COMP90042 Project Report

ID: 1305712

1 Introduction

Climate change is one of the most significant concerns at present. With the popularity of social media, misinformation regarding climate change spreads incredibly fast, resulting in negative impacts on the public's understanding of the nature and causes of climate change. This issue has brought attention to developing automated fact-checking systems to counter misinformation on climate change, which are typically built upon Natural Language Processing (NLP) models (Zeng et al., 2021). However, building an accurate fact-checking system is complex, as it requires proper information retrieval and reliability measurement of a given claim, which a single model can hardly achieve (Rashkin et al., 2017; Thorne et al., 2018). This underscores the need to design a pipeline to accomplish the two goals effectively.

This report demonstrates a pipeline for a climate change fact-checking system. The pipeline is built to accomplish the following task: given a claim related to climate change, search for relevant evidence from an extensive knowledge base, and classify the status of the claim as one of {`SUPPORTS`, `REFUTES`, `NOT_ENOUGH_INFO`, `DISPUTED`} depending on the relevant evidence. Based on the task, the pipeline can be modelled in three stages: evidence rough-filtering, evidence precise-filtering, and claim verification. Given a claim, initially, the evidence rough-filtering aims to obtain potentially relevant evidence N from a collection of ground truth M, usually $|N| \ll |M|$. Then, the process of evidence precise-filtering is deployed to predict truly relevant evidence $\mathcal{T}$ from N. Finally, the claim verification is to reach a verdict on the claim using $\mathcal{T}$ (Zeng et al., 2021).

This report aims to answer the following questions: **1)** What are the effective models for evidence retrieval and claim verification? **2)** What techniques can be utilized to enhance the performance of those models? In support of this aim, this report starts by discussing the capabilities and limitations of related work, followed by an overview of design decisions for constructing a fact-checking pipeline. Then it demonstrates experimental results and a detailed analysis of various models' performance in addressing the challenge.

2 Literature Review

Many researchers have dedicated efforts to building automated fact-checking systems (Nogueira et al., 2020; Zhuang and Zhang, 2022). The previous state-of-the-art fact-checking system against COVID-19 claims, namely VERT5ERINI (Pradeep et al., 2020), exploits an advanced version of BM25 ranking function (Yang et al., 2017) and T5 (Raffel et al., 2020), a sequence-to-sequence transformer architecture, for abstract retrieval and label prediction. Their experimental results indicate that VERT5ERINI outperforms its benchmark model by an F1 score of 4% in evidence retrieval and 12% in claim verification. Moreover, in the research conducted by Shaar et al. (2020), BM25 (Robertson et al., 2009) and BERT (Devlin et al., 2018) are utilized in conjunction to achieve information retrieval and document re-ranking, which yields a performance increase of 5% in mean average precision compared with its benchmark model.

However, despite their high performance, the dependency on BM25, a light-weighted ranking function primarily used for searching relevant information from an extensive knowledge source, might lead to non-optimal results. Because the BM25 score is calculated based on keyword matching, it struggles to capture complicated context hidden in a language sequence (Karpukhin et al., 2020). In contrast, deep learning models may yield better performance, but their running time is slower in searching through extensive evidence collection. Thus, how to design an effective and efficient model for evidence retrieval remains an open question.

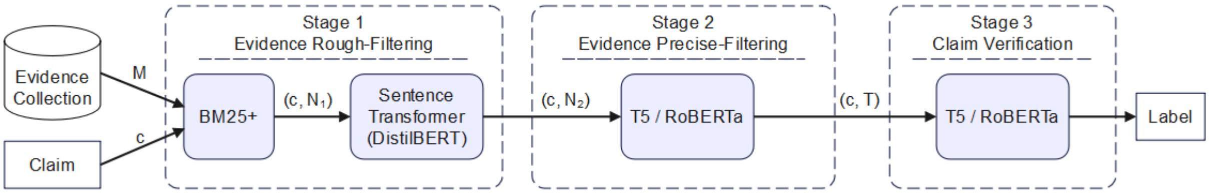

Figure 1: Pipeline for Automated Fact-checking System

3 Method

3.1 Pipeline Design

Figure 1 illustrates the pipeline to tackle the aforementioned challenge. At each stage, the task is formulated as a classification problem.

3.1.1 Stage One

Given a claim c and an evidence collection M, the objective of stage one is to output N, where $N \subseteq M$, such that N contains as much relevant evidence to c as possible, conditioned on $|N| \ll |M|$.

The input c is employed as a query by a BM25+ function, then BM25+ searches through M and calculates a similarity score for c against each piece of evidence. The output N_1 consists of the top 100 pieces of evidence with the highest similarity scores. BM25+ (Lv and Zhai, 2011) is a TF-IDF based measurement with an additional feature: it does not heavily penalize long documents when calculating similarity scores, which may be beneficial for searching relevant evidence as the length of relevant evidence may vary.

However, as BM25+ might struggle with understanding the context of a language sequence, an improved approach could be utilizing sentence embedding to calculate similarity scores (Karpukhin et al., 2020). Sentence Transformer (Reimers and Gurevych, 2019), an architecture with siamese and triplet network structures mainly designed for generating sentence embeddings, is a promising framework for precisely measuring the similarity between two embeddings. While efficiency is a concern in stage one, DistilBERT (Sanh et al., 2019), a pre-trained model with knowledge distillation, is an excellent candidate for this task as it is shown to maintain 97% language understanding while being 60% faster than its predecessor BERT (Devlin et al., 2018). Thus, combining Sentence Transformer with distilBERT's pre-trained weights(base, uncased) can leverage benefits from both, meaning that the model can yield better performance while being computationally efficient. Moreover, since DistilBERT's pre-training includes cosine distance loss as one of its objective functions, it is anticipated to perform better when used in a similar context. Hence, it would be reasonable to employ cosine similarity for measuring the relevance of two embeddings and cosine embedding loss for optimization during training. The combined model (see footnote[1] for hyperparameters) takes N_1 as better training data than random selection. Once the training phase is completed, the model generates and stores a collection of embeddings $E(M)$ for each instance in M. When a claim c arrives, the model encodes c into its embedding $E(c)$ and outputs the top 100 similar pieces of evidence (being classified as potentially relevant), denoted as N_2, from $E(M)$ based on cosine similarity.

3.1.2 Stage Two

Given a claim c and the output N_2, stage two aims to predict truly relevant evidence $\mathcal{T} \subseteq N_2$.

T5 (Raffel et al., 2020), a sequence-to-sequence pre-trained model, has demonstrated its versatility in various NLP tasks, including machine translation, summarization, and text classification. A sequence-to-sequence model transforms an input sequence from one domain into an output sequence in another. The rationale for deploying T5 (base, see footnote[2] for hyperparameters) in stage two is that by incorporating meaningful prompt tokens into the input sequence, the understanding of language and context that T5 gained from its pre-training process is exploited (Nogueira et al., 2020). Concretely, given a claim c and a piece of evidence $e \in N_2$, the input sequence is a concatenated string:

Query: c Document: e Relevant:

The output sequence is a single token that is either 'True' or 'False', indicating whether e is relevant to c.

An alternative model, RoBERTa (Liu et al., 2019), is an enhanced variant of BERT (Devlin

[1] Hyperparameters: num_epoch = 2; learning rate = 1e-5; batch_size = 8.

[2] Hyperparameters: num_epoch = 6; learning rate = 2e-5; batch_size = 1; class_weight = balanced.

et al., 2018) that has been modified with different hyperparameters and slightly altered pretraining objectives. These adjustments have allowed RoBERTa to achieve state-of-the-art performance on the General Language Understanding Evaluation benchmark, making RoBERTa (base, see footnote[3] for hyperparameters) a promising candidate for the task in stage two. In comparison to T5, the preprocessing needed for RoBERTa is more straightforward. Given c and e, the input sequence is simply:

<s> c </s> <s> e </s>

where <s> and </s> denote the start and the end of the sentence, respectively. The output is a binary value where 0 represents 'irrelevant', and 1 represents 'relevant'.

T5 and RoBERTa are independently fine-tuned in stage two. The output of each model is $\mathcal{T}$, in which each element is classified as truly relevant.

3.1.3 Stage Three

Given a claim c and the output $\mathcal{T}$, the objective of stage three is to classify the status of c as a label l using $\mathcal{T}$, where $l \in \{\texttt{SUPPORTS}, \texttt{REFUTES}, \texttt{NOT\_ENOUGH\_INFO}, \texttt{DISPUTED}\}$.

Both T5 (base, see footnote[4] for hyperparameters) and RoBERTa (base, see footnote[5] for hyperparameters) continue to contribute in this stage, leveraging their aforementioned capabilities. However, the expected input sequence and output differ. For T5, the input sequence is constructed as:

hypothesis: c sentence1: e_1 sentence2: $e_2 \dots$

where $e_1, e_2, \dots \in \mathcal{T}$. The output sequence is a single token l. For RoBERTa, the input sequence is:

<s> c </s> <s> e_1 </s> <s> e_2 </s> $\dots$

where $e_1, e_2, \dots \in \mathcal{T}$. The output is a value ranging from 0 to 3, each mapping to a corresponding label.

3.2 Evaluation Method and Metrics

The hold-out method is employed for evaluation due to its computational efficiency at the cost of potentially less reliable evaluation results.

For stage one, recall is regarded as the most crucial evaluation metric as the aim is to select as much relevant evidence as possible. For stage two, the F1 score of relevant evidence retrieval is employed for evaluation as both precision and recall are vital. The accuracy score is utilized for evaluating stage three since it is interpretable and intuitive.

3.3 Hyperparameter Tuning

Hyperparameters are tuned for optimal performance. The same sets of hyperparameter settings are experimented with for all models, including learning rate = {1e-4, 2e-5, 1e-5}, batch_size = {1, 2, 4, 8, 16}. All models are fine-tuned for up to 15 epochs. Validation is executed for each epoch. Hence, the number of epochs is chosen based on evaluation results, training, and validation loss.

3.4 Baseline

The performance of the baseline model is provided by the COMP90042 Teaching Team.

4 Results and Discussion

4.1 Stage One

Table 1 illustrates the evaluation results in stage one. As anticipated, BM25+ exhibits relatively poor performance. However, its effectiveness can be significantly improved using preprocessing techniques such as stop-word removal and stemming, achieved via the NLTK[6] library.

On the other hand, employing an embedding-based similarity measurement, such as a fine-tuned embedding encoder utilizing DistilBERT, yields comparable results. The integration of the Sentence Transformer architecture further enhances its performance, resulting in an approximate 10% improvement. Despite this promising enhancement, the overall results are still not fully satisfactory.

A technique similar to self-training further increased the model's performance. Initially, the training data is provided by the preprocessed BM25+. Following the completion of the current epoch's training, the model, with its updated parameters, selects the top 100 similar pieces of evidence from M for training in the subsequent epoch. The underlying intuition of this technique is that the model can enhance its performance by learning from its previous mistakes. However, the downside of this technique is that it may make the model prone to overfitting. As shown in Table 1, the performance starts to decrease monotonously after epoch 2. Hence early stopping is utilized. This

[3] Hyperparameters: num_epoch = 5; learning rate = 1e-5; batch_size = 16; class_weight = balanced.

[4] Hyperparameters: num_epoch = 4; learning rate = 1e-5; batch_size = 1; class_weight = balanced.

[5] Hyperparameters: num_epoch = 6; learning rate = 1e-5; batch_size = 8; class_weight = balanced.

[6] https://www.nltk.org/

Model	Recall (Top100)
BM25+	0.27
|—Preprocessing	0.43
DistilBERT	0.42
|—SentenceTransformer	0.53
|—Tweak (epoch 2)	**0.62**
|—Tweak (epoch 3)	0.61
|—Tweak (epoch 4)	0.57

Table 1: Model Performance in Stage One

Model	Evidence Retrieval F1
Baseline	0.07
T5	0.16
RoBERTa	0.19
|— Top 3	0.20
|— Top 4	**0.21**
|— Top 5	0.20

Table 2: Model Performance in Stage Two

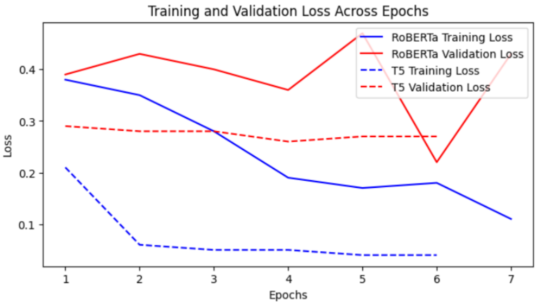

Figure 2: Learning Curve in Stage Three

approach is shown to be beneficial, resulting in an approximate 20% performance increase compared to the BM25-based model.

4.2 Stage Two

Table 2 presents the evaluation results in stage two. The baseline model demonstrates a relatively low performance, highlighting the complexity of this task and suggesting the need for more sophisticated models.

The T5 model offers a performance increase of approximately 10% over the baseline model. However, it was observed that T5 struggles significantly with the imbalance of class distribution in the training data. An attempt to alleviate this issue utilizing a weighted loss was made with limited success. This limitation is partly due to the fact that T5's loss value is calculated based on the entire output sequence rather than a single target token, such as 'True' or 'False'. T5 occasionally generates outputs other than the target tokens, which leads to sub-optimal performance.

In contrast, RoBERTa yields the best performance. Despite the imbalanced class distribution in the training data, applying weighted loss to RoBERTa is simpler than T5, as RoBERTa's output consistently focuses on the class label. However, RoBERTa occasionally over-predicts the number of relevant evidence, which can be addressed by selecting the top-k evidence based on relevance probability. As indicated in Table 2, RoBERTa yields the best performance when k = 4.

4.3 Stage Three

Table 3 shows the evaluation results in stage three. The baseline model obtains an accuracy score of 0.38, indicating that the task in stage three is easier to accomplish. T5 performs better in stage three due to a less severe class imbalance. However, the learning curve shown in Figure 2 suggests that T5 may be overfitting as the training loss remains low across all epochs. Despite T5's slightly better accuracy, RoBERTa, with a more reasonable learning curve, is chosen to avoid potential overfitting.

Model	Classification Accuracy
Baseline	0.38
T5	0.55
RoBERTa	**0.53**

Table 3: Model Performance in Stage Three

5 Conclusion and Future Directions

This report proposed a complete pipeline for an automated fact-checking system. It demonstrates the performance of various models in evidence retrieval and claim verification, along with handy techniques to further improve their capabilities in addressing the challenge. The final version of the pipeline, consisting of a Sentence Transformer with DistilBERT's pre-trained weights and RoBERTa, yields the best performance. However, the final evaluation on Codalab indicates that the system obtains an F1 score of 0.14 and an accuracy score of 0.45, suggesting the system may be over-tuned.

Future work is to explore techniques that can be utilized to avoid overfitting when constructing an automated fact-checking system.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

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*.

Yuanhua Lv and ChengXiang Zhai. 2011. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 7–16.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2020. Scientific claim verification with vert5erini. *Corpus*, 1(a2):a3.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937.

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.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 1253–1256.

Xia Zeng, Amani S Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438.

Yan Zhuang and Yanru Zhang. 2022. Yet at factify 2022: Unimodal and bimodal roberta-based models for fact checking. In *Proceedings of De-Factify: Workshop on Multimodal Fact Checking and Hate Speech Detection, CEUR*.