr and claim columns concatenated

Preprocessing includes normalizing and tokenizing, and the dataset does not automatically come normalized or tokenized, as this is a facet of feature engineering. Table 4 located in the Appendix depicts our dataset.

5.2 Evaluation Metrics

Evaluation metrics for this project were quite simple. In most cases we only needed to use accuracy and the classification report (precision/recall/F1) to evaluate the classifier. Since we only did binary classification, this was more than enough for intrinsic evaluation. To complete the task, we did include top-k when applicable. Since models like Logistic Regression calculated probabilities, top-k simply found the top probabilities, so it was an additional measure, but not an individual metric, as it ignores results outside of top-K results, even when they are relevant. For the transformer models, we included Mean Reciprocal Rank (MRR) and Precision@K (P@K) for more robust evaluation as these were recommended by task organizers.

MRR measures how well the system ranks the first relevant fact-check claim for each post. It calculates the reciprocal of the rank at which the first relevant claim appears and averages this across all posts. For this task, a higher MRR indicates that relevant claims are consistently retrieved early, making the system efficient and useful for fact-checkers (13).

P@K (6) ranks items such that the most relevant items appear at the top, to assess whether the model prioritizes relevant items effectively within the top-K results. It focuses on the quality of the most highly ranked results, which is crucial in applications where only the top few results receive an interaction (like search engines or recommendation systems).

6 Experiments

Logistic Regression (LR) was the recommended baseline model for many tasks per lecture slides, as it runs quickly and is easy to tune hyperparameters. We chose LR specifically because we are working on binary classification. The neural network models we used include bi-LSTM with Attention and XLM-RoBERTa. As mentioned earlier, we used a bi-LSTM as we had experience with it in class, and then XLM-RoBERTa as it outperformed many models in MLIR.

6.1 Baseline Model: Logistic Regression

We implemented two baseline models. The first model did not use pretrained embeddings, but it did include a list of different features and hyperparameters that we experimented with to try and achieve the best possible score.

- TF-IDF Cosine Similarity
- Length Difference
- N-gram Overlap
- Jaccard Similarity
- Levenshtein Distance
- Cosine-Length Interaction
- GridSearchCV

Metrics like TF-IDF Cosine Similarity, N-gram Overlap, and Jaccard Similarity assess semantic and contextual overlaps, while Length Difference and Cosine-Length Interaction capture variations in size and direction for nuanced comparisons. Levenshtein Distance helps with string-level edits. The biggest challenge when selecting features was choosing features that were not redundant. For example, when using cosine-length interaction, using length difference was unnecessary.

Hyperparameters used in GridSearch were taken from the Logistic Regression page from SKLearn documentation (12). The best results achieved were:

- Cross-Validation Accuracy: 0.6893
- Accuracy: 0.6903
- Macro F1: 0.66

The second iteration of the model included pretrained embeddings from HuggingFace (5). SentenceTransformers used Element-Wise Absolute Difference between embeddings. We used GridSearchCV here as well. The best results achieved were are below:

- Cross-Validation Accuracy: 0.7392
- Accuracy: 0.7394
- Macro F1: 0.74

Between the two models, evaluation metrics showed that Model 2 was more balanced overall, but leaves room for improvement. Model 1 was prone to False Positives, and missed a lot of relevant information. Since Logistic Regression lacks context in complex data, this could be the reason why Model 2 performed 4% better than Model 1. It is important to note, however, that using the data we did could also contribute to these scores. If we had 'posts' as a column, the numbers in general may look different. We could also use a OneVsRest or OneVsAll metric to compare the model's performance on each language. Since we skewed our own dataset, evaluating with this metric was not applicable, but could be in future experimentation. The classification report is visible in the Appendix as Figure 2.

6.2 Neural Network: bi-LSTM with Attention

The first Neural Network we created is a bi-LSTM with Attention model. Since we have used these in classwork, we attempted to use one here. Bi-LSTM (22) models with attention mechanisms are effective because they capture contextual dependencies in both directions. This allows nuanced relationships to be captured well. The attention mechanism focuses on the most relevant parts of the input sequence, improving the model's ability to rank and retrieve relevant items.

The bi-LSTM we implemented used the following features and evaluation metrics:

- TF-IDF Vectorizer (10000 max_features)
- Learning Rate (1e-5, 2e-5)
- Cross Entropy Loss
- Optimizers: Adam, AdamW, SGD
- Dynamic Padding
- Weight Decay Regularization
- Classification Report (Precision, Recall, F1)

All models had the Tf-IDf vectorizer and the cross entropy loss.

The first iteration of the model had a learning rate of 1e-5 and used the Adam optimizer. The results were as follows:

- Accuracy: 0.6534
- Macro F1: 0.6456

This model mislabeled certain claims due to overfitting. Moving forward, the models have been created by adding weight decay regularization and the dynamic padding to help the model learn and train better. This also helped in reducing overfitting.

The second model is created by incorporating multihead attention into the Bilstm model. The Adam optimizer is replaced with the AdamW optimizer. This model performed significantly better than the first model. It was further improved with batch normalization and dropout layers. The ReduceLROnPlateau (19) function was used to gradually decrease the learning rate of the model as the epochs increased.

The performance of this model is shown by the results below:

- Accuracy : 0.7143
- Macro F1 : 0.7056

This model gave the highest accuracy as it was able to balance out the relevant and irrelevant claims made.

The third model was created by replacing LSTM with GRU. A biGRU model was created with multihead attention and Stochastic Gradient Descent as a variation to the bi-LSTM model. The performance of this model is given below:

- Accuracy : 0.6751
- Macro F1 : 0.6654

This model with GRU did not perform well due to the difference between the LSTM and GRU working. In multilingual tasks, the relationships between tokens may span across larger contexts. Bi-LSTMs can better capture these dependencies due to their more sophisticated gating mechanisms.

Though bi-LSTMs are slow in training, they are known to be good with data retrieval and creating these models allowed us to see how the labels were being made for the given data.

6.3 Neural Network: XLM-RoBERTa

The proposed approach for this task used XLM-RoBERTa as our classifier. To implement this, we actually used a few different RoBERTa models to

compare each performance, and we experimented with the following features, parameters, and evaluation metrics:

- Models: XLM-RoBERTa-base, XLM-RoBERTa-large, RoBERTa-base
- Tokenizer: Auto, Self
- Similarity: Jaccard, Cosine
- Loss: BCE with Logit Loss, Triplet Loss
- Optimizer: Adam, AdamW
- Hidden State Embeddings
- Attention Masking
- Language Detection
- ReduceLROnPlateau Scheduler
- Gradient Clipping
- Weight Decay
- Accuracy
- Precision@K
- MRR

6.3.1 Version 1

We started by cleaning the data and making a super basic model with xlm-roBERTa-base which has 12 layers and 125 million parameters, to see how it would perform with different parameters. Here, we used a self-tokenizer, 3-layered neural networks ([512,256],[256,1]), dropout of 0.3, attention layer with a mask, and the loss function used was Binary Cross-Entropy with Logits (BCE Logit Loss) (20), and it was trained for 30 epochs.

The problems with this model were that the loss was not decreasing as well as expected, it generalized too well on the given data, showing signs of overfitting, and there were no new learnings in the last ten epochs.

Table 1: Version 1 Model Performance Metrics

Metric	Value
MRR	0.507
Accuracy	0.501
Loss	0.6931
Precision	0.600

6.3.2 Version 2

Similar to the previous version, this one also used similar data preprocessing. The only major change was the model. Here, we used XML-roBERTa-large (18), which used eight attention heads, a dropout of 0.25, and batch normalization. The large version of the model consists of 24 transformer layers, 1024 hidden dimensions, and 16 attention heads. It is trained on a diverse dataset of 2.5 TB of text in 100 languages.

We used gradient clipping, reduceLROnPlateau (19) with a cross-validation of 5 folds, and an Adam optimizer with weight decay to ensure that the model was not overfitting. We would assume that this model would give the best results, but it overfits the data irrespective of the measures and becomes too complex for the given dataset.

Table 2: Version 2 Model Performance Metrics

Metric	Value
MRR	0.500
Precision	0.400
Training Loss	0.690
Accuracy	0.490

6.3.3 Version 3

This version used roBERTa-base to tackle the problem. In this case, the preprocessing steps remain the same with some additional features, like removing the inconsistent casing, HTML tags, etc., that would improve the embeddings along with text normalization.

For the model, we added a similar 8-headed attention layer with 0.25 dropout and 5 fully connected neural network layers. The loss used was BCE with logit loss. To ensure the model was not overfitting, we used Adam w optimizer, weight decay, and ReduceLROnPlateau.

Table 3: Model Performance Metrics

Metric	Value
Precision@k	0.800
MRR	0.600
F1 Score	50
Accuracy	51

7 Analysis of Results & Conclusion

Overall, we found that Logistic Regression with pre-trained embeddings achieved the best results. The pre-trained embeddings introduced context LR typically struggles with. The bi-LSTM produced the best accuracy when paired with multihead attention and cosine similarity. Since it processes data sequentially, even though it works well with binary classification, it takes longer to train. RoBERTa-base was the best RoBERTa model we used. The best P@K value was 0.8, and the best MRR value was 0.58.

If our data had been augmented differently, the 'posts' column had been available (or the MultiClaim dataset used instead), the evaluation metrics may have changed. These results were achieved with severe limitations of our data, and it would not be a complete surprise to see the transformer model perform the best when the dataset is the intended dataset.

To further improve results, we could use Jina-ColBERT-v2 to alleviate high memory usage errors. This proposed model works well due to the Late Interaction Architecture that current models lack, and already implements the model we used. If we used this model, we could consider using RAGatouille as well, as it was designed to work well with ColBERT models. We learned a lot during this project, and we look forward to further experimentation.

8 Contributions

Our team contributions were pretty evenly split, as we each took a processing task and a model. Ishika did our original data preprocessing and visualization, and she implemented the XLM-RoBERTa model. Shriya helped manage references and worked on the bi-LSTM models. Kiara created the custom dataset, updated the report as we went, and created the baseline Logistic Regression models.

References

[1] Analytics Vidhya. (2021). Step by Step Guide to Master NLP: Information Retrieval. Retrieved from https://www.analyticsvidhya.com/blog/2021/06/part-20-step-by-step-guide-to-master-nlp-/information-retrieval/

[2] Devlin, J., et al. (2021). **BERT, mBERT, BiBERT: A Comparative Analysis for Multilingual NLP**. arXiv:2109.04588. Available at: https://arxiv.org/abs/2109.04588.

[3] Clavié, B. (n.d.). Ragatouille. Retrieved from https://ben.clavie.eu/ragatouille/#motivation

[4] Chen, Y., et al. (2024). **Ensemble Language Models for Multilingual Sentiment Analysis: The Role of XLM-RoBERTa**. arXiv:2403.06060v1. Available at: https://arxiv.org/html/2403.06060v1.

[5] Hugging Face. (n.d.). Sentence Transformers. Retrieved from https://huggingface.co/sentence-transformers

[6] Keylabs. (n.d.). Understanding Precision@K (P@K). Retrieved from https://keylabs.ai/blog/understanding-precision-at-k-p-k/

[7] Fluhr, C., Frederking, R. E., Oard, D., Okumura, A., Ishikawa, K., Satoh, K., Klavans, J., & Hovy, E. (1998). Multilingual (or Cross-lingual) Information Retrieval. In *NSF-EU MLIA Working Group White Paper*, Chapter 2. Retrieved from https://www.cs.cmu.edu/~ref/mlim/chapter2.html.

[8] Mansour, W., Elsayed, T., & Al-Ali, A. (2024). *Spotting Previously-Verified Claims Over Twitter*. Computer Science and Engineering Department, Qatar University, Qatar; KINDI Center for Computing Research, Qatar University, Qatar.

[9] Huang, K., et al. (2022). **Matryoshka: Learned Embeddings for Scalable Multilingual NLP**. arXiv:2205.13147. Available at: https://arxiv.org/abs/2205.13147.

[10] Nguyen, A., et al. (2022). **Bi-LSTM for Fake News Detection: A Lightweight and Interpretable Approach**. arXiv:2206.13982. Available at: https://arxiv.org/abs/2206.13982.

[11] Pikuliak, M., Srba, I., Moro, R., Hromadka, T., Smolen, T., Melisek, M., Vykopal, I., Simko, J., Podrouzek, J., & Bielikova, M. (2024). *Multilingual Previously Fact-Checked Claim Retrieval*. Kempelen Institute of Intelligent Technologies.

[12] Scikit-learn. (n.d.). Logistic Regression. Retrieved from https://scikit-learn.org/dev/modules/generated/sklearn.linear_model.LogisticRegression.html

[13] Shaar, S., Georgiev, N., Alam, F., Da San Martino, G., Mohamed, A., Nakov, P. (2024). *Assisting the Human Fact-Checkers - Detecting All Previously Fact-Checked Claims in a Document*. Cornell University, Sofia University, Qatar Computing Research Institute, HBKU, University of Padova, University of Wisconsin-Madison, Mohamed bin Zayed University of Artificial Intelligence.

[14] Sanh, V., et al. (2024). **Jina-ColBERT-v2: Advanced Version of ColBERT Model that Uses a Modified Version of XLM-RoBERTa as the Backbone**. arXiv:2408.16672. Available at: https://arxiv.org/abs/2408.16672.

[15] Wang, F., et al. (2024). **Distillation Techniques for Multilingual Information Retrieval**. arXiv:2405.00977v1. Available at: https://arxiv.org/html/2405.00977v1.

[16] Conneau, A., et al. (2020). **XLM-RoBERTa: A Strong Baseline for Cross-Lingual Language Understanding**. arXiv:1911.02116. Available at: https://arxiv.org/abs/1911.02116.

[17] Zenodo. (n.d.). Zenodo Record: 7737983. Retrieved from https://zenodo.org/records/7737983

[18] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2019). Unsupervised Cross-lingual Representation Learning at Scale. *arXiv*. Retrieved from https://arxiv.org/pdf/1911.02116

[19] PyTorch ReduceLROnPlateau https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

[20] PyTorch BCEWithLogitsLoss https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

[21] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, Yinhai Wang Bi-LSTM https://arxiv.org/abs/1801.02143

[22] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, Yinhai Wang Bi-LSTM https://arxiv.org/abs/1801.02143

9 Appendix

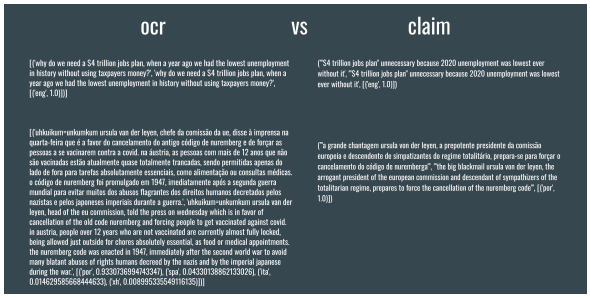
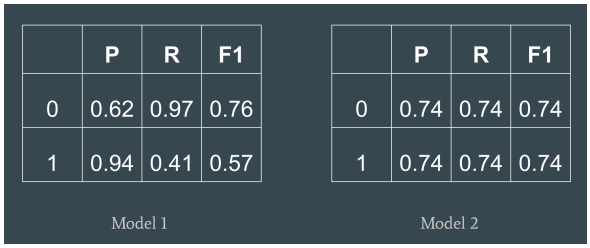


Figure 1: Example Data Instances



	P	R	F1
0	0.62	0.97	0.76
1	0.94	0.41	0.57

Model 1

	P	R	F1
0	0.74	0.74	0.74
1	0.74	0.74	0.74

Model 2

Figure 2: Classification Report Comparison

Post ID	Fact Check ID	OCR	Claim	Label	Features
2228	33	[('why do we need ...	('"$4 trillion jobs plan" ...	1	[('why do we need ...
2228	23568	[('why do we need ...	('america had the lowest ...	1	[('why do we need ...
2228	194577	[('why do we need ...	('"a year ago we had ...	1	[('why do we need ...
2229	33	[('why do we need ...	('"$4 trillion jobs plan"...	1	[('why do we need ...

Table 4: Sample Data Table