Automated Fact Checking For Climate Science Claims

Nofaldi Putranto

1 Introduction

Spreading information is very massive because of technology. Hence, a tool that is used to check the correction of information is important. Automated Fact-Checking is a tool to assess whether the claims by given evidence are true (Zeng et al., 2021).

The report aims to establish automated fact-checking which can find related evidences given claims and predict the relationship between evidences and claims. this project implements information retrieval and machine learning method.

1.1 Data

There are two main components consisting of the data, including a list of claims and a corpus with a large number of evidence passages ('knowledge source'). Details of the dataset are shown in Table 1. Compared to the amount of data for each dataset, the evidence passages source is huge. Except for the Evidence Passages and Test dataset, claims and evidence in Training and Development dataset are classified into 4 relations SUPPORTS, REFUTES, NOT_ENOUGH_INFO, and DISPUTED. The distribution of each class in the Training and Development datasets is demonstrated in Table 2. It is obvious that SUPPORTS take over 40% proportion in both 87 training and development datasets. This arises two main concerns (1) How to effectively retrieve the evidence passages from the evidence corpus with 1.2 million texts; (2) How to deal with the imbalance datasets.

Dataset	Training	Dev	Test	Evidence passing
Number of Data	1228	154	153	1208827

Table 1: Number of data in each dataset

Label Distribution	Training	Development
SUPPORTS	42.30%	44.20%
NOT_ENOUGH_INFO	31.40%	26.60%
REFUTES	16.20%	17.50%
DISPUTED	10.10%	11.50%

Table 2: Label Distribution

2 Methodology

Automated Fact Checking for this project consists of 2 different tasks. In the first task, students need to identify related pairs of evidence and claim. In the second task, students need to identify the relationship between the set of predicted evidence and a claim.

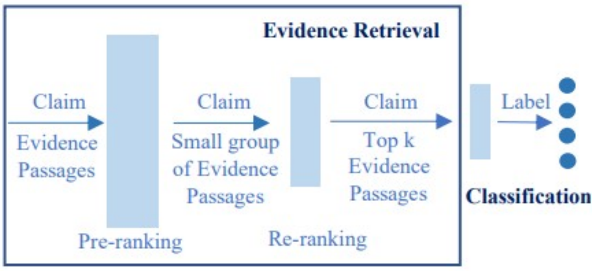

Figure 1: Methodology Overview

2.1 Information Retrieval

Vlachos and Riedel (2014) propose pair checking using a supervised approach (Vlachos and Riedel, 2014). This project can use Vlachos and Riedel method because the project has labeled combination sets of evidence and claims that are considered as pairs. The method will treat this task as binary classification. The input of this model is a combination of an evidence sentence and a claim sentence. The method will label 1 for the sentence and claim which are considered pairs in training data and the method consider label 0 for the sentence and claim which are not pairs.

However, the project has imbalance label because it has more unpair combinations than pair combinations of sentences and claims. For every claim, this method selects the top 100 most related unpair evidence to decrease the number of unpair combinations. The methods use 3 different methods to retrieve related evidence. The methods are BM25, Cosine similarity using TF-IDF embedding, and dot product between TF-IDF embedding.

First, the method does the pre-processing to establish TF-IDF embedding. We make all sentences in lowercase format and remove stopwords, non-alphabetic words, and punctuation in all sentences. the method also does lemmatization for every token because lemmatization is better to do information retrieval in sentences compared to stemming (Boban et al., 2020).

The method fits and transforms all evidence sentences into TF-IDF to get a word embedding matrix. word embedding matrix will consist of rows representing all evidence and column representing all word embedding. using the previous TF-IDF class, this method transforms the claim sentence to TF-IDF. The dot product method calculates the dot product between the claim's word embedding vector and the evidence's word embedding matrix then it takes the top 100 evidence with the largest dot product value. Cosine similarity uses a similar method but instead of taking the dot product, it calculates the cosine similarity between the claim and evidences. Meanwhile, BM25 establishes evidence word embedding using Roberta's pre-train model and uses the claim sentence as the query to get the most top 100 related evidence.

After establishing pair of sentences and evidence, the method uses two methods to predict labels as pair or not pair. The method uses fine-tuning on Bert-uncase pre-train model and Roberta-uncase pre-train model.

2.2 Predicting Evidence and Claim Relationship

The project also has a set of labels explaining the relationship between evidence and claim so it can treat this task as a multi-class classification. The method uses fine-tuning on Bert-uncase pre-train model and distilroberta-base pre-train model same as predicting the label pair. The project labels the relationship between claim and evidence as 0:DISPUTED, 1:SUPPORTS, 2:REFUTES, and 3:NOT ENOUGH INFORMATION.

3 Baseline

I also propose the baseline method that is easy to compute and doesn't require a lot of computational power. For information retrieval, The baseline method only takes the top 5 pieces of evidence for each claim using the dot product with Tf-idf embedding. For predicting the relationship between evidence and claim, I only use decision tree and use the Tf-idf embedding as input.

4 Evaluation

This project uses a dev-dataset for evaluation. This project also uses 2 evaluation metrics. The F1 scores metric is used to evaluate part 1 and the accuracy metric is used to evaluate part 2.

4.1 Selecting Number of Top Evidence

At first, the method considers multiple numbers to retrieve the evidence such as the top 50, top 100, and top 1000.

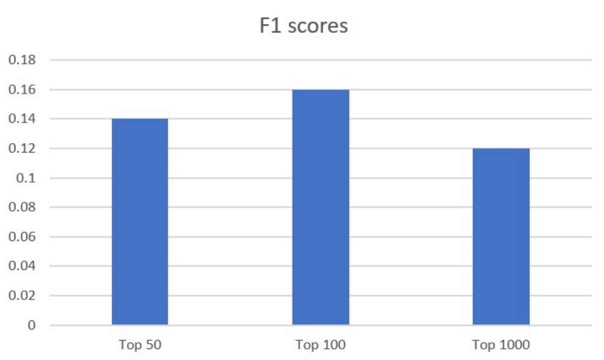

Figure 2: F1 result in retrieve word

From the figure, the Top 100 is the best compared to the top 50 and 1000 because the top 50 doesn't have enough evidence to compare, meanwhile, the top 1000 increase the imbalance in the dataset.

4.2 Method for Evidence Retrieval

The method evaluates based on percentages of real evidence that are retrieved during the retrieval process. For example, if a claim has 5 pieces of evidence in the training dataset and the method only retrieves 2 pieces of evidence from 5 then the score will be 40%.

the table shows that the dot product is slightly better than cosine similarity. The dot product

Method	Retrieval in Percent
BM25	25,4%
Cosine Similarity	34,3%
Dot Product	35,5%

Table 3: Evaluation Retrieval Method

between tf-idf vector and matrix shows good results because it captures the similarity between sentences by considering more unique words.

4.3 Binary Classification

The project uses 2 different pre-train models to predict whether combinations of claim sentences and evidence sentences are paired or not.

Pre-train model	F1 value
Bert	0.16
Roberta	0.07
Baseline	0.08

Table 4: Evaluation Pre-train Model

Bert shows slightly better results compared to Roberta. Roberta's result shows the worst performance than the baseline result. This method only trains 4 epochs for each pre-train model because after 4 epoch model doesn't show any improvement.

4.4 Multiclass Classification

The method also uses 2 different pre-train models to learn the relationship between the claim and predicted evidence. The evaluation uses accuracy to assess both models.

Pre-train model	Accuracy
Bert	0.49
Roberta	0.55
Baseline	0.12

Table 5: Evaluation Relationship

The result from Table 3 shows a different result from Table 2. The Roberta pre-train model shows better accuracy than Bert model. Both models show better performance results compared to the baseline.

5 Conclusion, limitation, and Improvement

There are several methods that I use for submitting a final model. For information retrieval, I use dot-product between tf-idf embedding. For predicting pair of evidence and claim, I use fine-tuning Bert pre-train model. For predicting the relationship between evidence and claim, I use fine-tuning Roberta pre-train model. The final method shows 0.16 F1 and 0.55 accuracy in the dev-dataset. When I use this method with the test dataset, The result is not that different. the result from test data shows 0.12 F1 and 0.44 accuracy. It shows that the method doesn't over-fit in the dev-dataset.

The limitation of this assignment is the lack of computer power. GPU is needed to fine-tune Bert and Roberta. Even, GPU can access using google colab but it has a time limitation to use it.

To improve information retrieval, I suggest fine-tuning with other pre-train models and also using fine-tuning from community pre-train models. I also suggest using part of the speech tag because lecture slide 5 stated that the noun of a sentence is important to information retrieval.

References

Ivan Boban, Alen Doko, and Sven Gotovac. 2020. Sentence retrieval using stemming and lemmatization with different length of the queries. *Advances in Science, Technology and Engineering Systems Journal*, 5(3):349–354.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. pages 18–22.

Xia Zeng, Amani Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15.