Automatic Fact Verification

Tong He 867488 **Yao Wang 869992**

Abstract

Misinformation from unreliable sources arouse wide concern from media coverage and recent researches. Automatic fact checking has brought to the forefront NLP release. In this paper, we present the system designed for this functionality based on TF-IDF (BM25), POS, NER and AllenNLP model that conduct document retrieval, sentence selection and claim judgement jointly for automatic fact verification. We employ the dataset of 3.47 GB claims generated by altering sentences extracted from Wikipedia. The dataset is labeled as 'SUPPORTS', 'REFUTES', and 'NOT ENOUGH INFO' with necessary evidences for the claim. For evidence retrieval (document retrieval and sentence retrieval) We leverage TF-IDF and PMI approaches to pre-designed term vector space model to match queries and sources. After employing AllenNLP Text Entailment model in the label selection module, in preliminary evaluation, our system achieves .443 label accuracy, and .30.7 F1 on the FEVER shared task dev set.

1 Introduction

Given the explosion of textual sources with unknown source and verification may lead to massive fake news propagation, people are eager for an automated fact checking system to assist them judge information from the web. This trend stimulates state-of-art research efforts on employing traditional IR models and machine learning system especially deep learning neural networks for automatic fact verification. Besides, as the ever-increasing amounts of textual content, researchers have the luxury of leveraging large-scale dataset to develop reliable fact checking systems. Recently, a new publicly available dataset, FEVER: Fact Extraction and VERification introduced a benchmark fact verification task to test a combination of retrieval and textual entailment performance [**1**]. This challenge is launched to verify a claim using formalized evidential documents and sentences extracted from Wikipedia corpus. Claim is asked to label as 'SUPPORTS', 'REFUTES' and 'NOT ENOUGH INFO' correspond to having evidence to support the claim, or to refute this claim and no evidence is provided for this claim, along with the one or more evidential sentences. This system can be divided into two processes, first to precisely find the evidential sentences from massive entries and then to conduct textual entailment between the claim and each sentence.

In this work, we propose the joint system which, given a claim, directly selects top-k relevant sentences ranking by the combination of TFIDF with different parameters, PMI, POS, NER, Title embedding and classifies the claim evidence respectively by AllenNLP model. We separately evaluate the performance of the sentence selection system as well as entailment system and finally evaluate the overall system.

2 System Architecture

The system is a pipeline model We constructed a simple pipelined system comprising three components: database system, sentence retrieval and textual entailment recognition. Figure 1 illustrates the system architecture.

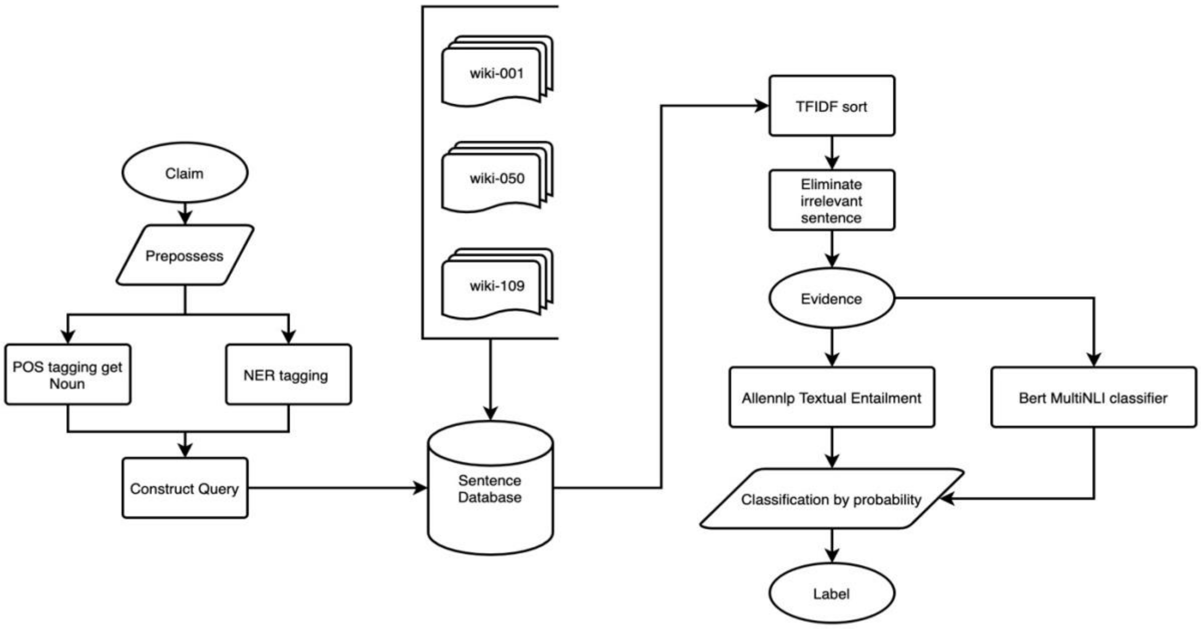

Figure 1: Architecture of the automatic fact verification system.

2.1 Database construction

The data source contains a total of 109 txt files in which each line represents a sentence derived from wiki. Reading and retrieving data directly from them is inefficient and complicated thus we leverage Lupyne python library based on PyLucene to preprocess and store the texts to simplify and speed up the query [3]. We also tried Whoosh which has similar functionalities as Lupyne to generate the database but it takes 10 hours to go whereas Lupyne takes 2 hours.

2.2 Sentence retrieval

Given a claim, it is obviously impossible to compare it with all the sentences in the dataset to judge whether it is correct or not and thus we need to narrow the scope first. Unlike normal IR processing, that is, match the most relevant documents and then selects the most relevant sentences within them as it may leave out the right evidence, we directly use TF-IDF and PMI to return top K sentences. We replaced the TFIDF algorithm with BM25, smoothing the effect of article length and word frequency on similarity. POS tagging and Named Entity Recognition (NER) are employed to analyze the keywords and give higher modified weights of them. AllenNLP provides NER module to identify NEs. For words other than NE, nouns are more important than other words. With NLTK POStagger we can extract the part of speech and give the noun a relative higher weight in the search [2]. Besides, the right evidence for a claim tends to appear in the wiki page whose entry contains the claim's NEs and pronouns will lead to a lack of keywords in a retrieved sentence. Hence, we also consider the title and sentence in consideration in the retrieval.

2.3 Textual entailment system

After completing the selection for the sentences, we use Allennlp pre-trained textual entailment model to evaluate the correlation between the claim and candidate sentences. the model receives two sentences, premise and hypothesis (the former is the inferred condition, the latter is the deduced conclusion), and return three probabilities of entailment, contradiction and neural here respectively corresponding to support, refute and not enough info [4]. We use claim as hypothesis whereas we have k selected sentences. Here are two ways. One is to bring each sentence into the model to get probability distribution and judge the results by manual logic such as taking the most common label. However, this method causes two problems: 1.the relationship between sentences is not considered in that a small part of the claims requires more than one sentence to be correctly combined to infer the conclusion 2. The bias of retrieval such as the proportion of ambiguous words will affect the final result. As such, we choose the second way in that we combine k sentences into one and input it as premise to get the only result. Although this introduces an irrelevant context to sentences, it reduces the impact of irrelevant sentences on the predictions.

3 Results

3.1 Sentence selection

Table 1 shows the performance of different algorithms with top 5 returned. the precision and recall evaluated from traditional TFIDF algorithm are very low, which cannot meet the need of label predictions. We have tried many modifications based on TF-IDF and finally found that combining BM25, NER, POS and Title has a relative higher performance.

k=5	Precision (%)	Recall (%)	F1 (%)
Baseline	----	----	20
TFIDF	9.8	42.8	15.9
PMI+N+P	14.7	38.6	17.9
TFIDF+N+P+Title	15.9	51.5	24.2
BM25+N+P+Title	19.6	55.45	28.96

Table 1: Precision, recall and F1 score for evidential sentence selection in different approaches in dev set where N refer to NER and P refer to POS tagging.

Table 2 shows the performance of the retrieved data when using a larger k value. Using only the first 5 pieces of data does not capture all the evidence well. After extending the range to 20, recall improved by 11.9. A possible improvement is to use some models to filter sentences to improve accuracy when returning more sentences.

k	Precision (%)	Recall (%)	F1 (%)
1	43.8	38.0	40.8
3	21.6	44.4	29.0
5	19.6	53.5	28.68
10	8.4	68.9	15.0
20	4.7	78.4	8.97

Table 2: Precision, recall and F1 score for evidential sentence selection based on k selected sentences in the final model.

We tried some tricks on dev to improve the F1, for example, for a five ranked candidate sentences, if the difference of score is greater than a certain value then cut off all subsequent numbers. We use the manual binary search to find the best difference 0.2 to value the F1 score is 0.52 and final accuracy of the system is 0.41.This is only an overfitting of the test data set and not applicable to the general fact checking dataset and thus we did not submit this result.

3.2 The overall system

Our final results on test set are shown in Table 3, which descended a little comparing with the result in the dev set. The possible reason is that our parameter settings such as the weight of the lookup have been overfitted to the dev set. Both label accuracy and sentence F1 score marginally better than benchmark,

Label Accuracy	42.62%
Sentence Precision	21.65%
Sentence Recall	48.18%
Sentence F1	29.87%
Docu Recall	53.48%
Docu Precision	36.9%
Docu F1	43.67%

Table 3: Final result on test set

4 Error analysis

As we test the performance of AllenNLP textual entailment by directly computing the entailment scores between the claims and the corresponding evidence the training dataset gives and the accuracy is 0.76, the system performance is limited by whether the document and sentence retrieval models find the right evidence. To better assess the performance of the system, we performed a manual error analysis on the causes respectively for 1.failing to return the 5 right candidate sentences evidence 2.wrongly identifying the right evidence from the candidates 3.obtaining incorrect relation between the claim and the right evidence by the entailment module.

To find the causes for 1, we sampled 500 of the return in which we compared each claim, its right evidence and candidate sentences. Of these, we found three typical errors. The first error is name disambiguation including name repetition, say several people’s name are ‘Ann Richards’ leading to a mismatch of the right person’s wiki entry. Another name disambiguation is target name entity is part of candidate entity, for example, ‘Mohra’ in a claim refers to a film name but candidates contain the names ‘Miana Mohra’ and ‘Mohra Sandhu’.

The second error is that the system will prefer the sentence which contains duplicates of the words in the claim than synonyms/antonyms, hypernyms/hyponyms and holonyms/meronyms of the claim words. A claim, for example, contains ‘advertising’ and ‘sound-based’, and our system gives a higher score to the sentence containing two ‘advertising’ than the sentence contains ‘advertising’ and ‘audio’, in the case where the other parts of these two are basically the same.

We also found that there is this condition when a claim needs two evidential sentences, say, the claim, ‘PERSON did A(non-NER)’, supported by ‘PERSON did B(NER)’ and ‘A is a type of B(non-NER)’, and the system only found the former evidential sentence as in our design NER has more weight than non-NER nouns. For instance, inputting the claim ‘Drake Bell (PERSON) put out an extended play(non-NER)...’, the system only found ‘Drake Bell release Reminder...’ and other sentences contain this PERSON entity with high TF-IDF score but cannot find the other evidence ‘ Reminder is an extended play...’. Same case happened in other NER. Our further assumption is to first leverage the whole dataset to train neural networks to get the correlation between words. As such, the system will consider the strong correlation score between ‘Reminder’ and phase ‘extended play’ thus improve evidence hit rate.

As for the cause of 2, due to the nature of low precision in the five-candidate mechanism in the case where the vast majority of claims have only one evidential sentence, if we want to improve F1, our system must truncate candidates precisely. It is not surprising that our system always gives the highest TF-IDF ranking to quite short sentence if there is one among 5 candidates. Quite often however, the very short sentence does not entail complete evidence and thus is often not the right evidence. This causes the remaining low precision when we cut off the candidates through a fixed difference, say delete all candidates behind one if it exceeds the subsequent by 20 scores. Hence, our future work is to adjust the parameters of document length in TF-IDF and add more features to jointly decide the ranking.

Finally, we find a more state-of-art alternative open sourcing pre-training technology, Bert, which can be employed to build entailment system and substantially improve the accuracy [5]. According to the new statistics from Google, refreshed the records of 11 diverse Natural Language Understanding (NLU) tasks, among which Multi-Genre Natural Language Inference (MultiNLI) achieves 86.7 accuracy. Our next step is to systematically learn the principles of Bert including Transformer network and use Cloud TPU to pre-training the model.

5 Conclusion

Although the competition ends, we will continue to experiment and tweak our sentence retrieval module to improve the sentence recall thus improving the final accuracy. We believe when we put the inspirations from the above error analysis into experiment, the performance of our system will have a qualitative leap. In the evidence retrieval part, observing most of error retrieval, analysing the cause of the errors, extracting the optimization scheme and modify the relevant weights is a long and recurrent process, 1% increase tends to result in an overall improvement in final accuracy of the system. With the advent of Bert and Cloud TPU, there will be a huge breakthrough in automatic fact verification system, which is what we aim at. Even with correct and authoritative evidence, the relation between claim and evidence is complex and thus still a long-existing open problem.

References

[1]T. WANG, Q. ZHU and S. WANG, *"Fact Statements Verification Based on Semantic Similarity",* Chinese Journal of Computers, vol. 36, no. 8, pp. 1668-1681, 2014. Available: 10.3724/sp.j.1016.2013.01668.

[2]Loper, E., & Bird, S. 2002. NLTK: the natural language toolkit. arXiv preprint cs/0205028.

[3]E. Hatcher, O. Gospodnetic and M. McCandless, Lucene in Action. Saintmpford: Manning Publications Co, 2010.

[4]M. Gardner and J. Grus*, "AllenNLP: A Deep Semantic Natural Language Processing Platform",* 2019. [Accessed 29 May 2019].

[5]J. Devlin, M.W. Chang, K. Lee, & K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding.