COMP90042 Project 2023: Automated Fact Checking For Climate Science Claims

1259524

Abstract

The BERT family is popular in the Natural Language Processing(NLP) field. In order to explore and enhance the understanding of BERT, an automated fact-checking system for climate claims was introduced. The system can validate a claim by identifying the relevant evidence and classifying the claim by labels. In this project, the performance of the BERT base model and RoBERTa base model will be analyzed and compared under limited hardware resources, and the best-performance model will be integrated into the system.

1 Introduction

Fact-checking is an essential demand in the world of information. Misinformation leads to a distortion of public opinion. The climate has deteriorated in the last decades which is one of the most important concerns to humanity. To help people to build a correct cognition of climate knowledge, an automated fact-checking for climate information system is introduced in this paper. The system process the claims in 2 steps. First, use TF-IDF to retrieve top-k relevant evidence. Second, use the BERT/RoBERTa model to classify the claim by labels. Finally, an output file is generated by the system that can be used to measure the performance. The performance had been examined under different parameters to fine-tune hyperparameters. However, due to the restriction of hardware performance, the experiment covered a narrow range of parameters, balancing the trade-off between time efficiency and performance.

2 Related Work

The system is constructed by two well-known techniques: TF-IDF and BERT-based models. The inspiration was extracted from the work of Soleimani et al. (Soleimani et al., 2019) who used BERT for document retrieval and claim verification from different labels including 'NOT ENOUGH INFO', 'SUPPORTED', and 'REFUTED', which had a similar objective to this project. And they also mentioned that there was a simple implementation to find the most relative document that was using TF-IDF and cosine similarity. TF-IDF can count the word frequency and score it which can be used to calculate the cosine similarity to find the top-k relevant evidences. Then BERT can focus on these top-k relevant evidences instead of training with all evidences that can effectively mitigate the limited hardware resource issue.

RoBERTa is another model in the BERT family. In general, RoBERTa performs better than BERT in larger datasets. Naseer et al. (Naseer et al., 2021) made an empirical comparison between BERT and RoBERTa on fact verification that is closely related to this project. They indicate RoBERTa performs slightly better accuracy than BERT. RoBERTa usually performs better with larger datasets and requires more computational resources. However, the hardware resources are limited in this project. RoBERTa performance will be evaluated in the following section.

3 Methods

3.1 Implementation

The training dataset was constructed by four elements: claim-id, claim-text, claim-label(4 label types) and evidences(a list of evidence-ids). A full evidence dataset was provided for evidence retrieval, each evidence was constructed by an evidence id and evidence text. The total number of training data was 1228, the development data was 154, the evidence dataset was 1208827 and the test data was 153.

Used TF-IDF to implement a basic filter function, set TfidfVectorizer parameters as default. Found top-k relevant evidences by calculating scikit learn cosine similarity between TF-IDF vectorized claim text vector and TF-IDF vectorized

evidence texts matrix. The function takes a claim text and returns a list of top-k evidence ids.

Converted the dataset to PyTorch dataset and dataloader. During creating PyTorch dataset, call the function above to retrieve the evidence ids, and use model's tokenizer to set the maximum sequence length to truncate and padding the sequence where necessary. Imported HuggingFace's official BERT-based and RoBERTa-based model, and used AdamW optimizer that helped to mitigate overfitting. Predicted test data and export the predicted results as JSON files that were used to evaluate Evidence Retrieval F-score(F), Claim Classification Accuracy(A) and Harmonic Mean of F and A(H-FA). See Figure 1 which interprets the process of prediction. Evidence Retrieval F-score evaluates the performance of evidence retrieval, Claim Classification Accuracy evaluates the claim label classification accuracy, and Harmonic Mean of F and A gives a general score for these two scores. Note: In this project, the BERT-based model and Roberta-based model can share the same parameters in the code so we can simply switch the model without changing other parameters.

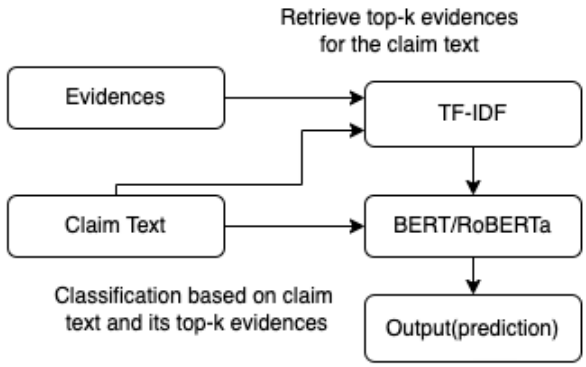

Figure 1: Predict a test claim text

3.2 Fine-tune Hyperparameter

Balancing the trade-off between time efficiency, limited hardware resource and performance, testing parameters in the order and ranges shown below:

- batch size: 1, 2, 4
- max sequence length: 128, 256, 512
- learning rate (lr): 1e-5, 2e-5, 3e-5, 4e-5, 5e-5
- k: 1, 2, 3, 4, 5
- Num of epochs: 1, 2, 3, 4, 5
- N-gram in TF-IDF: unigram, bigram, trigram

When the batch size was more than 4, it caused the kernel crashed, so the range is set to no more than 4

The initial settings for the two models were: max sequence length = 128, lr = 3e-5, k = 3, num of epochs = 3, n-grams = unigram.

Since max sequence length and n-grams significantly affect memory usage, so set them as the smallest value in the range can prevent time wasting at the initial stage. For other parameters, select the middle value to avoid bias. Note: No stop words were removed in this model because removing stop words may cause misunderstanding of some text and the dataset has multi-language.

Evaluated the performance, and selected the batch size that performs the best in the following stages and so on until all parameters were fine-tuned.

4 Evaluation and Analysis

The first thing to do is find the best batch size for two models from range 1, 2, 4. The initial parameters for both models: max sequence length = 128, lr = 3e-5, k = 3, num of epochs = 3, n-grams = unigram.

Model	batch size	F	A	H-FA
BERT	1	0.0959338	0.4610389	0.1588201
	2	0.0959338	0.5064935	0.1613136
	4	0.0959338	0.4870129	0.1602925
RoBERTa	1	0.0910482	0.4415584	0.1509673
	2	0.0910482	0.4675324	0.1524148
	4	0.0910482	0.4935064	0.1537337

Table 1: Find optimized batch size

According to Table 1, the optimized k in BERT is 2, and the optimized k in RoBERTa is 4.

The 3 outputs shown in Table 1 were rounded to 7 decimal places, and this will be applied to the whole project. As we can see in Table 1, F-score didn't change by batch size but it doesn't mean they were completely the same. They had tiny changes by the change of batch size that didn't bring significant changes to the F-score. Thus, the batch size could significantly impact the Claim Classification Accuracy. The system split evidence retrieval(TF-IDF) and claim classification into two parts(BERT/RoBERTa), and batch size was the parameter in BERT/RoBERTa so it didn't effectively affect F-score. RoBERTa performs better in larger datasets so its optimized batch size is 4 which was larger than BERT. However, BERT performs slightly better than RoBERTa. We will explore deeper in the following analysis.

Take the optimized batch size, and find the optimized max sequence length.

Model	max seq len	F	A	H-FA
BERT	128	0.0959338	0.5064935	0.1613136
	258	0.0959338	0.4740259	0.1595731
	512	0.0959338	0.4935065	0.1606404
RoBERTa	128	0.0910482	0.4935064	0.1537337
	256	0.0910482	0.4870129	0.1534151
	512	0.0910482	0.4870129	0.1534151

Table 2: Find optimized max sequence length

According to Table 2, the optimized max sequence length was 128 for both BERT and RoBERTa. Same to finding the optimized batch size, the max sequence length only significantly impacted the claim classification accuracy.

Model	lr	F	A	H-FA
BERT	1e-5	0.0910482	0.4350649	0.1505831
	2e-5	0.0910482	0.4350649	0.1505831
	3e-5	0.0959338	0.5064935	0.1613136
	4e-5	0.0910482	0.4545454	0.1517083
	5e-5	0.0910482	0.4805194	0.1530893
RoBERTa	1e-5	0.0910482	0.4415584	0.1509673
	2e-5	0.0910482	0.5064935	0.1543502
	3e-5	0.0910482	0.4935064	0.1537337
	4e-5	0.0910482	0.4090909	0.1489465
	5e-5	0.0910482	0.4415584	0.1509673

Table 3: Find optimized learning rate(lr)

The learning rate(lr) is a crucial hyperparameter in the BERT model that can affect the model's convergence speed, and it can significantly impact the performance of the model. According to Table 3, the optimized lr for BERT was 3e-5, the optimized lr for RoBERTa was 2e-5. Same to finding the optimized batch size, the learning rate only significantly impacted the claim classification accuracy.

Model	k	F	A	H-FA
BERT	1	0.0978355	0.5064935	0.1639936
	2	0.0891775	0.4610389	0.1494477
	3	0.0959338	0.5064935	0.1613136
	4	0.0909194	0.4025974	0.1483391
	5	0.0888219	0.4935065	0.1505479
RoBERTa	1	0.0978355	0.4870129	0.1629384
	2	0.0891774	0.4415584	0.1483866
	3	0.0910482	0.5064935	0.1543502
	4	0.0909194	0.4935065	0.1535500
	5	0.0900123	0.4155844	0.1479745

Table 4: Find optimized k

According to Table 4, the optimized k for both model was 1. Top-k relevant evidences was the most important parameter in this system. As shown in Table 4, it significantly affect both TF-IDF and BERT performance especially on BERT classification accuracy. However, if the system only retrieves only one relevant evidence for claims, it is easy to lose potential relevant evidences. And vice versa, if k is too large, the system retrieves too much irrelevant evidences that increase potential noise which results in decreasing of prediction accuracy. The hidden test data was different from the train and development data so it was prudent to set optimized k as 1. The second-best performance was from k = 3, this was a more compatible choice in the trade-off between losing relevant evidence and increasing noise. For deciding optimized k, testing k = 1 and k = 3 results by test datasets on Codalab, the k generated best performance would be the optimized k.

Model	k	F	A	H-FA
BERT	1	0.0675	0.4605	0.1178
	3	0.0721	0.5263	0.1268
RoBERTa	1	0.0675	0.4211	0.1164
	3	0.0883	0.5132	0.1507

Table 5: Find optimized k on test datasets

As shown in Table 5, in test datasets, the performance of k = 1 was apparently worse than k = 3. Therefore, the optimized k for both models was 3.

Model	Epochs	F	A	H-FA
BERT	1	0.0910482	0.4415584	0.1509673
	2	0.0910482	0.4480519	0.1513423
	3	0.0959338	0.5064935	0.1613136
	4	0.0910482	0.4545454	0.1517083
	5	0.0910482	0.4675324	0.1524148
RoBERTa	1	0.0910482	0.4415584	0.1509673
	2	0.0910482	0.4285714	0.1501893
	3	0.0910482	0.5064935	0.1543502
	4	0.0910482	0.4805194	0.1530893
	5	0.0910482	0.4805194	0.1530893

Table 6: Find optimized number of epochs

The number of epochs represents the model training time. Ideally, the longer training time would perform more accurate results. However, as shown in Table 6, the optimized number of epochs was 3 for both models. The longer training time would cause overfitting, so e=5 performed worse than e=3. Same to finding the optimized batch size, the number of epochs significantly affected the claim classification accuracy but couldn't effectively impact evidence retrieval.

The evidence retrieval module was built by only TF-IDF. N-gram was one of the most important factors in TF-IDF to impact evidence retrieval accuracy. Table 7 showed that the optimized N-gram was unigram. Bigram and trigram were restricted by the train dataset size. The larger dataset can pro-

Model	N-grams	F	A	H-FA
BERT	Unigram	0.0910482	0.7792207	0.1630453
	Bigram	0.0453772	0.3831168	0.0811436
	Trigram	0.0362554	0.4025974	0.0665204
RoBERTa	Unigram	0.0910482	0.4415584	0.1509673
	Bigram	0.0453772	0.4285714	0.0820653
	Trigram	0.0362554	0.4740259	0.0673589

Table 7: Find optimized ngram

vide more context for bigram and trigram, which produce lesser noise and improve performance.

Model	batch size	max seq len	lr	k	e	N-gram
BERT	2	128	3e-5	3	3	Unigram
RoBERTa	4	128	2e-5	3	3	Unigram

Table 8: Optimized parameters

The optimized parameters were shown in Table 8. And Codalab test results of these two models with optimized parameters was shown in Table 9.

Model	F	A	H-FA
BERT	0.0721	0.52630	0.1268
RoBERTa	0.0883	0.5526	0.1523

Table 9: Codalab results

All optimized parameters were shown in Table 8. And Codalab test results of these two models with optimized parameters was shown in Table 9. As a result, RoBERTa performed better than BERT, and it would be used as the final version of this system. RoBERTa as a variant of BERT, was designed to have longer training time than BERT. As mentioned above, the longer training time may cause overfitting but if the training time is appropriate, it improves the model's performance. Otherwise, RoBERTa uses a dynamic masking strategy and BERT uses a static masking strategy. A dynamic masking strategy randomly determines the mask for each sentence which reduces more data bias than a static masking strategy during the training process. Therefore RoBERTa possesses a more robust understanding of the context than BERT.

5 Final Evaluation

The final evaluation was shown in Table 10. The result was apparently worse than the result in the ongoing evaluation stage. Although RoBERTa has better performance than BERT, the train data was not enough to build a robust enough model. To improve the model performance, the most effective way is to use a larger training dataset and obtain more hardware resources. More train data is helpful to build a stronger model but it should be limited to a certain range to avoid overfitting. In the fine-tune hyperparameter section, all the tests were restricted by the hardware, if the model can be trained on a high-end machine, wider parameter ranges can be tested so the more appropriate parameters can be found that can enhance the model performance. Additionally, the system can be upgraded by integrating other techniques that have higher performance but more hardware resources.

Stage	F	A	H-FA
Ongoing	0.0883	0.5526	0.1523
Final	0.0639	0.4286	0.1113

Table 10: Codalab Evaluations

References

Muhammad Naseer, Imran Razzak, Hafiz Khurshid, and Kaya Kuru. 2021. An empirical comparison of bert, roberta, and electra for fact verification. In *2021 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 1–8. IEEE.

Amir Soleimani, Christof Monz, and Marcel Worring. 2019. Bert for evidence retrieval and claim verification. *arXiv preprint arXiv:1910.02655*.