Sentiment Analysis of Citation

Fan Liang
Tuo Wu

Abstract

In this project, we have established a citation sentiment analysis, we have tried multiple different models and applied different preprocessing and tokenizer skills, even though the final score is not quite significant, we still made an improvement of previous works and we have some new ideas that could be developed in the future.

1 Objective

We are interested in citation sentiment analysis; it is a minority topic, but we think if we can successfully train a citation sentiment analysis model. It is quite useful to analyze large scientific scholar databases. We plan to set our model based on N-Gram, but we are open to trying other methods like TFIDF and add additional data preprocessing techniques to the N-Gram base.

2 Dataset

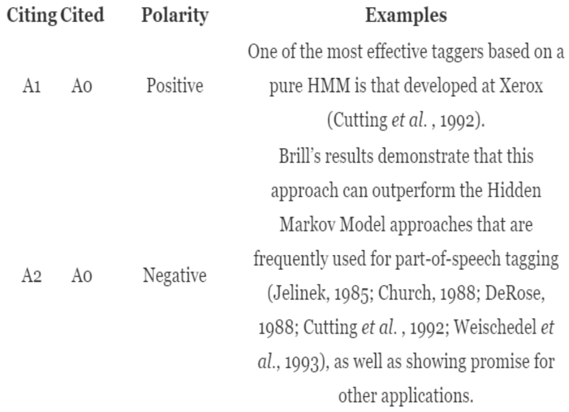
Our data is from ACL Anthology. There are 8736 rows of records in our dataset. We have five variables. They are source paper, target paper, sentiment, label, and Text. Sentiment consists of positive, negative, and objective. There are three numbers in the label column corresponding to three different, they are 0, 1, 2. Text is the content of the citation.

2.1 Data Challenge

There are two main challenges in this project. The first one is imbalanced data. When we check the distribution of sentiment, we find there are 7627 objective sentiment, 829 positive sentiment and 280 negative sentiment. The imbalanced data might give us a pretty good overall predictive result, but it will have worse performance in predicting negative sentiment and positive sentiment. The second one is not strong enough emotion. We use python package Vader sentiment to analyze the emotion component of the text. For example, when we input a sentence this phone is super cool, we find 67.4% of this sentence is positive, 0% is negative and 32.6% is neutral. Then we use the same method to analyze the text which is labeled as negative in our dataset, but 27% of text has a larger negative component than positive component which means negative text is not that negative. Also, we did the same analysis for positive text, we find 67% of text has larger positive component than negative component. As a result, we think the emotion of text is not strong enough.

3 Background of approach

With the development of Natural Language Process techniques, more and more researchers focus on applying these techniques to sentiment analysis for different text genres in the past decades; however, there are fewer researchers placing their emphasis on extraction of perspective from scientific literature, so that we consider citation sentiment analysis as our topic. Also, citation sentiment analysis plays a significant role in analyzing research. Researchers always need to analyze multiple papers that are related to their topic at the same time, but this process is time-consuming because there are too many papers. In order to improve efficiency, researchers can use citation sentiment analysis to quickly understand people’s opinions, emotions towards certain papers, and plot scientific idea flow. For example, when we check the following picture, “we can see one of the ideas introduced in paper A0 is Hidden Markov Model (HMM) based part-of-speech (POS) tagging, which has been referenced positively in paper A1. In paper A2, however, a better approach was brought up making the idea (HMM based POS) in paper A0 negative. It shows how sentiment analysis of citation plays an important role in plotting scientific idea flow and shows how the evolution of scientific ideas happened when old ideas are replaced by new ones.” (Liu, 2017)



Citing	Cited	Polarity	Examples
A1	A0	Positive	One of the most effective taggers based on a pure HMM is that developed at Xerox (Cutting *et al.* , 1992).
A2	A0	Negative	Brill's results demonstrate that this approach can outperform the Hidden Markov Model approaches that are frequently used for part-of-speech tagging (Jelinek, 1985; Church, 1988; DeRose, 1988; Cutting *et al.* , 1992; Weischedel *et al.*, 1993), as well as showing promise for other applications.

Table 1: Examples of positive and negative citations.

We also read some papers about applying the Word2vec technique to do sentiment analysis of citation, such as Sentiment Analysis of Citations Using Word2vec. However, we found it has very worse result when we apply them in our data. Then we choose to use word embedding instead of Word2vec.

4 Mythology

In the project, we used multiple different method to do data preprocessing and predictions. We are going to introduce them in the following contents.

4.1 Preprocessing Techniques

In this project, we have tried two different preprocessing skills, one is the normal preprocessing method, we remove stop-words, punctuations, numbers, lemme, and also we break the <word_structure> and convert the citation to a citation token <CIT>. Another way we tried is leaving the citation token <CIT> and then we only leave Verb, Adverb, Noun, and Adjective in each sentence. All these parts could be done in the NLTK package and the Spacy package. The performance of these two preprocessing techniques tested in several different models, and we believe the second preprocessing technique is performing better than the first one.

4.2 Text Data Augmentation

Since our dataset is super imbalanced, we implemented text augmentation techniques on training set after preprocessing. This section is important because of the original dataset we have skewed heavily to the objective class. If we cannot give our models more feature information about the other two classes, it is super difficult to let the model predict more correct positive and negative cases.

The first method we tried is Oversampling, which is an implementation of Bootstrap, however, as it sampled many positive and negative cases, it also created redundancy since Bootstrap is just a method that creates duplicates, and we easily get overfitting during our modeling process.

The second method we tried is called Smote, it is a very popular text augmentation technique nowadays, however, it performed terribly.

The third technique we tried is to build a simulator, and we believe this technique could be developed in the future in order to optimize the performance of text augmentation. The main idea is the implementation of Markov Chain, which we could learn from existing text and simulate new texts that never appeared, even though the new simulated sentences might not be a sentence contains meaning, but it could express key features that these original texts contained. Below attached are several examples we simulated from negative class:

○ supervision bootstrap algorithm cluster less especially less successful study here class other maximum entropy model limit lacks

○ current incomplete set instance much simple synchronous grammar present paper intend replacement sentence based method

○ fill gap average training set extracted rule must deal subset problematic aspect pure knowledgebase

We can see these sentences are false in grammar, but they contain the keyword that expressing the negative sentiment in which we can give our models more information.

4.3 Vectorized

In this project, we have tried several different vectorizers and they all have different performances. The two basic vectorizers are the n-gram and tf-idf, they are quite useful in most of the NLP tasks, we applied these two techniques when we were using basic classification models, and after test, we think 1-3 gram works quite good compare to other methods. We also tried embedding from tensorflow.keras when we am using the deep learning models, there are many different ways to create the vector, even though they have the difference, but since the dataset is super imbalanced, we did not see huge gaps between each method.

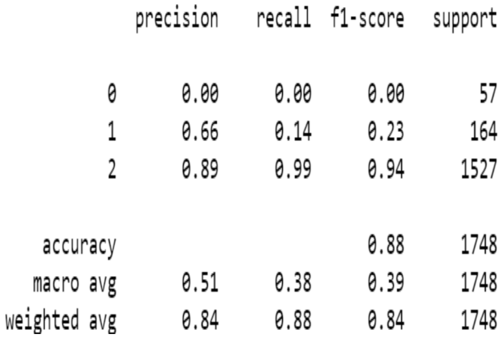
4.4 Predictive models

We tried different classification models, such as logistic regression, random forest and SVM. Besides that, we also tried convolutional neural network and Bert.

5 Results

For classification models, we tried Logistica Regression, Random Forest and SVM, we tested two different preprocessing methods and two different text data augmentation method on all models.

Random Forest:

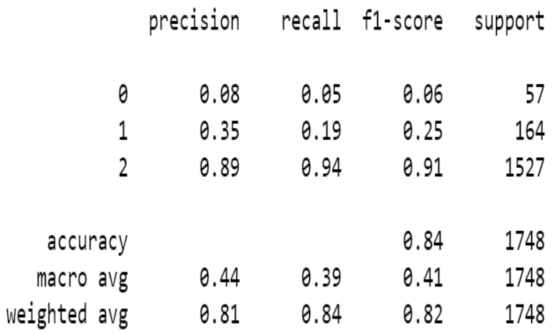


```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        57
           1       0.66      0.14      0.23       164
           2       0.89      0.99      0.94      1527

    accuracy                           0.88      1748
   macro avg       0.51      0.38      0.39      1748
weighted avg       0.84      0.88      0.84      1748
```

Naïve Bayes:

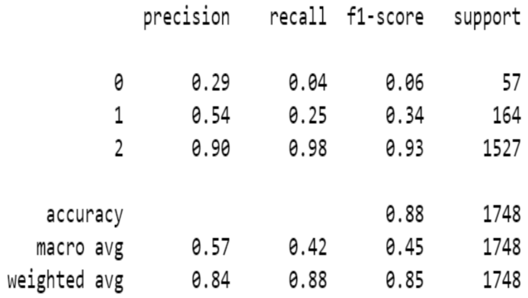


```
              precision    recall  f1-score   support

           0       0.08      0.05      0.06        57
           1       0.35      0.19      0.25       164
           2       0.89      0.94      0.91      1527

    accuracy                           0.84      1748
   macro avg       0.44      0.39      0.41      1748
weighted avg       0.81      0.84      0.82      1748
```

SVM:

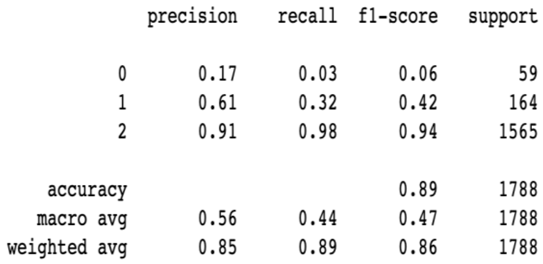


```
              precision    recall  f1-score   support

           0       0.29      0.04      0.06        57
           1       0.54      0.25      0.34       164
           2       0.90      0.98      0.93      1527

    accuracy                           0.88      1748
   macro avg       0.57      0.42      0.45      1748
weighted avg       0.84      0.88      0.85      1748
```

We tested two different preprocessing methods and two different text data augmentation method on all models, the best model we got is a Logistic Regression model using first preprocessing method and augmented data with the simulator.



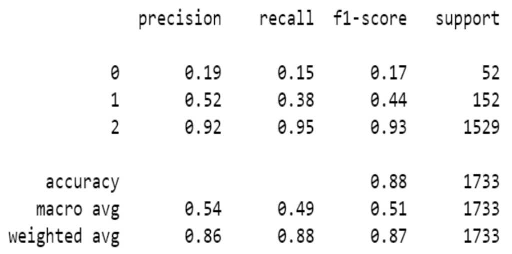
```
              precision    recall  f1-score   support

           0       0.17      0.03      0.06        59
           1       0.61      0.32      0.42       164
           2       0.91      0.98      0.94      1565

    accuracy                           0.89      1788
   macro avg       0.56      0.44      0.47      1788
weighted avg       0.85      0.89      0.86      1788
```

It is the highest overall accuracy we got. However, there were other models that could perform better in each class and the highest weighted F1 is 0.5. However, the result from these models is not significant since our goal is to get precision and recall of all classes higher than 0.8.

For the CNN model, we also tested the different preprocessing methods and text data augmentation techniques.



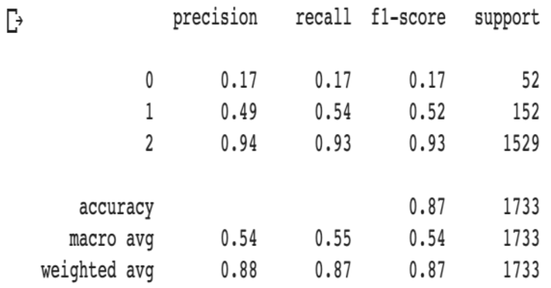
```
              precision    recall  f1-score   support

           0       0.19      0.15      0.17        52
           1       0.52      0.38      0.44       152
           2       0.92      0.95      0.93      1529

    accuracy                           0.88      1733
   macro avg       0.54      0.49      0.51      1733
weighted avg       0.86      0.88      0.87      1733
```

The CNN model with base case data performed better than these classification model but when we applied the augmentation technique, they somehow performed worse. But it still makes sense because we might not find the best CNN model at this time.

Bert is a pre-trained deep learning model by Google and it is quite powerful. The implementation of Bert is quite difficult because it was written on Tensorflow 1.x. The result of the Bert model shows its power. Without any text augmentation technique, the result of the model applying the second preprocessing method could get a macro F1 at 0.54, which is higher than some publications.



```
              precision    recall  f1-score   support

           0       0.17      0.17      0.17        52
           1       0.49      0.54      0.52       152
           2       0.94      0.93      0.93      1529

    accuracy                           0.87      1733
   macro avg       0.54      0.55      0.54      1733
weighted avg       0.88      0.87      0.87      1733
```

6 Discussion

Overall, we want to summarize some experiences and suggestions in the last section. First of all, we think the most challenging part is the data source, the citation data with sentiment labels from ACL Anthology might be the only free source we can get. The quality of the dataset made a huge influence on the project:

● The dataset is super imbalanced, even we know that when scientists publish scholars, they might be quite careful to state positive or negative views of others' achievement, so objective manner is a wise decision. But when we are training a machine learning model, the more information the model got, the better result it could present. If we want to enhance the citation sentiment analysis, we need a better dataset.

● The sentiment in the dataset is not quite clear. In order to check the sentences in the dataset, we applied the Vader Sentiment Analysis package, we have checked the positive and negative status on positive and negative classes, and we found only 27% of the sentences in the negative class have higher negative score and 66% sentences in positive class have a higher positive score. This might be quite similar in the first bullet point that authors are always quite careful when they are trying to express their manner, this made these sentences in the dataset labeled negative are not that negative.

● Amount of data. We only have 8.7k sentences and we believe if we want to perform the model on the Arxiv database, we need more data to make the model more accurate.

Besides the dataset, we want to talk about preprocessing methods, text augmentation methods, and models. There are many different preprocessing techniques we can find online, and we believe once we understand what the key features in a sentence is, then we can have a good result.

We spent lots of time on the text augmentation part is because we don't have a good dataset and if in the future, we still cannot find a better data source, then text augmentation would play a huge role in this project. The Markov Simulator is not performing as good as we think but it helped a lot, especially in these classification models. The oversampling method is good, but it also creates redundancy and the models are getting overfitting.

It is hard to tell which model is better because none of them helped us get the expected result. However, we believe Bert has the potential after we have either a better data source or a stronger text augmentation technique.

References

Haixia L (2017, April 1). Sentiment Analysis of Citations Using Word2vec. Retrieved from https://www.groundai.com/project/sentiment-analysis-of-citations-using-word2vec/1

Athar, A. (n.d.). Sentiment Analysis of Citations using Sentence Structure-Based Features. Retrieved from https://www.aclweb.org/anthology/P11-3015/