Sentiment Analysis - Classical Method and LSTM Machine Learning for Natural Language Processing 2021

LI Mo
ENSAE Paris
mo.li@ensae.fr

LI Zhonghao
ENSAE Paris
zhonghao_li@ensae.fr

Abstract

This report is based on our work on the sentiment analysis based on the tweets regarding the reviews of American airlines. Several methods of vectorization and classification were used and the performances were compared. Corresponding Python code to this project is available from personal page[1].

1 Problem Framing

Twitter has become a main platform where people express freely their opinion on their recent experiences. Some users on Twitter may also express their feelings after a flight, and in this project, we are interested in a sentiment analysis model to predict if the review given by the travelers after a flight is a positive, neutral or negative one.

We used the feelings expressed by travelers on Twitter in February 2015 from https://www.kaggle.com/crowdflower/twitter-airline-sentiment (which was also used in the lab session 3) as the sentiment analysis job about the problems of each major U.S. airline. Twitter data was scraped from February of 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as ”late flight” or ”rude service”).

2 Experiments Protocol

We first cleaned the data by removing those special characters and numbers, converting capital letters into lowercase ones, etc.

After cleaning the data, We first recalled the scheme adopted in the lab session, using a certain feature extraction method (Count Vectorizer) to convert phrases into vectors, whose length equals to the number of unique words appeared in the data-set. Then, a random forest classifier was used as a multi-class classifier to get the very first method.

[1] https://github.com/MoLI0914/sentiment-analysis

	precision	recall	f1-score	support
negative	0.77	0.97	0.86	1855
neutral	0.66	0.38	0.48	607
positive	0.86	0.44	0.58	466
accuracy			0.76	2928
macro avg	0.76	0.59	0.64	2928
weighted avg	0.76	0.76	0.73	2928

Figure 1: Score of the first method

We tried out the word-vectorization method, which used word2vector library to train the entire data. Then we get a model who was able to transform any word into a vector of 100 dimensions, with which we got the vectorized form of the average of all the words in every piece of tweet. This was used to represent the features of the tweets. Again, a random forest classifier was used to generate the prediction.

Moreover, we examined the performance of the LSTM model in this situation by using pytorch framework. We used also word2vector library in this case, and we finally got a 3D data as input of our LSTM model, while assuring that the length of each vectorized tweet shall be the same. During the optimisation process, Adam was used as optimiser.

3 Results

The first method, which was also implemented in the lab session, gave us a f1-score of .76 (see figure 1).

The second method gave us a similar f1-score .76 (see figure 2) as the previous one, which makes sense because they share the same type of classifier. However, a big advantage of this method is that it requires far less RAM, since the dimension

	precision	recall	f1-score	support
negative	0.76	0.97	0.85	1855
neutral	0.71	0.34	0.46	607
positive	0.79	0.47	0.59	466
accuracy			0.76	2928
macro avg	0.75	0.59	0.64	2928
weighted avg	0.76	0.76	0.73	2928

Figure 2: Score of the second method

	precision	recall	f1-score	support
0	0.86559	0.90683	0.88573	1889
1	0.63862	0.57586	0.60562	580
2	0.73005	0.67756	0.70282	459
accuracy			0.80533	2928
macro avg	0.74475	0.72008	0.73139	2928
weighted avg	0.79938	0.80533	0.80157	2928

Figure 3: Score of the third method

of the vectorized phrase (100) was far smaller than the previous one $(10, 520)$. This in turn gives us a strong advantage in storage.

The last method gave us the highest f1-score .81 (see figure 3), while using the same word vectorization method as the second one, but with an advanced predict model, LSTM, which is more complicated but also more powerful especially for the NLP problems.

4 Discussion/Conclusion

With this project being proceeded, it was realised that there are more and more newly-developed and powerful methods in the domain of NLP. Specifically, as the methods and the frameworks of Deep Learning being developed, the NLP models are growing more powerful, while we are having more choices for various problems. Choosing the appropriate method depends on not only the accuracy of those models, but also their difficulties and complexities.