# Automatic Fact Verification

*Wenbin Cao (1008253) Yiming Zhang (889262)*
*CodaLab team name: UKnowNothing*
*CodaLab Public Board Result: Sentence F1: 72.0, Label Accuracy: 64.4*

## 1. Introduction

The Automatic Fact Verification system is a system that can automatically validate whether a claim is true, false or unverifiable based on the information in a large text corpus. The system can find the sentence in the text corpus which related to the query claim, and then use the evidence to predict the validity of the claim. The text corpus consists of 109 wiki TXT format files and the files contain one Wikipedia sentence per line, formatted as Page identifier, sentence number, sentence text (Thorne et al., 2018).

## 2. Preprocess

The preprocessing of text corpus is the first stage of the Automatic Fact Verification system. We observed that the sentence text often contains pronouns such as “we”, “he”, etc and it will be difficult to further sentence retrieval and label prediction. And we also observed that the page identifier is very useful to the sentence text and contains important information. Hence we preprocessed the Wikipedia test corpus and added the page identifier to the sentence part for each sentence line. Moreover, during the preprocess we found there are many non-English words, and brackets such as “LRB”, “LSB”, “RSB”, “RRB” in the text corpus. We decided to remove these because they are irrelevant to data analysis.

## 3. Information Retrieval

The information retrieval method in the Automatic Fact Verification system is to find the sentence in the corpus which is most related to the query claim. It employs a pipeline architecture which contains document retrieval and sentence retrieval. The figure1 below illustrates the pipeline of the information retrieval process. For example, if the claim is “Bermuda Triangle is the western part of the Himalayas”, then after document retrieval method the system will find documents related to “Bermuda_Triangle” and “Himalayas”. And after the sentence retrieval, it will find the sentence on the Bermuda Triangle with sentence index 0.

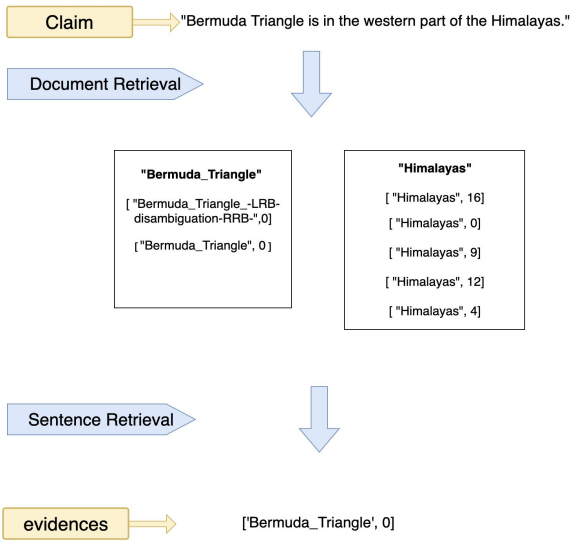


Figure 1: Illustration of the Information Retrieval Pipeline

### 3.2 Document Retrieval

#### 3.2.1 Document Retrieval Basic Method

The document retrieval method is rule-based information extraction. By observing the training dataset, we found that the evidence of the claim frequently includes the upper case word in the claim. Hence we started this method by building a dictionary of page identifiers and sentence number of all the Wikipedia text corpus. Then the rule-based information extraction method will find all the uppercase word in the claim and if the

uppercase word exists in the dictionary then we will extract this document.

### 3.2.2 Document Retrieval Error Analysis

The basic document retrieval method will only extract the uppercase word. However, it will miss any n-gram words hence it would miss many documents. Some typical examples of this type of error are listed in the table below.

| **Documents retrieved by the Basic method** | **Expected Documents** |
|---|---|
| Home | Home for the Holiday |
| Holiday | Home for the Holiday |
| American | American |

Table 1: Illustration of the errors in document retrieval

### 3.2.3 Document Retrieval Advanced Method

The advanced document retrieval method is to modify the rule-based information extraction method. The advance rule-based information extraction method can find unigram, bigram, trigram and 4-gram uppercase word with a preposition. For example, in the basic method, if the claim is " Home for the Holidays stars an American actress", then it will find "Home", "Holidays" and "American" from the claim. And the advanced method will find "Home for the Holiday" and "American" in the claim. Also, we update the page identifier dictionary and add new values if the page identifier has other meaning with brackets. For the above example, after updating the dictionary, we can now find documents such as "Home for the Holiday -LRB-1991-RRB-" by the advanced method. After applying the advanced method, the document F1 score increased from 63.17% to 68.56%.

## 3.3 Sentence Retrieval

The sentence retrieval method is based on the result of the document retrieval. After the document retrieval, there are only a few sentences left that are relevant to the claim. Then we preprocessed the claim and the retrieved sentences to remove stop words, tokenization, and lemmatization. After preprocessing, we applied several sentence retrieval method to find the best sentence or sentences as evidence.

### 3.3.1 Sentence Retrieval Basic Methods

We experimented four sentence retrieval methods to calculate the cosine similarity between the claim and evidence includes BOW, TF-IDF, TF-IDF with SVD, and TF-IDF with a threshold (Tata et al,. 2007). The TF-IDF with threshold performs best among these methods and has the highest sentence F1 score 51.46% while the TF-IDF method has the best recall which is 64.50%. The evaluation of these methods is shown below.

| | Sentence Precision | Sentence Recall | Sentence F1 |
|---|---|---|---|
| BOW | 26.86% | 51.90% | 35.40% |
| TF-IDF | 30.20% | 64.50% | 41.14% |
| TF-IDF SVD | 15.24% | 36.20% | 21.45% |
| TF-IDF Threshold | 44.43% | 58.15% | 50.37% |

Table 2: Sentence Retrieval Basic Methods Evaluation

### 3.3.2 Sentence Retrieval Error Analysis

We found several problems for the sentence retrieval approaches above:

1. TF-IDF will find many sentences with low cosine similarity, hence we set a

threshold to filter low cosine similarity evidence.

2. We observed some word has ‘s or ‘ suffixes in the claim and it will affect the performance of the method. Hence we removed some suffixes and it improved the performance.
3. We observed that some words such as “Monday”, “March” never appears in the actual evidence list, hence we remove some document page identifiers.

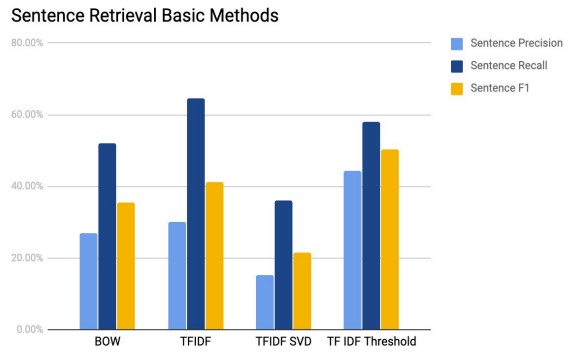


Figure 2: Sentence Retrieval Basic Methods Evaluation

### 3.3.3 Sentence Retrieval Advanced Method

It can be observed that for the TF-IDF approach, the Sentence Retrieval recall is high (64.50%) but the precision is rather low (only 30.20%), which represents that although some true sentences are covered, there are still many incorrect selections. We fine-tuned a pre-trained BERT model (Devlin et al,. 2018) based on the TF-IDF results to solve this problem.

After the Document Retrieval process with rules, we use the sentences which have top 7 TF-IDF similarities as the candidate evidence for each claim. Then we combine each claim and a piece of candidate evidence as a sentence pair and label them as “1” (for have relation) or “0” (for not have relation) according to the gold standard (the training set). Finally, we fine-tuned the pre-trained BERT model on these claim-evidence sentence pairs, let the model help us to detect which candidate evidence should be kept and which should be discarded. Table 4 displays the detail configurations of the fine-tuned BERT model.

| | |
|---|---|
| Pre-trained Model | BERT-Base, Cased |
| Lower Case | FALSE |
| Seq Length | 60 |
| Batch Size | 32 |
| Learning Rate | 2.00E-05 |
| Epochs | 1 |

Table 4: Configures of fine-tuning BERT model

There are about 900,000 sentence pairs generated from the training set. It took about 5 hours to train on our GTX 1080 GPU for 2 epochs. It was noticed that the performance started to drop off since the 2nd epoch so we chose the model trained by 1 epoch.

After the sentence selection by BERT model, the Sentence F1 score reaches **70.81%** and Document F1 was **81.44%** on the develop set.

## 4. Label Prediction

It is a 3-class classification task for the label prediction step. Since there is no evidence provided for “NOT ENOUGH INFO” class in the training set, we use the top 3 TF-IDF sentences as the “evidence”

for these "NEI" claims, which is similar as the *NEAREST* approach in the Fever paper. The details of the methods we tried will be discussed next.

**4.1 TF-IDF Features with the MLP Model**

For this baseline model, we refer to the simple approach by Riedel et al, (2018). on the Fake News Challange. We chose the top 5,000 raw TF features for both claim and evidence, and 1 TF-IDF similarity feature. Then these 10,001 features are feed to a single-layer MLP with 100 hidden units and a softmax layer for output.

**4.2 BERT as Features with the MLP Model**

We also tried to use doc embeddings generated by the pre-trained large BERT model as training features. In this approach, we use the "*bert-as-service*" tool by Hanxiao (2019) with pre-trained Large -Cased BERT model to generate doc embeddings as features. Then we use two NN dense layers to process these features and output.

**4.3 Fine-tuned Base BERT Model**

Finally, we adopt a similar method as the sentence selection step - A fine-tuned BERT model. For this time, the sequence length is set as 150 since the pieces of evidence are concatenated together. This model took about 2 hours to train on a Tesla V100 server GPU for 2 epochs.

All these models are trained on the ground-true evidence. The performance for these 3 classifiers are shown below:

| | Acc. on ground true data | Acc. on practical data |
|---|---|---|
| Method 4.1 | 72% | Not tested |
| Method 4.2 | 89.50% | 59.80% |
| **Method 4.3** | **93.60%** | **64.10%** |

Table 5: Performance of Label Prediction classifiers

**5. Conclusion**

In conclusion, we use a Rule-Based approach to retrieve the documents. Then select the evidence from the sentences which has top 7 TF-IDF similarity with the claim and filter them with a fine-tuned BERT model. Finally, the labels are predicted by another fine-tuned BERT model.

| | |
|---|---|
| Document Selection F1 | 80.9% |
| Sentence Selection F1 | 72.0% |
| Label Accuracy | 64.40% |

Table 6: Final performance on the test set (public board)

Our system got a competitive performance on this Fact Verification task (Table 6). However, the sentence selection model always discards some true evidence to guarantee its loss low, and this caused the label predictor loss some critical information. This problem could be overcome in possible future works.

**References**

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Sandeep Tata and Jignesh M. Patel. 2007. Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM SIGMOD Record*, 36(2):7–12.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language. *CoRR*, abs/1810.

Riedel, Benjamin, Augenstein, Isabelle, Georgios P., Riedel, and Sebastian. 2018. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. May.

Hanxiao. 2019. hanxiao/bert-as-service. May.