COMP90042 Project Report
AutomaticFactVerification

Rongxiao Liu (927694), Jiazhen Hu (971800)
Codalab: rongxiaol, saaltfiish
Team name: reallysaltyfish

1. Introduction

In the age of information explosion, abundant information becomes available for us. However, false information can mislead readers easily. Therefore, it is necessary to implement an automatic fact verification system to check authenticity of information. Verification is also important in other domains such as online news[1]. This report will introduce the methods we use to build such a system, show experiment statistics and present error analysis with ways on further improvement.

2. Methods

In this project, the system can be divided into 3 parts including document retrieval, sentence selection and claim verification.

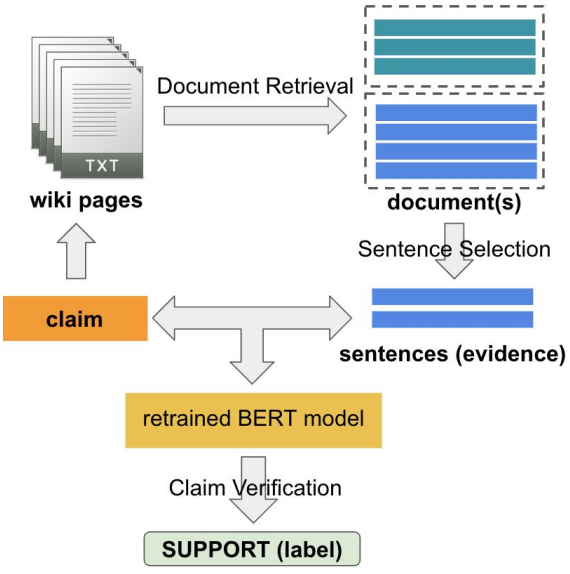

Fig.1 Overall Processes

2.1 Document Retrieval

(1) Named Entity Recognition

The most important part of a claim is often the entities. We find 2 kinds of entities in a single claim, First Entity and Rest Entity. First Entity often appears at the start of a claim as the subject, before the first verb, while Rest Entity appears after the first verb. A typical Rest Entity follows the pattern that all words within it has the first letter(s) of either itself or surrounding word on both sides capitalized. We extract every longest word sequences that follows this pattern among the words after the first verb as a Rest Entity. We treat all words that appear before the first verb as a raw First Entity and then refine it if the length exceeds a given length threshold. Refining a long entity is to regard it as a new claim and do Rest Entity parsing, as described above.

(2) Document Selection

After entities are extracted from a claim, we use this information to search for the most relevant documents. We find that the topic of a document can represent itself much better than all the words in that document. Therefor, instead of doing tf-idf query among the whole wiki corpus, we only compare the edit distance between an entity string and a document topic string, and select those with the distance value less than a given difference threshold. We have noticed that wiki use parenthesized content to clarify ambiguity of a topic.

These contents, with the parentheses, are removed before comparing the edit distance, and restored if the document is regarded as relevant, for further searching in sentence selection.

2.2 Sentence Selection

(1) Corpus Indexing

Considering that corpus size may scale out in the future, the whole corpus should never be read into memory. We have sorted all documents with string order of their topics, and built an index to map a particular topic to the exact wiki file containing it, within the sorted wiki corpus. Sorting is implemented in a streaming way. We first sorted each wiki file respectively, and adjusted the margin of adjacent files to ensure topic ranges of them don't intersect. The whole indexing can be done under 30 minutes.

(2) Sentence Selection

When the content of each relevant document is retrieved, we compare sentence relevance between the claim and every sentence in the document. To compute semantic similarity of the claim and a given sentence, we first extract all useful words in the 2 word sequence separately and divide them into 4 categories, including non-auxiliary verbs, nouns, adjectives + adverbs and others. We computed the sum of word similarity between the 4 word sequences of both sentence, normalize it with the length of every word sequence, and take a weighted sum of these 4 similarity values. Sentences with a similarity value over a given threshold will be regarded as relevant.

(3) Link Search

We have noticed some cases that new named entities appear in evidences. These evidences should be treated as a new claim, with documents and sentences relevant to it retrieved to help verify the original claim in combination. To avoid introducing non-related new claims and entities, we only do link search on relevant sentences with similarity value over 0.8.

2.3 Claim Verification

(1) Model Choosing

There are two models (Elmo and BERT) achieved a great success in the last year. Finally, we chose BERT-Base (Cased) [2] to do retraining. The BERT model and codes are downloaded from Github and modified to suit our dataset.

(2) Dataset Pre-processing

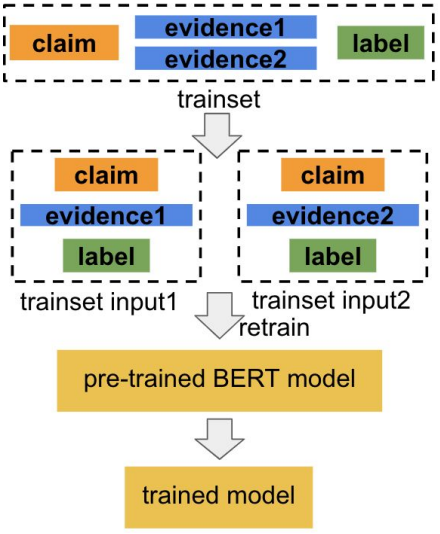

Fig.2 Model Training

We noticed that the evidences with "NOT ENOUGH INFORMATION" in given trainset and devset are empty. While we need to generate examples of evidence with this label, we add our selected sentences that are most related to claim as evidence.

Furthermore, we replaced all pronoun (she, he, it, they, her, his, its, their...) in the sentence with topic content because name entities are more meaningful than pronoun.

(3) Model Training

Because BERT is a general-purpose "language understanding" model on a large text corpus and may not suit to our wiki

dataset well, we choose to retrain the model using given trainset.
We seperate the sentences of evidence and the input for each example in trainset becomes (sentence1, claim), (sentence2, claim), (sentence3, claim)...
Considering that our training set is very large, we decide to run our code on colab (using free GPU/TPU). After around 10 hours running on GPU or 40 minutes on TPU, training, evaluating and predicting stages all finished.

(4) Label Prediction

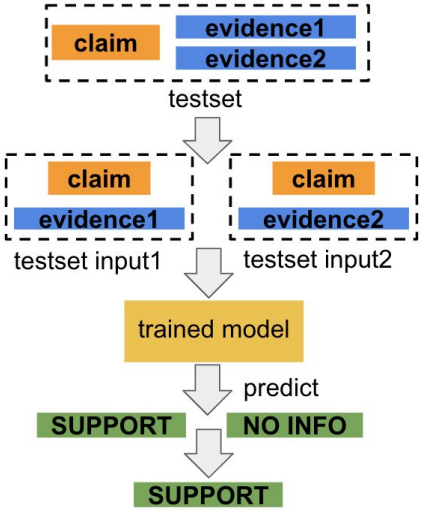

Fig.3 Label Prediction

The examples in testset are inputted in the same way as trainset. Finally, we get labels for each evidence sentence. If there are all "NOT ENOUGH INFORMATION" labels for sentences, then the final label for this claim will be "NOT ENOUGH INFORMATION". Otherwise, we will choose the label with majority between "SUPPORT" and "REFUTES" (if equal, then "SUPPORT").

3. Experiments

3.1 Information Retrieval (IR)

Many factors can influence the final result, including NER, document retrieval, sentence selection and link search. We have constructed a non-link search dev set based on the original one, and evaluated entity extraction, document only retrieval and the final result of document and sentence. Statistic is shown in Tab.1.
As the result of Tab.1 shows, our system has a good performance on NER and document retrieval without link search. However, the score drops immediately when considering link search. Moreover, the document recall as well as sentence precision and recall also drops sharply no matter considering link search or not.

data	dev set with link search			dev set without link search		
score	precision	recall	F1	precision	recall	F1
named entity	69.68%	71.15%	70.41%	86.18%	93.80%	89.83%
document only	66.51%	68.93%	67.70%	80.69%	84.84%	82.71%
final document	81.79%	54.57%	65.46%	80.03%	64.67%	71.53%
final sentence	54.46%	45.31%	49.46%	48.99%	52.38%	50.63%

Tab.1 IR Evaluation

3.2 Natural Language Inference (NLI)

devset

baseline	33.33%
eval_accuracy	84.53%

test result (from codalab)

baseline	33.33%
test_accuracy	54.20%

Overall, it achieved remarkable performance on claim verification (21% higher than baseline 33.3%).

4. Error analysis

4.1 IR

Results in Tab.1 indicates that there are 2 main problems in IR, method of sentence selection, and requirements on link search.

(1) Sentence selection error

Apparently, sentence relevance involves not only semantic similarity, but also sentence structure similarity, which is not included in our system.

To further involve sentence structure into consideration, extraction of more sentence features such as positional information is essential. As BERT extracts sentence features in a comprehensive way, and gives us a good performance on NLI, we believe that it can also do well in sentence relevance scoring. Therefore, for further improvement on sentence selection, BERT might be a model to get start with.

(2) Link search error

Since the ability of our system on sentence selection is not competitive enough, it would be risky to do link search on too many evidences, because link searching a wrong evidence will introduce more wrong evidences.

The current sentence similarity score requirement for link search is 0.8. The high requirement is the cause of low recall of documents in dev set with link search. Only after our sentence selection becomes competitive enough, can we lower this threshold and introduce more sentences, to fix this error.

4.2 NLI

The error in this part mainly results from wrong evidence generated from the last step (sentence selection). In the sentence selection part, in order to increase F1, we decrease the number of returned evidence sentence. some “SUPPORT” and “REFUTES” claims receive empty evidence list, which means these claims will definitely be labelled as “NOT ENOUGH INFORMATION” because there is no evidence to judge their label. Furthermore, we only label claim as “NOT ENOUGH INFORMATION” when all its evidence sentences are “NOT ENOUGH INFORMATION”. Therefore, once there is a sentence labelled incorrectly, the entire claim will be labelled wrongly.

5. Conclusion

We have implemented an end-to-end automatic fact verification system. We’ve done the IR part by ourselves, and trained an NLI model based on BERT and our IR results. The final score is at a competitive level, but there still exists many problems in each step of the pipeline. For further improvement, it is necessary to introduce BERT for IR and train a stronger model for NLI when facing lack of information.

References

[1] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: a large-scale dataset for fact extraction and verification. arXiv preprint arXiv:1803.05355.

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[3] Hanselowski, A., Zhang, H., Li, Z., Sorokin, D., Schiller, B., Schulz, C., & Gurevych, I. (2018). UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification. arXiv preprint arXiv:1809.01479.