Tue3PM_Group6 Project: Report on Textual Classification and Evidence Retrieval

Tue3PM_Group6

Abstract

This report presents a comprehensive model for information retrieval and contextual classification. The approach is structured into three main sections: data pre-processing, information retrieval, and textual classification. Initially, the BM25 algorithm, augmented with Named Entity Recognition (NER), is applied to filter the evidence dataset that is essential for the classification phase. Subsequently, a Bidirectional Long Short-Term Memory (Bi-LSTM) network and Supported Vector Machine are utilized to decide whether each evidence in the first set of selected evidence is relevant to the claim. The proposed approach demonstrates significant potential in improving the accuracy and reliability of textual classification tasks.

1 Introduction

With the development of the Internet, information can be accessible easily through various online platforms. Consequently, numerous topics have been concerned, among which climate change is one of the hottest. Even though abundant information about climate change is available on the Internet, some information in the form of video and short articles may mislead the readers and viewers which make them have the wrong idea about climate change. Hence, a classification model is essential since it can be greatly helpful to distinguish whether the opinions shown by the authors have corresponding evidence to support them.

2 Approach

The approach used is mainly divided into three parts which are data pre-processing, information retrieval, and classification based on the concatenation of evidence and claim. (As shown in figure1)

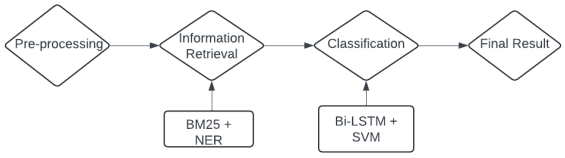

Figure 1: Approach Procedure

2.1 Data Preprocessing

Data files are all in the form of json files. In the evidence file, there are over 1.2 million pieces of evidence that need to be processed. And for the training, dev, and test data, we also have over 1,000 claims that need to be preprocessed.

First, Named Entity Recognition imported from spacy is used to create a map and a weight is assigned to each kind of entities.

Then, because some evidence is entirely composed of non-English words, which is clearly not relevant to all the claims. Only the words composed of number or English were kept and words like symbols or combinations of other languages were removed.

After these steps, data was tokenized. The tokenized result included two parts: the two words before and the two words after each entity, and the original claim tokens. The tokenized result included two parts: the two words before and after each entity, and the original claim tokens "lemmatized and with stop words removed. This tokenization approach keeps more context information and helps recognizing special terms, which is useful for evidence retrieval and classification.

2.2 Evidence Retrieval

After completing the data pre-processing, the following step is evidence retrieval.

In this part, WordNet from NLTK is imported to look up synonyms of entities. Synonyms

were randomly selected and subsequently three additional claims were created. BM25(Robertson and Zaragoza, 2009) scores were calculated for these claims. The original claim's BM25 score was given a weight of 1.5 while the three new claims were given a weight of 1. We added up all the scores. Then we sorted the scores and selected the top four pieces of evidence and returned the corresponding evidence IDs. These pieces of evidence were used as the result of evidence retrieval. This choice was based on the trade-off between recall and precision, which will be explained in the next section on evaluation and experiment.

Furthermore, we retrieved the top 50 pieces of evidence from both the training data and the development data. These pieces of evidence will be used in the next steps to further classified labels, ensuring diversity and completeness of the data during the training process.

2.3 Classification

First, we imported the claim and evidence and translated the data from JSON to DataFrame. We then classified the label into three types: *support*, *refute*, and *irrelevant (NOT_ENOUGH_INFO)*. In this step, we ignored the *disputed* label since it contained both supporting and refuting claims without a clear proportion. This label was excluded in classification training phase to avoid noise.

Lau and Baldwin (2016) provides a new idea about text to vector of data, Doc2Vec is utilized to transform the text data into vectors. We first preprocessed the text by tokenization and removed non-English word. After that, we used the Doc2Vec model from gensim to create vectors of fixed length. We set a vector size of 200 and trained for 50 epochs.

We developed a neural network model based on LSTM to execute the claim classification. The model includes a bidirectional LSTM layer with an embedding dimension of vector size 200 and a hidden dimension of 256 and a fully connected layer to generate the final classification results. We trained the model using the cross-entropy loss function and the Adam optimizer. Also, we used early stopping to prevent overfitting during 400 training epochs. This model will generate a three-dimensional array of possibility for each claim and its corresponding evidence. (One-to-one model)

Since we only had about 1,000 training samples, we imported WordNet from nltk to look up synonyms and we used these synonyms to replace the original words and create synonymous sentences. We paired these new sentences with the origin top 50 pieces of evidence and labels. To enhance the classification between *SUPPORT* and *REFUTE*, we created two additional synonymous sentences for these two labels. And for *NOT ENOUGH INFO* and *DISPUTED*, we created one more synonymous sentence. This increased the training data volume and improved the performance of classification training.

Finally, these data was fed into an SVM model. We trained the model on each claim with its 50 pieces of evidence and the three-dimensional array generated from previous model. (One-to-many model) This process output our final four labels of each claims. This multi-step training improves the overall performance and accuracy of the model.

3 Experiments

3.1 Evaluation Method

Due to the limited number of submissions we have to test the metrics on the test dataset, and since the test dataset on Codalab is divided into public and private parts which means that the data volume for each part is smaller and its metrics will be less stable, in this part of the report, we will primarily use the metrics from the development dataset to evaluate our model.

In the evidence retrieval, the key evaluation metric is F1 score. In each evidence retrieval method we get the top-n evidences, and calculate the recall and precision with respect to the true evidences given by the development dataset, such that we retrieve the F1 score. We tune top-n in each method and the best n with highest F1 score is selected for comparison between methods.

In the claim classification, accuracy of classification is the key metric. With the best potential evidences selected from the retrieval method, we evaluate the models in terms of the correct classification instances on the development dataset.

3.2 Experimental Details

3.2.1 Data Preprocessing and Evidence Retrieval

In Figure 2, we present the experiment and their corresponding F1 scores for dev data in data preprocessing and evidence retrieval.

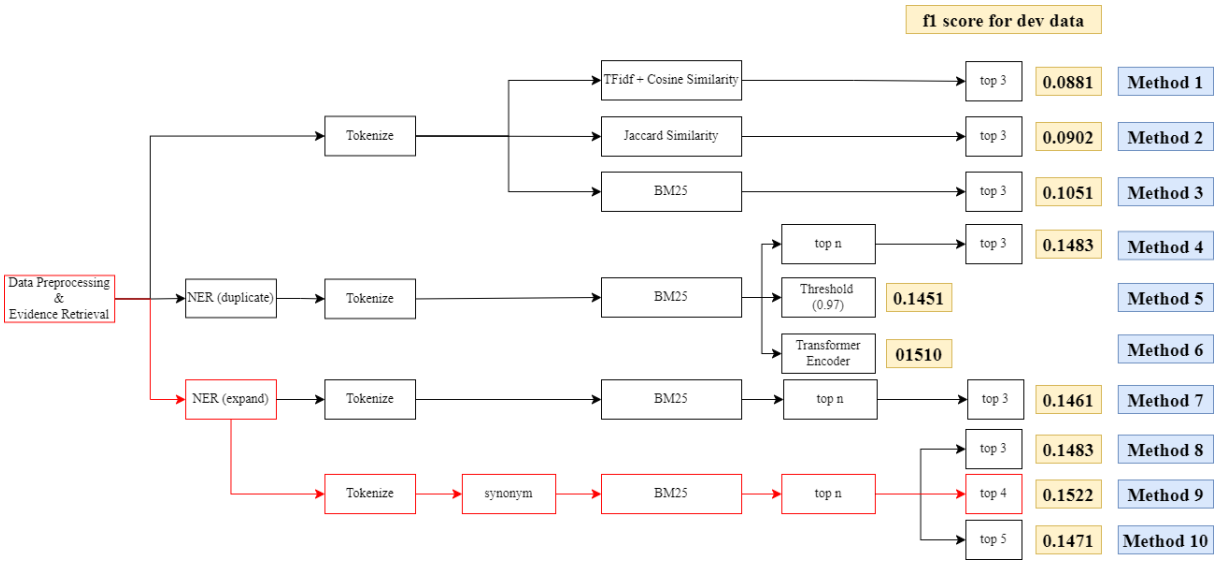

Figure 2: Experiment (Data Preprocessing and Evidence Retrieval)

At the top of the Figure 2(Method 1 to 3), after basic tokenization (lemmatized and stop words and non-English words removed), we tested three approaches: tfidf with cosine similarity, jaccard similarity, and BM25 in evidence retrieval. We found that jaccard similarity is more effective when sentence structure and word usage are similar to the original claim. However, not every evidence follows the exact sentence structure and the order of the claim. Also, tfidf reflects a common limitation that frequent terms might overshadow less frequent but more relevant terms. This issue can be addressed by adjusting the parameter k1 in BM25. Moreover, BM25 allows for adjusting parameter b to control document length normalization. Therefore, we decided to use BM25 for scoring.

Next, we introduce the concept of NER into our approach.(Method 4 and 7 in Figure 2) We imported it from spacy and created a map of entities. In the first NER method (duplicate), we simply duplicate the entity three times and add it to the claim token. In the second method (expand), we assign the weights based on the types of entities and duplicate the entities according to their weights. Additionally, we consider that the context before and after the entity may be removed during tokenization but is actually relevant to the entity. Therefore, we include the two words before and after the entity in our tokenization results. Through NER, we can enhance the importance of special nouns and optimize the results of evidence retrieval.

When selecting BM25 parameters, we used the NER duplicate method mentioned earlier to test different combinations of k1 and b and plotted a heatmap(As shown in Figure 3). It showed that 0.6, 0.6 are the most suitable k1 and b for development data. As a result, these parameters will be chosen as the default values for BM25 in subsequent methods.

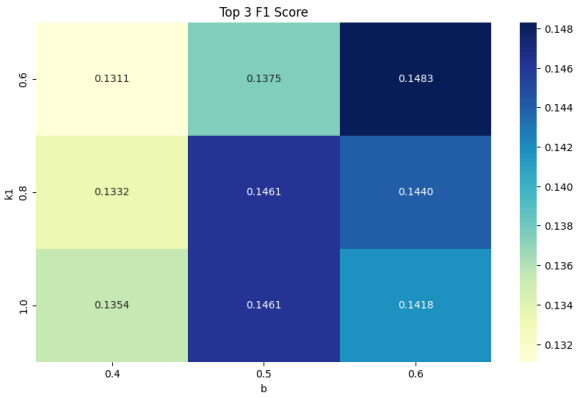

Figure 3: BM25 parameter heatmap

Although the results of duplication(method 4) are slightly higher than expansion(method 7) on the dev data, expansion considers complicated scenarios and achieved better results when predicting test data.

While testing the NER duplicate results, we also experimented with several methods to analyze the BM25 score (Methods 4 to 6 in Figure 2). Method 4 directly selects the top n pieces of evidence as the result. Method 5 is used to mitigate any potential significant differences in BM25 rankings

by excluding evidence when its score is lower than the previous one multiplied by our defined threshold (the most effective threshold found by our testing was 0.97). Method 6 is inspired by the sentence re-ranking part of (Diggelmann et al., 2021). We firstly pre-trained a transformer encoder model using masked word task on the entire evidence corpus, and then fine-tuned the model with classification of evidence and non-evidence on the best BM25 selection results. Lastly, it is used to filter out the lowest predicted scores among the BM25 results.
From the results, it showed that Method 5 performs the worst, while transformer encoder performs the best. However, considering the complexity and computational resources, the top-n approach is simple but generates similar results. To enhance our coding efficiency, we choose to use the top-n method.
Finally, Method 8 to 10 in Figure 2 show our final approach of this report. As mentioned in section 2.2, we introduced the concept of synonyms entities and expanded the claims number when calculating BM25. We tried to address the issue of insufficient data in this process. Followed by the top-n method, we found that its results even better than using BERT, particularly in the top 4. Hence, it was selected as the method in our report.

3.2.2 Classification

Method	Accuracy
Doc2Vec + BiLSTM + Logistic regression	0.38
Synonym replacement + Doc2Vec + BiLSTM + Logistic regression	0.42
Synonym replacement + Doc2Vec + BiLSTM + SVM	0.46
Transformer encoder	0.31
Synonym replacement + Transformer encoder	0.43

Table 1: Accuracy of methods

The table 1 above shows the accuracy of each model. In the experiment, we mainly focus on two models: one is the Doc2Vec+BiLSTM model discussed in the approach section, and the other is the transformer encoder pre-trained in the evidence retrieval part , which is fine-tuned on the labeled claims using the point-wise approach as described in this paper(Soleimani et al., 2019).

One major issue in the dataset is the unbalance classes. The *SUPPORT* class takes almost half of the training dataset. This cause our models to bias towards the major classes. To fix this issue, we use synonym replacement to generate more training cases for the minor classes in the training process. As a result, we observed using synonym replacement in the BiLSTM model improved performance by 9.5%, which is not stable because Doc2Vec is non-deterministic while in the Transformer encoder, it improved performance by 27.9%.

But we must admit that the performance of classification models is poor than simply classifying all the instances as *SUPPORT*. As described in the entailment prediction section (Diggelmann et al., 2021), Bert trained on the FEVER dataset (Thorne et al., 2018) which contains more data than our Climate-FEVER dataset achieved label-accuracy of 77.68% on our dataset. We believe that with limited training data, even with data augmentation techniques such as synonym replacement, the model cannot effectively recognize the semantics of *REFUTE* and *DISPUTED*, and can only achieve some recognition ability on the more abundant *SUPPORT* category.

4 Conclusion and future work

In conclusion, the integrated model described in the report uses the model for evidence retrieval and the model for classification. It can be highlighted that BM25 and Named Entity Recognition are crucial to extract the relevant evidence. Additionally, in the classification process, we used Bi-LSTM to generate three-dimensional arrays one-to-one and SVM to generate labels one-to-many. The results are acceptable since the data size is limited.
One striking method is using synonym replacement techniques to generate more training samples. In evidence retrieval process, synonyms slightly helps improve the volume of the claim and the performance of evidence retrieval. In the classification process, synonyms can have a positive impact by helping alleviate the risks of insufficient and imbalance data. Meanwhile, it helps interpret semantic issues with different wordings used in the evidence and the claims.
Furthermore, some improvements are necessary to enhance the overall performance of the model. The noisy data is a significant limitations when training and testing the model. Also, the limits on using pre-trained models may make it not effectively

capture semantics and contexts within the claim. Hence, future work may focus on expanding datasets, or alternatively, use the pre-trained model more efficiently if there is no such a constraint that pre-trained model is not allowed in training process. This will help the model capture complex semantics within the claim more accurately.

5 Team Contribution

All team members in the project group, which comprises 4 members, has contributed significantly to the project. Below is the detailed information of each member's contributions:

Yueheng Huang:
Model development and optimization: He is the core member who developed the model which can generate a good result.
Presentation: He contributed in presentation by creating and designing the slides. he covered the part of Introduction, project overview and dataset overview. He did play a key role in answering questions in QA session.
Report Writing: He mainly focused on part of experiment.

Yue Peng:
Model Development and optimization: He worked alongside with Yueheng on optimizing the model. He also contributed to model development by reading articles and implementing.
Presentation: He contributed by creating and designing slides and covered the part of classification.
Report writing: He, Christine and Yueheng are mainly focusing on big part, the experiment.

Christine Huang:
Model development: She participated in initial model development and implementing algorithms.
Presentation: She played a key role in presentation since she covered an essential part, information retrieval, and designed the slides by her own. She played a key role in asking the critical question in QA session.
Report Writing: She covered the main part of Approach and worked with Yue Peng and Yueheng to write the part of Experiment.

Risheng Wang:
Model development: He contributed to initial model development by implementing several models and algorithms.
Presentation: He covered the data pre-processing part and contributed to the design and creation of slides.
Report writing: He contributed to the report writing by covering abstract, introduction, part of Approach and conclusion.

References

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2021. Climate-fever: A dataset for verification of real-world climate claims. *Preprint*, arXiv:2012.00614.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. *CoRR*, abs/1607.05368.

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

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

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.