NLP Report

1320186

1 Introduction

Climate change is one of the most pressing issues affecting humanity. Unfortunately, the increase in unsubstantiated claims related to climate science has resulted in a distortion of public opinion. Hence, developing an automated fact-checking system is critical to classify such statements accurately. Our project aims to create a fact-checking system that encompasses both contextual understanding and text classification. We will leverage the Roberta/BERT model to build DPR models that can comprehend contextual information to achieve this goal. Then, we will employ the Roberta/BERT model to perform text classification. Ultimately, our efforts will facilitate evidence-based discussions and improve public understanding of climate change. This report will analyze the impact of the DPR approach on the accuracy and efficiency of the fact-checking process and compare the specific performance of the BERT and RoBERTa models in addressing this issue.

2 Relative Work

2.1 Dataset

Our task is to give each claim one of the four labels (SUPPORTS, REFUTES, NOT ENOUGH INFO, DISPUTED) and find the evidence that supports this label. Datasets involve train(1,228) and dev(154) datasets, each whose data is combined by claim text, claim label, and evidence. Additionally, we must combine the evidence dataset(1,208,827) by evidence id and evidence text. Although most text lengths are below 200(around 11,000), some highly long texts can be removed to reduce noise. Once the datasets are combined, we aim to locate the relevant evidence texts for a given claim text and then predict the corresponding labels.

2.2 DPR

DPR (Deep Pre-training with Reading Comprehension) model combines the strengths of retrieval-based and generation-based models. DPR pre-trains a dual encoder architecture on a large text corpus and fine-tunes the model for downstream tasks such as question-answering and information retrieval. Compared to traditional passage retrieval methods using sparse vector space models like TF-IDF and BM25, DPR can better capture semantic relationships and improve retrieval results(Karpukhin et al., 2020).

DPR aims to map input channels to a d-dimensional vector using multiple encoders and select the k channels that are most similar to the problem. The use of negative sampling during training increases the number of training instances. DPR's main advantage is its ability to handle long documents and retrieve relevant evidence passages from knowledge sources. This makes it suitable for fact-checking tasks such as those in the project. Section 3 will provide more details about the DPR model and how we have reproduced and improved it.

2.3 BERT

BERT (Bidirectional Encoder Representation from Transformers) is a pre-trained language model that has achieved more advanced performance in many natural language processing tasks. BERT can generate rich textual representations by understanding the context of words in sentences without requiring substantial modifications to the task-specific architecture. It is versatile and can handle various natural language understanding tasks, including text classification and question answering. By pre-training and fine-tuning BERT on DPR models, we can enhance its performance and make it suitable for our fact-checking system. Specifically, in the pre-training phase, BERT is trained by "Masked LM" and "Next Sentence Prediction

(NSP)" tasks(Devlin et al., 2018). In the fine-tuning phase, the pre-trained BERT model's parameters are fine-tuned for specific downstream tasks, such as DPR, allowing it to handle a wide range of natural language understanding tasks.

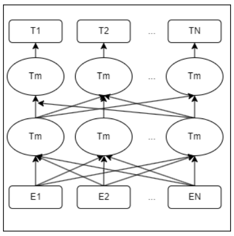

Figure 1: BERT-Encoder

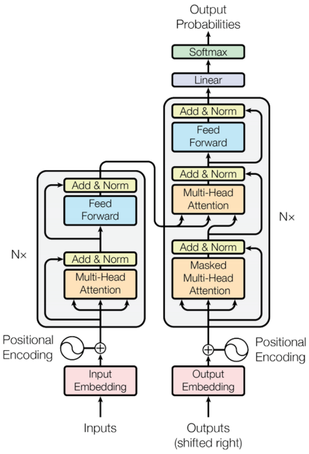

Figure 2: Transformer Model

The main model structure of BERT transformer (shown in Figure 2), where the cell structure of each layer in the encoder of BERT is shown in Figure 1.

2.4 RoBERTa

As an extension of BERT, the Roberta (Robust Optimization BERT method) model can improve language knowledge learning from training data and the performance of downstream tasks, Roberta uses a more extensive training corpus, dynamic masking, a more robust optimization algorithm, and byte-level BPEs, improving BERT's pre-training strategy(Liu et al., 2019). Roberta achieves dynamic masking by randomly changing the masking token in the training data during each iteration. Roberta also improves its performance on downstream tasks such as GLUE, SQuAD, and RACE using larger batch sizes and more training data and steps.

These enhancements make RoBERTa suitable for tasks requiring high levels of language understanding, such as fact-checking systems, where it can improve classification accuracy and efficiency. Using Roberta, DPR can better understand question contexts and retrieve more relevant answers.

3 Build System and Results

3.1 Overview

Our system consists of two parts. Task 1 constructs two independent BERT/Roberta-based models based on the structure of the DPR model in Figure 4, processing the claims and their associated evidence texts separately to find similarities between contexts. Task 2 will modify the BERT/Roberta-base model and add some layers and activation functions to achieve the classification effect. For the hyperparameters, we set the maximum length of the input text, the name of the model referenced in the huggingface open source community, the batch size, and the number of epochs. The learning rate and other parameters that may affect the training effect of the model are continuously adjusted according to the specific training situation, so they are not used as hyperparameters.

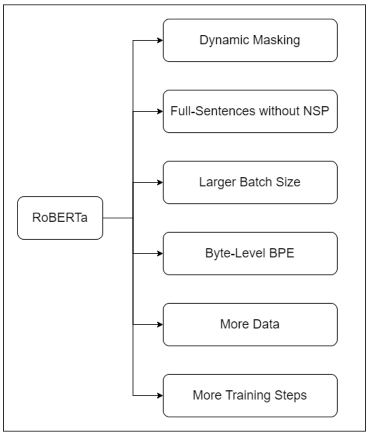

Figure 3: The figure shows the improvements of RoBERTa to the BERT model

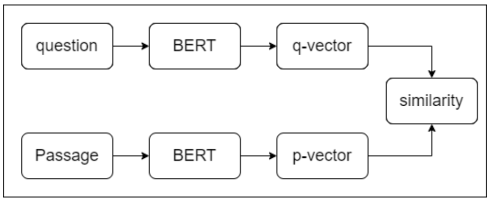

Figure 4: The diagram shows the model of DPR

3.2 Context Understanding

We attempt to replicate the data preprocessing method used in the DPR paper, which combines Gold negative samples and BM25 negative samples (according to the experimental results in the paper, this combined sampling approach leads to better predictions)(Karpukhin et al., 2020). The Gold method can be described as having B positive samples within a batch of size B. To avoid introducing excessive negative samples, which would result in a large amount of encoding computation and extended calculation time, we use positive samples from outside the current batch as negative samples for the current batch(Karpukhin et al., 2020).

$$S = Q * P^T, P = (P_1 + N_2 + .. + N_B) \quad (1)$$

The BM25 sampling method has been replaced with one that uses cosine similarity to find negative samples. The specific sampling process is as follows: First, find all the evidence used for labeling claims in the training and validation datasets, organize them, and output them as a new evidence dataset (about 3,000 pieces). They use Roberta to tokenize and encode the claims and evidence in the training and validation datasets separately, performing dot product operations on their results. Next, use the Tanh function to normalize the tensor, select data with higher relevance (by finding the corresponding evidence index through the tensor), and filter out evidence more relevant to the claims (about 4,000 pieces of data).

$$Similarity(A, B) = \frac{(A * B)}{||A|| * ||B||} \quad (2)$$

Finally, use the same steps to search for similarities between the original evidence dataset and the filtered associated evidence dataset, obtaining the evidence texts (about 30,000 pieces of data) as the new evidence dataset for encoding with the training dataset. As before, use the cosine similarity function to compare the actual texts in the training dataset and each predicted potentially relevant text for similarity. Select the group with the highest similarity as the negative samples to be stored in each data set in the training dataset.

By the experiments in the DPR paper, we use the dot product of claim embeddings and evidence embeddings (with the embedding output being the last hidden state of size batch-size * 768). We combine tanh normalization and softmax activation functions to create the loss function, compute the loss for the current iteration, and then apply loss.backward() for the next round of computation. The loss function formula is as follows:

$$L(q_i, p_i^+, p_i^- .. p_i^-) = -log \frac{e^s im(q_i, p_i^+)}{e^s im(q_i, p_i^+) + \sum_{j=1}^{n} e^s im(q_i, p_j^-)} \quad (3)$$

3.3 Text Classification

To infer the corresponding label for each claim among (SUPPORTS, REFUTES, NOT ENOUGH INFO, DISPUTED) using the existing positive and negative samples, we must modify the BERT and Roberta base models, which output the last hidden state (batch size * sequence length * 768). We add layers and activation functions to transform the output, extracting the first position of each sequence to form a new tensor (batch size * 768) and using a fully connected layer (768, 4) to map the 4-dimensional feature representation to the sample label space, yielding a tensor of size (batch size, 4).

We introduce the cross-entropy loss function to minimize the loss during model training. The smaller the loss, the better the model. When updating parameters using gradient descent, the model's learning speed depends on the learning rate and the partial derivative value. The larger the model's error, the worse its performance, resulting in a more considerable partial derivative value and a faster learning speed. Its formula is as follows:

$$L = -\sum_{i=1}^{n} t_i * log(p_i) \quad (4)$$

By using the logistic function to obtain probabilities and combining it with cross-entropy as the loss function, we achieve a faster learning speed when the model's performance is poor and a slower learning speed when the model's performance is good. This approach optimizes model training, balancing speed and performance.

4 Discussion

4.1 Result

Figure 5 compares the loss function values and accuracy of the Roberta and Bert models regarding context understanding. The loss patterns of the two models are almost the same. Roberta has slight fluctuations in the first few rounds as it has not been fully trained, while Bert's loss continuously decreases. In the final few epochs, both models tend to converge to the same loss value. However, there are significant fluctuations in the accuracy of the two models. After reaching an initial peak of 0.025, they both experience a substantial decline, with Bert's performance being even better in some instances, far surpassing Roberta's during the middle rounds. Roberta performs better than Bert in text classification tasks, as its loss consistently

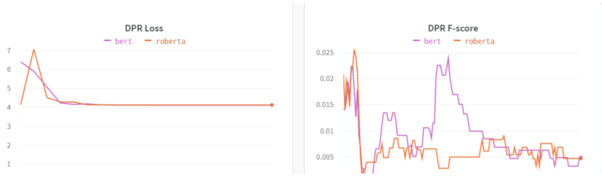

Figure 5: Context Understanding Result

decreases in Figure 6. However, during the middle rounds (4-6), Bert's loss value remains stable and experiences a slight increase. The accuracy of the two models is nearly indistinguishable, but the Roberta model exhibits more significant fluctuations in its accuracy.

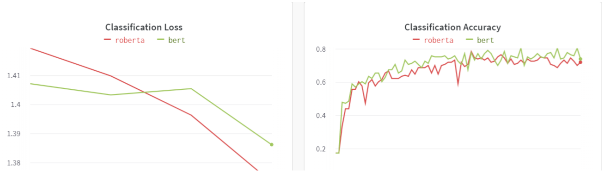

Figure 6: Text Classification Result

The prediction evaluation results on CodeLab are not satisfactory, the Harmonic Mean score is only about 0.02, so I will discuss the current defects from both the design of the model and the design of the experiment, and analyze what impact these defects will have on the experiment.

4.2 Model Analysis

Based on the description of the results, it is evident that Bert's overall performance is better than Roberta's. We have made the following conjectures regarding the possible reasons for this:

1. Roberta uses a larger batch size during pretraining, which helps reduce the Perplexity of the retained training data and improve downstream metrics. However, we controlled the batch size between 2-16 during training (to prevent excessive GPU memory usage leading to interrupted training). This might have prevented Roberta from fully utilizing the additional data, leading to suboptimal learning of the data distribution and poor convergence.

2. Compared to Bert, Roberta employs a dynamic masking strategy. For each training instance, it masks input according to different masking patterns(Liu et al., 2019). This means that Roberta requires more training iterations to learn better representations. With a small batch size, the model may need more training instances to exploit this strategy fully. Additionally, our dataset is relatively small, which may need to provide more training time for the model, resulting in continuous fluctuations in accuracy observed in the graph.

3. Hyperparameter selection: At different batch sizes, hyperparameters (such as learning rate, optimizer, and weight decay) may need to be adjusted continuously. With a small batch size, our chosen learning rate might need to be bigger, leading to poor model learning performance.

4.3 Experiment Analysis

There may be some issues in the experimental design and model selection that could have an impact on the experimental results:

1. Quality and scale of the dataset: The performance of the DPR model might be affected due to the large size of the evidence dataset and the lack of representativeness of much of the data. As a result, the model may be unable to learn sufficient information to provide optimal performance across various tasks.

2. Long texts and fragmented information: Our system limits the maximum length of input data to prevent out-of-memory errors, as the average length of the data exceeds 200 characters. Although DPR can handle long paragraphs, the model may need to catch critical information in lengthy texts during retrieval. Additionally, the DPR model may need help handling fragmented information spread across multiple sources. For example, in our problem, we identify potentially related evidence texts based on the claim and then assign a label to the data. Thus, the DPR model might need to locate evidence information that supports or opposes the claim accurately.

5 Conclusion

This article discusses using BERT and RoBERTa models in conjunction with DPR models to solve climate change-related automated fact-checking problems. By using DPR models with a deep reading understanding based on BERT and RoBERTa, the system can capture semantic relationships and thus find relevant evidence more accurately. Experimental results show that RoBERTa outperforms BERT on text classification tasks, but BERT performs better in context understanding. The article analyzes the performance of these two models in handling fact-checking tasks through extensive experiments. It suggests some factors that may affect the performance of the models, such as the dataset's quality and size and the input data's length. Nevertheless, it helps to promote evidence-based discussions and improve public understanding of climate change.

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*.