Natural Language Processing: *Automated Fact Checking For Climate Science Claims*

Student ID: 1165986

Abstract

The goal of this project is about automatically fact-checking for climate science related claims. A large number of evidence passages (knowledge source) were provided for trying to prove claims, then identify if claim has been supported by selected evidences. Through this way, can figure out incorrect claims which lead to a distortion of public opinion.
This report mainly focus on the process of dealing with this Natural Language Processing problem and its strategy selection.

1 Introduction

On this report, training, development, test datasets (contains 1228, 154, 153 data respectively) and evidence passages discussed is offered by Codalab competition `https://codalab.lisn.upsaclay.fr/competitions/12639`. Evidence passages contains 1,208,827 sentences, and there have no sequential relationship correlated. Training and development datasets contain "claim-text", "claim-label" and "evidences". Test dataset only contain "claim-text", used to predict "evidences" and "claim-label". The format of "evidences" is **id** list from evidence passages. "Claim-label" contains 4 levels: NOT-ENOUGH-INFO, REFUTES, DISPUTED, SUPPORTS. To predict "evidences" and "claim-label", break the problem into two part:

- **Similarity Part:** Find the most similar sentences in evidence passages.
- **Classification Part:** Given claim and its evidences, classify their relationship.

2 Prepossessing

This report mainly used StanfordCoreNlp library (Manning et al., 2014) to do preprocessing.

Data is provided as sentences independently, therefore, first step of preprocessing is word tokenization. The domain of this dataset is climate science, by a quick look of the data, except for letters and numbers, some token contains other identifiers, for instance, *ºF*, *nf-$_k$b* and *2-carboxyethylgermanium*, which are important on this task. Instead of just keeping letters and numbers like normal NLP preprocessing tasks, this scenario as a special case that only remove some common identifiers. Besides, stop-words offered information for some models, therefore, stop-words are not dropped initially.

It is important that evidence passages in a big number of sentences, dealing with Named Entity Recognition (NER) is costly and it would not been too much help on it. Pertained NER taggers can not recognized entities like CO2 which give a lot of information, they can only identify some organizations that do not need to pay more attention on for this project. Specially, a evidence sentence that contains more than two special characters which does not occur in training evidences, been removed and allocated with null string.

3 Similarity Model

On this stage, similarity model need to find similar sentences (at most 5 out of 1,208,827), and there are only 1228 training data. The base idea to identify top k sentences is use any model trying to calculate similarity between claim and evidences.

3.1 Prepare Dataset

Training a similarity model can not only train it with positive samples. Negative samples are also important. For instance, BERT Model (Devlin et al., 2019) random selects sentence as negative sample when pretraining it. Prepare a negative sentence that worth to learn might help to accelerate training process. That makes model learn sentence semantics better, rather than training without negative sample which only get a words level similarity. Therefore, trying to find some word level similar sentences and use them as negative samples. Strategy used in this project is to find key information in claims and used them to search any evidence sentence that contain similar information as candidates.

Finding the `key words` on claims by using POS tagger to identify any tag start with NN, VB or JJ. Those words contain the core information on a sentence. Train a Gensim (Řehůřek and Sojka, 2010) Word2Vec Model (window=10, min-count=2 for 5 epoch) with stemmed sentences, to find the top 10 similar token appeared in evidence passages. An Information Retrieval technique **inverted index** (Manning et al., 2008) been used here, to quickly retrieve sentences using token.

After each `key word` stemmed, Word2Vec Model offers 10 similar tokens. Those tokens could be considered to express similar information. Trying to select out at least 50 similar sentences from evidence passages using `key words` per a claim text. Since a claim text usually have more than one `key words`, and each of them linked to a bunch of evidences, combination idea bring here to select. Initialised select evidences contains more than k (initial with 5 and gradually decrease) key information (any combination of more than k are accepted).

At training step, model would randomly select sentence from candidate list as negative sample for every true evidence.

3.2 Model Selection

Common machine learning models, bag-of-words and tf-idf can be used to calculate similarity. Those methods are count based, which only based on words, could not capture the syntactic relationship between words and sentence structure. *E.g.* for a claim describes about atmosphere, use count based model can only give a high similarity to evidences contain atmospheric, but evidences about stratosphere would not be searched. Therefore, count-based model may not as a good selection.

To proving that, **BM25** (Robertson and Zaragoza, 2009) algorithm has been used based on key features distilled by tf-idf vector. After prepossessing part discussed, count based method have one more step – stemming. Results shown, almost no true evidence at the top-five similar evidences, and true evidences position on more than 10,000 on average.

Rather than common machine learning model, Neural Network can explain more complex information by using word embedding instead of sparse matrix, and it can generalise unseen sequences better. This dataset only contains textual data. Thus, fine-tuning pretrained models might be a good choice. BERT (Devlin et al., 2019) used **Word Piece** - subwords tokenization technique. This dataset contains some domain vocab, therefore, learning ability of unseen tokens is an important improvement. Original BERT model embedding

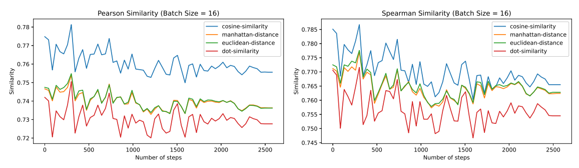


Figure 1: Sequence-BERT Similarity

tokens sequential, and each token with high dimensional representation. Although use padding to fix sentence length different problem, searching through entire evidences corpus is costly.

Sentence-BERT (Reimers and Gurevych, 2019) structure use 1-d vector to represent a sentence. It is a sentences level representation rather than token level like BERT. S-BERT have relatively low computational cost when calculating the similarity between claims and evidences.

3.2.1 Retrieve Model

Embedding part of *SentenceTransformer* class initialised with pretrained "bert-base-uncased" model. Because 99% of claims and evidences are less than 50 words, the max sequence length sets to 50. Loss calculated by class *MultipleNegativesRankingLoss*.

epoch	10		20		30	
k	train	dev	train	dev	train	dev
5	.361	.206	.576	.224	.603	.207
10	.500	.285	.748	.291	.777	.270
15	.581	.328	.826	.336	.845	.334
20	.635	.375	.869	.375	.883	.374

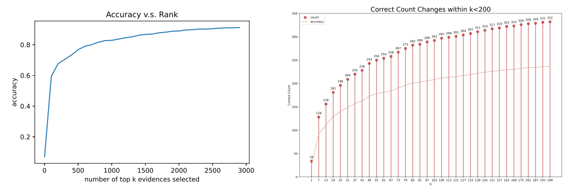
Table 1: Sentence-BERT performance (batch-seize=16, num-steps-per-epoch=274, warmup=0.05)

Prepared dataset is used to train on this model, and trying to find top k similar sentences from evidence corpus. The similarity scores figure 1 identifies changes during training. Plot indicates that cosine similarity gives higher score than other calculation methods. Table 1 shows correctness of top k evidences. By comparing epoch 20 and 30, accuacy of dev set decreased and training set increased a lot. So the epoch 20 selected to use.

On chosen model, figure 2 (a) shows that almost all correct evidences are ranking at top 0.2%. It is better than using count based method mentioned above, but still need to improve.

3.2.2 Re-rank Model

Goal of similarity model is to search for most similar evidences on top 5 rather than top 3000. However, results shows training more would lead to over-fitting problem. Matsubara et al. (2020) deal with QA task mentioned that re-rank can correctness retrieved documents, which can apply on this data. Treat S-BERT ranking model as retrieving part, and train a more accurate model (*CrossEncoder* structure, loss used *CECorrelationEvaluator*) only focusing on retrieved sentences.



(a) k <3000 (b) k <200

Figure 2: Sentence-BERT accuracy v.s. rank k.

By restriction of computational cost, re-ranking 3000 sentences for each claim is expensive. Plot on figure 2 (b) is about changes in detail for k <200. It shows that growth of accuracy similar to log function which can not find a good position to cut. Hence, some training strategies are tried:

(1) Top retrieved and true evidences with 100 in total used to train model. Labeling retrieved similar evidences as 0.3, true evidences distinguish NOT-ENOUGH-INFO and the rest labels by labeling them as 0.6 and 0.9.

(2) Each claim randomly selects negative samples from retrieved in a ratio of 1:4×nun-of-true-evidences from 100. Label retrieved similar and true evidences as 0.3 and 0.9 respectively.

(3) Randomly select from 3000, and the rest setting same as (2).

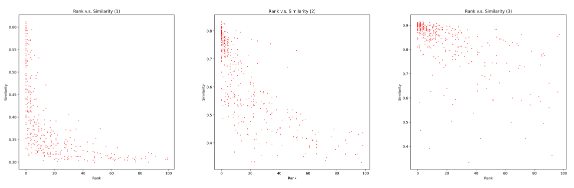


Figure 3: Re-rank True evidences distribution (epoch=1, batch-seize=32, warmup=0.05)

Distribution of correct sentences for each strategy show on figure 6. Strategy (2) have more true evidences on higher rank (lower k). In general, re-rank model has improved over retrieve model.

3.2.3 Multiple Choice by Classification

Through checking accuracy of retrieval part, find that score has been impacted by predicted some incorrect evidences. After re-rank model, add a classification model (*CrossEncoder* structure, loss used *CEBinaryClassificationEvaluator*) to distinguish correct and incorrect evidence rather than similarity. Only use top 20 re-rank evidences to train it.

4 Classification Model

Based on evidences passage selected by similarity model, use classification model to find relationship between claim and evidences be found.

4.1 Prepare Data

Distribution of labels on training set is imbalanced. Notice that labels of relationship are NOT-ENOUGH-INFO, REFUTES, DISPUTED and SUPPORTS, which contain generalised information from more than one evidences. Classify single evidence sentence can not represent whole passage. *E.g.* a sentence supports claim well and next sentence refutes it. On this case, single sentence prediction cannot capture relationship cross sentences. Therefore, classification model,need to put evidence passage sentences together.

4.2 Model Selection

Naive Bayes algorithm have independence assumption, but for natural language problems, correlation between words is high. Logistic Regression require a large number of data to get a good performance, in training set only have 1228 data. Support Vector Machines awkward on multi-class classification problem, re-framing should be used for solving it. Decision Tree can solve non-linear-separable problem. Therefore, re-framing Support Vector Machines and Random Forest (a bagging of Decision Tree) are applied.

4.2.1 Support Vector Machines (SVM)

Preprocessing part of SVM added stemming step. Afterwards, calculate tf-idf vector (training data and evidences passages to fit), and use PCA (only training data to fit) to select 200 features.

Original SVM framework only support for *ovo* and *ovr* algorithm, but both of them not appropriate for imbalanced classification problem. Based on distribution of labels, trying to re-framing a sequential SVM model. Classify SUPPORT, NO-ENOUGH-INFO and the rest every two classes sequentially, and result shows 0.8159 on training set and 0.4285 on development set (using correct evidences).

4.2.2 Random Forest

Used the same data as SVM model, trying to compare those two models. Training model with class weight $\frac{max-class-count}{class-count}$. Performances of Random Forest is highly correlated to hyper-parameters are chosen. On figure 4 left shows classification performance for different hyper-parameters, and max-depth=8 with good accuracy and low variance on average. True and predicted labels on dev set shown on the figure 4 right, which shows 'REFUTES' and 'DISPUTED' are predict badly. Training set and development set (use correct evidences) accuracy are 0.8338 and 0.4416 respectively.

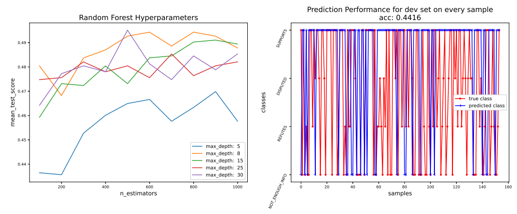
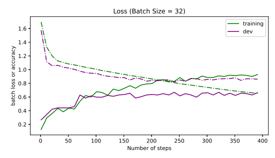


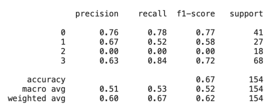
Figure 4: Random Forest parameters & predictions (cv=5)

4.2.3 Fine-Tuning BERT

SVM and Random Forest model can not deal with this classification problem well, even if true evidences are given. Consider to apply neuron network model on it. Since S-BERT was used for calculate sentence similarity, BERT embedding layer within S-BERT already learned some information within climate domain. Use BERT embedding layer of Sentence-BERT to initialise embedding part of classification model. Simply back propagating loss (use AdamW with lr=2e-5, loss *BCEWithLogitsLoss* with weight=$\frac{max-class-count}{class-count}$). Dev set accuracy on figure 5 (a) keeping around 0.65 after 240 steps, and its loss gradually increase after 250 step. Hence, step 240 might be the best model.



(a) Loss and accuracy of train & dev sets change by training step



```
              precision    recall  f1-score   support

           0       0.76      0.78      0.77        41
           1       0.67      0.52      0.58        27
           2       0.00      0.00      0.00        18
           3       0.63      0.84      0.72        68

    accuracy                           0.67       154
   macro avg       0.51      0.53      0.52       154
weighted avg       0.60      0.67      0.62       154
```

(b) classification report on dev for step 240

Figure 5: Fine-Tuning Classification BERT (batch-size=32, steps=400)

4.3 Analysis

Figure 5 (b) output performance of development set (use true evidences), it show that Fine-Tuning BERT lose ability to identify label 2 – DISPUTED. Similar cases also happened on SVM and Random Forest models. Trying to figure out this problem by plotting distribution of labels. One explainable reason is shown by figure 6. One claim can be explained by 1-5 evidences, and it is difficult to learn well for label that with little group samples. Sentences carrying more complex information are more likely to be predicted as NOT-ENOUGH-INFO.

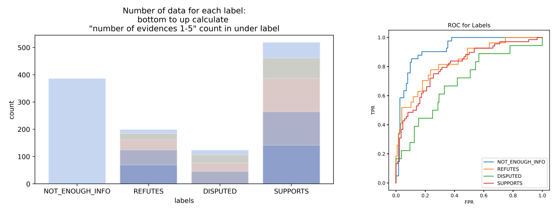


Figure 6: Class Imbalance

5 Findings

Classification models discussed above did not contains impact from incorrect evidences. Repeats same steps described (Fine-Tuning BERT) above to apply on results from retrieve, re-rank and multi-choice models. Results shown on Table 2.

		retrieve	rerank	multi-choice
eval-dev	Retrieval	.1808	.2048	.2178
	Classification	.4935	.4415	.5000
Codalab-test	Retrieval	.1280	.2265	.2149
	Classification	.4605	.4342	.4868

Table 2: Results

After first retrieval, difference between dev and test is huge. Re-rank model fitted more generalized key information. Performance on test set gives a 30% increase on retrieving. Multi-choice model reduce variance between dev and train set, which with a high average accuracy stably.

6 Furture Study

Improvements on separately predict evidences and claim-label might be restricted. Use a model can treat them correlated and train simultaneously. To applying on this problem might need more supervised data.

7 Conclusion

Re-rank model helps explaining semantic information a lot in specific domain. Text information is noisy, and difficult to distinguish special tokens within domain and noise identifiers.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. 2020. Reranking for efficient transformer-based answer selection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1577–1580, New York, NY, USA. Association for Computing Machinery.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.