COMP90042 Natural Language Processing Report

1136448

Abstract

This paper proposes an automatic fact-checking system aimed at improving the information reliability and accuracy of climate change-related statements. The system uses TF-IDF, BM25, and Bert models for evidence retrieval and classifies validated evidence into support, opposition, insufficient evidence, and controversy. Research results show that solving sample imbalance and using pre-trained language models for evidence retrieval are crucial, and the system can efficiently check statements to ensure that public opinion is based on accurate and reliable evidence.

1 Introduction

Climate change has always been a topic of discussion among scientists, policy makers, and the public. However, in recent years, the proliferation of unverified claims optimized by climate science has led to distorted public opinion. To ensure that the information obtained by the public is reliable and accurate, it is crucial to conduct fact check on these claims. Unfortunately, most fact checking requires manual verification by fact-checking platforms, which is both labor-intensive and expensive. Therefore, the development of an automated system is crucial to effectively fact-check these claims.

This paper aims to develop an automatic fact-checking system and explores various two-stage fact-checking methods. In Stage 1, the model retrieves relevant evidence from a knowledge base, while in Stage 2, it verifies selected evidence with climate science-related statements. Finally, the statements are classified into four categories: support, refute, not enough evidence, and dispute. The model can dig deep into evidence and understand the relationship between statements and evidence.

2 Data analysis

Using the datasets {train-claims, dev-claims, test-claims}.json, the Meteorological declaration and its evidences' ids were extracted, and the evidences were sourced from evidences.json, which contains a total of 1,208,827 evidence samples.

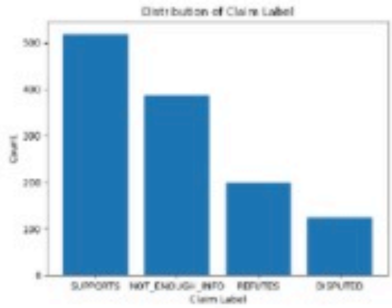

Figure 1: Label distribution

As shown in Figure 1, the tags for claim and evidence exhibit a significant imbalance. The majority of the tags are supporting and lack sufficient information, while the proportion of disputed and rebuttal tags is relatively low. Therefore, it is essential to address the issue of sample imbalance in subsequent model learning.

3 Task 1 : Evidence Retrieval

Given the claim and evidence provided, this step aims to select the most relevant sentences from the evidence as supporting evidence, which can be considered as a ranking problem. Therefore, the following model will be used for ranking。

3.1 TFIDF

TF-IDF is used to determine the importance of a word in a document or corpus. It takes into account both the frequency of a term in a document and its rarity in the overall collection of documents. The basic idea behind TF-IDF is that words that appear frequently in a document are likely to be important, while words that occur rarely are less important.

To calculate TF-IDF, first, the term frequency (TF) of a word in a document is calculated by counting how many times the word appears. Then, the inverse document frequency (IDF) is calculated by calculating the logarithm of the total number of documents in the corpus divided by the number of documents in which the word appears. Finally, TF-IDF is calculated by multiplying TF by IDF.

3.2 BM25

The BM25 algorithm is currently the most mainstream algorithm for calculating the similarity score between a query and a document in the field of information retrieval. Its main idea is to perform morphological analysis on the query, generating words q_i. For each search result D, the relevance score between each word q_i.and D is calculated, and then the relevance scores of q_i relative to D are weighted and summed to obtain the relevance score between the query and D.

Due to the large sample of evidence documents, we choose to use the BM25 model for initial screening in information retrieval tasks.

3.3 Bert

Bert(Devlin et al., 2019)(Bidirectional Encoder Representaions from Transformers) is a bidirectional encoding pre-trained language model that can effectively encode natural language text and generate relevant outputs through fine-tuning. The Hugging Face Community provides two official versions of Bert models, one with 12 layers of transformer and the other with 24 layers of transformer. Compared to traditional unidirectional encoding models, Bert can consider context information simultaneously, better capture syntax structure, semantic information, and contextual relationships, which makes Bert perform well in many natural language processing tasks such as text classification, named entity recognition, question-answering systems, etc.

The 12-layer transformer-based Bert model is suitable for basic NLP tasks such as sentiment analysis and keyword extraction, while the 24-layer transformer-based Bert model is more suitable for complex tasks such as machine translation and text summarization.

3.4 SimCSE

Inspired by the overfitting in neural networks caused by Dropout technology, SimCSE(Simple Contrastive Learning of Sentence Embeddings) (Gao et al., 2022) uses two different dropouts to pass a sentence through the same model twice, obtaining two sentence embeddings as positive examples and other embeddings as negative examples. Unsupervised learning is performed by comparing the loss. Compared to BERT, which is used for sentence similarity tasks, SimCSE can generate denser sentence vectors because one of BERT's pre-training tasks is completing fill-in-the-blank exercises word by word, while SimCSE can compensate for BERT's shortcomings in sparseness of sentence vectors. The advantage of SimCSE is that it brings closer the distance of positive example embeddings while pushing away the distance of negative example embeddings.

3.5 Evidence Retrieval Model Analysis

Evidence retrieval aims to identify evidence related to a specific query from a large corpus of text data. To evaluate the effectiveness of different evidence retrieval methods, this study tested four models separately.

3.5.1 TFIDF Retrieval Model

Select all evidence samples from the training set and test set, and remove duplicates to obtain a 3443-entry evidence library. Use TF-IDF to encode claim and evidences, and sort them in descending order based on scores. Select 2, 3, and 4 pieces of evidence for verification. Finally, select three pieces of evidence that perform best, with an F1 score of 0.0721 on the test set.

3.5.2 TFIDF with Bert

By comparing the simple use of TFIDF for evidence retrieval, we expanded the retrieval scope of TFIDF by screening 15 pieces of evidence initially. Then, we used the Bert pre-training model (bert-base-uncased) to encode the claims and evidences separately. The specific process was to take the output of the last hidden state of Bert and concatenate it with average pooling to obtain the corresponding sentence vector. Next, the encoded vectors of claims and evidences were generalized using L2, and finally, the dot product between them was calculated to get a similarity score. Finally, the top three evidences were selected in descending order based on their scores. F1 score on the test set is 0.1101. The improvement over simply using TFIDF is due to the fact that TFIDF scores only consider word frequency and text matching degree, while the Bert-encoded vectors contain semantic information.

3.5.3 BM25 with BERT

This algorithm uses the BM25(Lv and Zhai, 2011) algorithm to process evidence files and select claims and evidences. By calculating the BM25 score, it can find the relevance of each claim and evidence. It selects the top 60 candidates with the highest BM25 scores for each claim and evidence as options. This algorithm can consider the impact of document length, avoiding the influence of document length on retrieval results. Then, like using BERT encoding before, it encodes the 60 candidates and their corresponding claims, and calculates their cosine similarity. It selects the three most similar ones as final screening results. This algorithm has stable performance and a wide range of applicability, which can quickly find the most relevant evidence, improve the scope and reliability of candidate answers.

The F1 score on the test set is 0.1101, much lower than that of TFIDF. Through analysis, there are several problems: firstly, BM25 is used for comparing a sample and a document-level sample, considering the impact of document length by introducing a document length factor to adjust the weight of terms, thus avoiding the influence of document length on retrieval results. However, the text length distribution of evidence does not belong to long text, some are even very short; unlike TFIDF, which uses a full evidence library, BM25 uses a full evidence library, including a large number of negative samples (because of computational cost, they did not use TFIDF and BERT to calculate similarity between full data). The 60 selected evidence may be almost entirely negative samples, resulting in very poor performance.

3.5.4 Retrieval of Final Models

Based on the above model performance, the TFIDF with BERT trained after SimCSE model was ultimately evaluated based on the TFIDF with Bert and improved on Bert's sentence vector representation. SimCSE used all claims and evidentials from the training and validation sets as the dataset, and loaded the initial Bert pre training parameters for unsupervised SimCSE. Its F1 scores were (TFIDF with Bert) 0.1101 and (TFIDF with BERT trained after SimCSE) 0.1144. SimCSE unsupervised has had an effect because the overall quality of sentence vectors has improved. The comparison results of the models are shown in Table 1

Model	Test:F1
TFIDF with BERT	0.4211
TFIDF with BERT trained after SimCSE	**0.1144**

Table 1: Comparison of Evidence Retrieval Models

4 Task 2 : Fact Checking

The fact-checking task is conducted after the processing of evidence retrieval. In this task, the claim and evidence text are concatenated using the following format: [CLS] claim [SEP] Evidences [SEP] [Evidences] [SEP]. This construction method is based on the next sentence prediction in BERT's pre-training, which is used to determine whether the sentence after the [SEP] label is the next sentence. We convert it into a fact-checking task, map it to a 4-class classification through a linear layer, and finally use the model to determine the relationship between subsequent evidences and claims.

4.1 Data preprocessing

The distribution of claim and evidence text length in the training set is shown in Figures 2 and 3, respectively.

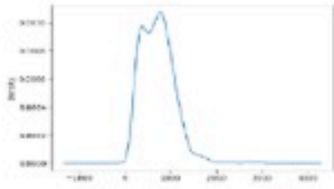

Figure 2: Training Set Data Distribution

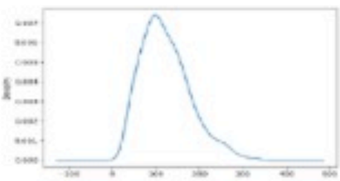

Figure 3: The distribution of evidence length

By analyzing the above chart, we can observe that the sample length of claims is around 120, and the text length of evidences is between 300-1000. However, Bert's input length is only 512 tokens. Therefore, to split each evidence into sentences, calculate cosine similarity between each sentence and claim, select the top 5 sentences with the highest similarity and concatenate them after the claim. Finally, it was saved as {train, dev, test}.json files.

4.2 BERT：Learning Rate 5e-5

The performance of the ACC score in the development set is 0.5514, which is due to hyperparameter tuning. The model was trained using 1e-5 and 8e-5 for both training and validation, with 5e-5 being the optimal performance on the validation set. However, the model's performance on the test set is 0.3421, which may be due to overfitting during training.

4.3 BERT : Learning Rate 5e-5 with Focalloss

To address the imbalanced data distribution, It can be solved by Focalloss(Lin et al., 2017), which improves the original cross-entropy loss function by adding a modulation factor $(1-p_t)^\gamma$ on top of the original

cross-entropy. This factor is designed to balance the importance of difficult and easy samples while increasing class weights α, which can balance the importance of different class samples. The best α value set in experiments is 2. Finally, the ACC on the validation set is 0.5714, and the test set performance is 0.4211. Compared to the baseline model, the model's generalization ability has increased, which indicates that the punishment for the majority class and the weighted addition of the minority class have played a crucial role.

4.4 BERT:Learning Rate 5e-5 with Focalloss and Bidirectional LSTM and Bidirectional GRU

Adding BiLSTM(Chen et al., 2017) and BigRU after BERT model improves the model complexity. BiLSTM and BigRU can enhance the model's ability to represent sentences, so connecting them to BERT and then to a linear layer output can result in better sentence vector representation. The ACC on the validation set is 0.5974.

4.5 SimCSE-BERT:Learning Rate 5e-5 with Focalloss and Bidirectional LSTM and Bidirectional GRU

Using the first stage of SimCSE to train BERT vectors as sentence representations resulted in a rapid decrease in loss during model training. This was due to the denser and better sentence representations obtained by the BERT vectors. The validation set achieved an ACC of 0.6233.

4.6 Evaluation od Final Models

Based on the above model performance, the selected BERT: Learning Rate 5e-5 with Focalloss and Bidirectional LSTM and Bidirectional GRU models to evaluate. The test set ACCs were 0.4605 and 0.4211 respectively. We found that the SimCSE-BERT model performed lower on the test set than the BERT model but higher on the validation set. The reason for overfitting was that during training of SimCSE, claims and evidence were trained together, while when input facts checking model, they were concatenated together, which led to a large difference in the sentence representations learned by the model and more noise being learned by the model. The comparison of the effectiveness of the fact verification model is shown in Table 2

Fact heck Model	Dev:ACC	Test:ACC
SimCSE-BERT：Learning Rate 5e-5 with Focalloss and Bidirectional LSTM and Bidirectional GRU	0.6233	0.4211
BERT：Learning Rate 5e-5 with Focalloss and Bidirectional LSTM and Bidirectional GRU	**0.5974**	**0.4605**

Table 2: Comparison of Evidence Retrieval Models

5 Conclusion and Future Work

In conclusion, for Task 1: Evidence Retrieval, this project do not show a ideal results and high level of performance. It is difficult to retrieve relevant evidence from such a large evidence database, and using similarity calculation to vectorize sentences may lead to a lot of noise. For Task 2, the model's generalization ability is still insufficient, and using the method of claim concatenation with evidence may have limitations. Therefore, the following improvement ideas are proposed for the future:

1. Use vector retrieval libraries such as Faiss to build an evidence vector index library, and then retrieve similar evidence through vector recall.
2. For evidence, generate a summary using the model to generate a golden summary, and calculate the similarity between the summary and the claim to obtain the best evidence. Send the golden evidence into the model training for fact-checking tasks.
3. Adopt a Joint Fact-Checking Model that first generates an implicit summary automatically in the intermediate stage and then outputs classification through a fully connected layer.
4. Introduce more text features, such as sentiment features, style features, etc.
5. Use ensemble learning algorithms that involve cross-validation and multi-model voting to improve the model's generalization ability.