The Report on Optimization and Comparison of Retrieval and Classification Modules for Evidence-Based Claim Verification

Yu Zhang
1634674

Ximing Wan
1177855

HaiYang Huang
1071147

Abstract

This project implements a dual-module system for fact verification, comprising a retrieval module for extracting relevant evidence and a classification module for predicting claim labels. The objective is to identify the strengths and limitations of different retrieval and classification approaches in the context of automated fact verification. Our findings demonstrate that fine-tuning pre-trained models or implementing prompting techniques with Large Language Models (LLMs) yields significantly better results than training models from scratch. Our system, implementing a two-phase retrieval method combined with a quantized Gemma-2-9b-it model, achieved an evidence retrieval F-score of 0.17 and a classification accuracy of 0.57.

1 Introduction

This project aims to address the fact verification task through a dual-module system comprising retrieval and classification components. The retrieval module implements two distinct approaches: a baseline handcrafted BiLSTM sentence pair matching system that utilizes contextual embeddings and semantic matching strategies, and a more sophisticated two-phase system consisting of a retriever and a reranker. The classification module evaluates three models: a BERT-base-uncased classifier, a RoBERTa-base classifier, and an 8-bit quantized version of gemma-2b-9b-it utilizing in-context learning techniques. Our two-phase retrieval system is inspired by the work of (Ullrich et al., 2024), who reframed fact-checking as a Retrieval-Augmented Generation (RAG) task using Large Language Models. This paradigm, widely adopted in industry applications, has proven effective and adaptable to our fact checking task.

2 Task Definition and Dataset

This project focuses on developing an automated fact-checking system for climate science claims. The system performs two main tasks: first, retrieving relevant evidence passages from a provided knowledge source when given a claim, and second, classifying each claim into one of four categories (SUPPORTS, REFUTES, NOT_ENOUGH_INFO, or DISPUTED) based on the retrieved evidence. We work with four major datasets. The foundation is our knowledge base, containing approximately 1.2 million pieces of evidence, which is utilized across training, evaluation, and prediction phases. Our training set comprises 1,228 claim-evidence pairs. Each claim is associated with 1 to 5 pieces of evidence, with an average of 3.36 pieces per claim. The distribution of claim categories is shown in 1. For claims classified as NOT_ENOUGH_INFO, there are consistently 5 pieces of evidence provided. For development and unlabeled testing, we have 154 claims in each set. The development set follows a similar label distribution to the training set, as shown in 1. The average number of evidence pieces per claim in the development set is 3.19, indicating relatively consistent characteristics between source and target domains. The unlabeled test set contains only claims, allowing us to evaluate our system's performance through a leaderboard.

Metric	**Train**	**Dev**
Claims	1228	154
Label Distribution		
DISPUTED	124 (10.10%)	18 (11.69%)
REFUTES	199 (16.21%)	27 (17.53%)
SUPPORTS	519 (42.26%)	68 (44.16%)
NOT_ENOUGH_INFO	386 (31.43%)	41 (26.62%)
Evidence		
Avg pieces/claim	3.36	3.19
Min pieces	1	1
Max pieces	5	5

Table 1: Dataset statistics

3 Methods and Implementation Details

3.1 Retrieval Task

3.1.1 A Baseline BiLSTM Evidence Retrieval Model with GloVe Embeddings and Hard Negative Mining

The Bi-LSTM-based sentence pair matching system aims to retrieve semantically relevant evidence for a given claim. In the data preprocessing stage, we first tokenize, lemmatize, and remove stop words for the claim and evidence sentences, and then map each word to an index in the vocabulary. After processing, we construct training samples by pairing claim and evidence, including positive and negative samples. In order to improve the model's ability to identify semantic differences during the training phase, we introduce two strategies in the construction of negative samples: sampling 125 random negative examples from all non-relevant evidence, and 125 hard negative samples selected based on TF-IDF similarity(Robertson, 2004).

During the modeling process, we initialize the embedding layer with pre-trained 100-dimensional GloVe word vectors to enhance the semantic perception of the model(Pennington et al., 2014). The resulting word vector sequence is input into the bidirectional LSTM to capture the forward and reverse contextual dependencies in the sentence respectively. The output of the LSTM is then compressed into a fixed-length vector representation through the maximum pooling operation in the time dimension, which is used for claim and evidence respectively.

To model the semantic relationship between claim and evidence, we concatenate four vector features to form the final matching vector: the claim vector, the evidence vector, their element-wise absolute difference, and their element-wise product. The spliced vectors have a dimension of 1024 and are then fed into a two-layer feed-forward neural network containing a ReLU activation function and Dropout regularization, which outputs a scalar representing the matching score.

In the training phase, the model learns based on triples: each claim corresponds to one matched (positive example) evidence and one unmatched evidence. We use margin ranking loss as the loss function, the goal is to encourage the model to assign higher scores to positive examples and ensure that the score is at least a fixed margin value (set to 1.0 in this experiment) higher than the negative example.

3.2 Two-Phase Retrieval System

Our retrieval phase is divided into two steps: retrieving and reranking. We use two separate models for two key reasons. First, speed is crucial when dealing with our massive dataset of approximately 1.2 million rows of evidence. We need an efficient way to store vector representations (embeddings) of each piece of evidence in a vector database. Second, precision is essential. Since evidence embeddings are calculated independently of claims, they cannot capture the same level of interaction as a cross-encoder model. Our retrieval system begins with a sentence embedding model. We selected an open-source model with fewer than 1 billion parameters, specifically the top-performing model in the MTEB leaderboard (Enevoldsen et al., 2025) for retrieval: Snowflake/snowflake-arctic-embed-l-v2.0 (Inc., 2024). This model has 568M parameters with an embedding dimension of 1024 (citation needed). We evaluated several other models, including mixedbread-ai/mxbai-embed-large-v1, intfloat/e5-large-v2, and Alibaba-NLP/gte-large-en-v1.5, but none performed as well as our chosen model. We used this model as our base retriever and fine-tuned it with training data using sentence-transformers. For training, we employed MultipleNegativesRankingLoss, which is ideal for our case where we only have positive pairs (claim, evidence). This approach samples n-1 negative pair samples for contrastive learning. We set our batch size to 32 to balance the relationship between claims and related evidence against computational efficiency. We used an initial learning rate of 2e-5 and applied a linear learning rate decay. The warm-up ratio was set to 0.1 to stabilize training. The optimizer used is PyTorch's AdamW and trained for 5 epochs, keeping the checkpoint with the best result at the development set to prevent overfitting. After fine-tuning, we embedded all 1.2 million pieces of evidence and stored them in a FAISS vector database. For the cross-encoder model, we chose ms-marco-MiniLM-L6-v2 (22.7M parameters) (Reimers and Lab, 2021) for two main reasons. First, it's easily trainable and accessible through the cross-encoder module from sentence-transformers. Second, despite being lightweight, it outperforms similar models like BAAI/bge-reranker-large in our task. This superior performance likely stems from its training on the MS MARCO dataset, which contains real user queries from Bing search engine with annotated relevant text passages. Models trained on this dataset excel as rerankers for search systems, and fine-tuning this compact model helped us achieve higher performance in our target domain, especially given our limited data. We used CachedMultipleNegativesRankingLoss, which functions similarly to MultipleNegativesRankingLoss, and through hyperparameter tuning, we set the number of negative samples to 13 which minimizes the loss in development set. We used a similar training configurations as describe above.

3.3 Classification Task

3.3.1 BERT-base-uncased Classification Model

The BERT-base-uncased model, initially proposed by Devlin et al. (Devlin et al., 2019), is a pre-trained transformer model widely adopted for various natural language processing tasks, including text classification. The model consists of 12 transformer layers, each comprising self-attention mechanisms and feedforward networks. It is pre-trained on a large corpus of English text (BooksCorpus and Wikipedia) using two main tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

In this project, the BERT-base-uncased model is employed as the primary classification model to determine the relationship between a given claim and the retrieved evidence passages. The model is fine-tuned using a sequence classification task, where the input is a concatenation of the claim and evidence text, separated by the special [SEP] token. The [CLS] token at the beginning of the sequence is used to aggregate the semantic representation of the entire input, which is then passed to a fully connected layer to predict the classification label.

The input format for the classification model can be represented as follows:

$$\text{Input} = \text{[CLS]Claim text[SEP]Evidence text[SEP]} \quad (1)$$

During the training phase, the cross-entropy loss function is employed to optimize the model parameters. The cross-entropy loss is defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)\right] \quad (2)$$

where y_i represents the true label, and $\hat{y_i}$ denotes the predicted probability for each class.

The optimizer chosen for this model is AdamW, with a learning rate set to 2×10^{-5} and a batch size of 8. The training process spans 20 epochs.

3.3.2 Roberta-base Classification Model

The Roberta-base model, an enhanced version of BERT proposed by Liu et al. (Liu et al., 2019), leverages more extensive pre-training data and longer training durations to achieve improved performance in downstream NLP tasks. Unlike BERT, Roberta removes the Next Sentence Prediction (NSP) objective and focuses solely on Masked Language Modeling (MLM), allowing it to capture deeper contextual dependencies within the input text.

The Roberta-base model comprises 12 transformer layers, each with a hidden dimension of 768 and a total parameter count of 125 million. The input structure for the classification task is designed to accommodate the claim and evidence text as a single sequence, separated by the $< s >$ and $< /s >$ tokens. This structure is particularly beneficial in multi-sentence classification tasks, as it enables the model to effectively contextualize both segments within a single input representation.

The input format for the Roberta-base model is as follows:

$$\text{Input} =< s > \text{Claim text} < /s >< s > \text{Evidence text} < /s > \quad (3)$$

The optimizer configuration remains consistent with that of the BERT-base-uncased model, utilizing AdamW with a learning rate of 2×10^{-5} and a batch size of 8. The training duration is also set to 20 epochs.

3.3.3 Quantized Gemma-2-9b-it Model

For the classification component, we used lmarena.ai's Chatbot Arena (Chiang et al., 2024) to evaluate different open-source LLMs by testing them with sample query claims and evidences. Our evaluation showed that gemma-2b-9b-it performed remarkably well on sample queries. It demonstrated robust performance even when we deliberately removed relevant evidences and introduced noise, successfully identifying irrelevant evidences while maintaining accurate claim predictions. However, due to our hardware constraints (T4 GPU), running the full precision model was computationally infeasible. We addressed this limitation by using an 8-bit quantized version of gemma-2b-9b-it (Bartowski, 2024), which requires 10GB of VRAM.

To enhance performance, we implemented chain of thought reasoning and utilized three-shot learning with carefully selected examples: two refuting cases and one "not enough information" case. Using two refuting examples proved more effective than using one supporting and one refuting example, as it strengthened the model's ability to identify contradictions and prevented the model from overly predicting support, especially since more than 40% of the claim labels were supportive. We intentionally avoided using rigid and prescriptive instructions such as "analyze each individual piece of evidence and classify it as SUPPORTING, REFUTING, or NOT_RELEVANT" or quantitative rules like "If a clear majority (more than 60%) refutes the claim, label it as REFUTES" or "Label as NOT_ENOUGH_INFO if there are fewer than 3 relevant evidences." This decision was based on two key observations: First, while explicit instructions improved the model's adherence to guidelines, they made it overly meticulous in analyzing each piece of evidence for support or refutation. This approach proved counterproductive given that only 30% of retrieved evidences were actually relevant. Second, we found that numerical thresholds were problematic because the model sometimes miscounted evidence verdicts, especially with larger evidence sets. The model also showed inconsistent sensitivity to numerical criteria, often ignoring specific count-based instructions. Instead, we allowed the model to weigh evidence organically and develop an overall assessment of the relationship between the query claim and the evidence set, rather than focusing on individual claims and counts. We set the temperature to 0.3 to promote analytical rather than creative thinking. As a safeguard, if we cannot identify the label in the initial response, we prompt the model again with a specific format request "Label: [LABEL]" at a temperature of 0.1 to ensure consistency in the final decision.

4 Results & Discussion

4.1 Baseline BiLSTM Evidence Retrieval Model

During the model development process, we continued to optimize the structure and training strategy of the retrieval model, and the F-score performance at the evidence level gradually improved. Starting from the initial simple matching model based on BiLSTM, we successively introduced pre-trained GloVe word vectors, margin ranking loss, and triple training combined with TF-IDF similarity, which effectively enhanced the model's ability to distinguish semantically relevant and irrelevant evidence. With the continuous accumulation of these improvements, the model's F-score increased from close to zero (about 0.001) to about 0.04, reflecting a steady performance growth.

However, despite substantial progress, the overall retrieval effect is still not ideal. The model may have a certain degree of underfitting in semantic representation learning, especially when faced with a large evidence base with significant semantic noise, it is difficult to accurately identify fine-grained semantic differences. A key factor causing this problem is the limitations of our current task modeling method and training data structure. Our system treats claim-evidence matching as a binary pairing problem, rather than a functional response to the claim based on the semantics of the evidence. The training samples are constructed into positive and negative pairs only based on whether the evidence ID appears in the annotated set, without considering the semantic implications or position judgments. This design makes the model more inclined to learn surface text similarity rather than semantic associations at the reasoning level, which in turn affects its ability to model the logic of evidence support in complex language structures.

In addition, the expressive power of static word vectors and the BiLSTM architecture itself is also limited, making it difficult to capture polysemy in the context, long-distance dependencies, or complete multi-hop semantic reasoning. In addition, the evaluation mechanism is highly strict, only accepting results that are completely consistent with the annotated evidence ID, so that some semantically reasonable but slightly deviated predictions are also considered errors, further suppressing the model's retrieval performance.

In summary, although the current modeling method based on pairing and sorting provides a simple starting point for the task, if a more breakthrough improvement is to be achieved, it may be necessary to introduce a more expressive pre-trained language model (such as transformer architectures such as BERT) and combine it with a richer semantic modeling strategy to more fully explore and express the deep semantic relationships and reasoning paths required in fact-checking tasks.

4.2 Two-Phase Retrieval System

Based on our observation that claims have a maximum of 5 associated pieces of evidence, we designed our two-phase model as follows: In phase one, the retriever model extracts the top K relevant pieces of evidence. In phase two, we rerank these K pieces using a reranker model and select the top 5 based on their cosine similarity scores. Additionally, we implement a score_gap_threshold that calculate the score distance between the an evidence reranked score and the highest reranked score for a claim, then divided by the highest reranked score. If this gap is higher than the threshold, it should be excluded from the retrieved evidence set. We optimized both the number of retrieved evidence pieces (K) in phase one and the score_gap_threshold through grid search. The optimal set of k and score_gap_threshold are found to be 10 and 0.18 on development set, which we applied in the test set prediction. We compared the performance of three retrieval system configurations: Our baseline BiLSTM model, the retriever model without reranking and the complete two-phase system with reranking. The results, shown in 2, demonstrate that our two-phase model significantly outperforms the baseline. While implementing the reranker resulted in a 5% decrease in recall, it dramatically improved precision by 70%. Overall, the retrieval F-score increased by 27% with the complete two-phase system.

Metric	Baseline	w/o Rerank	Rerank	Imp(%)
F-score	0.037	0.219	0.279	+27.2
Precision	–	0.182	0.310	+70.5
Recall	–	0.334	0.316	-5.4
Claims	154	154	154	–

Table 2: Evidence retrieval performance comparison

4.3 Comparison between Classification Models

We evaluated our classification models under two experimental conditions: (1) providing models with all correct evidences as input, and (2) using only the evidence retrieved by our retrieval system. As shown in Table 3, when given the complete set of correct evidences, the fine-tuned BERT-base achieved the highest accuracy, while the quantized Gemma-2-9b performed lowest at 0.558. The fine-tuned RoBERTa demonstrated comparable performance to BERT-base at 0.61, indicating successful knowledge transfer through fine-tuning. Despite its significantly larger parameter count, Gemma-2-9b-it did not show exceptional performance on our specific task. However, when working with noisier evidence sets (where not all evidences were relevant to the claim), we observed different patterns. RoBERTa maintained the best performance with only a 6% accuracy drop compared to the perfect evidence scenario. Gemma-2 showed even greater robustness, with just a 1% drop in accuracy. In contrast, BERT-base experienced a substantial 36.8% decrease in accuracy. As detailed in Table 4, while BERT-base achieved 90% recall for the "not enough information" class, it struggled to identify disputed claims in noisy settings. This challenge with disputed claims was consistent across all models, with both Gemma-2 and fine-tuned RoBERTa showing their lowest accuracy on this class compared to the other three categories. The fine-tuned RoBERTa-base model demonstrated the best overall performance, particularly in identifying supporting claims. Gemma-2-9b-it showed superior performance in identifying refuting claims, possibly due to our few-shot learning approach using two refuting examples. BERT-base excelled at identifying "not enough info" cases with 90% recall, suggesting effective label determination capabilities. However, its performance may have been limited by our retrieval system's inability to provide sufficient evidence for accurate classification.

Model	**F=1**	**F=0.2788**	**Drop %**
BERT-base	0.668	0.422	36.8%
Roberta	0.610	0.571	6.3%
Gemma-2-9b-it	0.558	0.552	1.1%

Table 3: Model performance with retrieval quality degradation

Model	Label	Prec.	Rec.	Acc.
gemma-2-9b-it	DISPUTED	.18	.28	.12
	NOT_ENOUGH_INFO	.75	.29	.27
	REFUTES	.64	.52	.40
	SUPPORTS	.61	.79	.53
bert-base	DISPUTED	.00	.00	.00
	NOT_ENOUGH_INFO	.34	.90	.33
	REFUTES	.43	.11	.10
	SUPPORTS	.66	.37	.31
roberta-base	DISPUTED	.30	.33	.19
	NOT_ENOUGH_INFO	.56	.34	.27
	REFUTES	.57	.44	.33
	SUPPORTS	.64	.82	.56

Table 4: Model's precision, recall and accuracy of each label class

5 Team Contribution

Haiyang Huang - two-phase retrieval system, llm in-context learning.
Ximing Wan - Build a BiLSTM-based semantic matching model, as well as preprocess and analyze claim and evidence texts.
Yu Zhang - BERT-base-uncased classification model and Roberta-base classification model.

References

Bartowski. 2024. gemma-2-9b-it-gguf. https://huggingface.co/bartowski/gemma-2-9b-it-GGUF.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *Preprint*, arXiv:2403.04132.

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 2019*, pages 4171–4186. Association for Computational Linguistics.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, and 67 others. 2025. Mmteb: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*.

Snowflake Inc. 2024. Snowflake arctic embed l v2.0. https://huggingface.co/Snowflake/snowflake-arctic-embed-l-v2.0.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers and UKP Lab. 2021. Cross-encoder ms-marco-minilm-l-6-v2. https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2.

Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520.

Herbert Ullrich, Tomáš Mlynář, and Jan Drchal. 2024. Aic ctu system at averitec: Re-framing automated fact-checking as a simple rag task. *Preprint*, arXiv:2410.11446.