DATA MINING AND TEXT MINING PROJECT2

A PREPRINT

Wen-Yuh Su
UIN: 671937912
wsu23@uic.edu

Lanxin Zhang
UIN: 679917816
lzhan43@uic.edu

September 7, 2019

ABSTRACT

This project proposed to use different models with several pre-processing methods for sentiment prediction. First, we conduct several machine learning methods such as Naive Bayes, Support Vector Machine with distinct pre-processing methods. In addition, we implement a deep learning model called LSTM; moreover, we add pre-trained word embedding when training. Experiments conducted on two datasets which are Obama and Romney attest the effectiveness of different models.

1 Introduction

Sentiment Analysis is an active research area in natural language processing which analyzes the sentiment or emotion from people's opinion or comments on a topic. The study of sentiment analysis has been spread from computer science to other fields such as social science, financial science and so on. For project2, we are asked to use the machine learning method to find the label given two datasets including Obama and Romney.

In recent years, there have been several document embedding techniques. For example, Naive Bayes [1], Support Vector Machine [2] which mentioned in class, all of these algorithms performs well on text classification. Among these methods, deep learning is regarded as the most state-of-art technique on sentiment analysis.

To advance the cutting-edge technique, in this project we propose to use LSTM [3] to compute the document embedding and do classification tasks as well. In addition, we combined the pre-trained word embedding from FastText [4] into our model. At last, our model contains a fully-connected layer and softmax layer for the prediction of document sentiment.

In our experiment, we show the result of the basic machine learning method under different pre-processing. Furthermore, by using deep learning methods and pre-trained word embedding to improve the performance.

2 Methodology

2.1 Pre-Processing

For the basic machine learning model, to reach high accuracy, it is vital to do the feature selection. First, we do some simple pre-processing such as removing the stopwords, doing the lowercase, and removing punctuation. Computing TF-IDF as the document representation. Experiments conducted on those data with simple pre-processing, TF-IDF feature selection, indicate that pre-processing is not sufficient since the word dictionary is still large.

Therefore, we tried to remove the HTML tags and the URLs which is not meaningful in sentiment analysis. This results in significantly improve the performance of the two datasets. Moreover, we attempt to deconstruct the text, for instance, changing "I'll" to "I will", or modify the misspelling. However, it only gives a little improvement in the classification task.

For deep learning, we run models with the preprocessing and pre-trained word embedding respectively. If we are going to use pre-trained word embedding, we would not use the standard preprocessing steps such as removing stopwords because that might lose the valuable information which is helpful for the neural network to do classification. The most

Table 1: Effect of preprocessing

	Obama		Romney	
Method	NB	SVM	NB	SVM
Removing punctuation, stopwords and doing lowercase	55.59%	55.02%	53.01%	53.15%
method 1 and removing HTML tags and URLs	**57.60%**	57.37%	53.74%	56.27%
method 1, 2 and deconstruct	57.33%	57.40%	**53.86%**	**56.45%**
method 1,2 and correct misspelling	57.54%	**57.47%**	53.74%	56.41%

important thing is to make our vocabulary as close to the pre-trained word embedding as possible. After removing the punctuation, we found that our vocabulary is covered 80% of the FastText word embedding.

Last, to have the same percentage of each class in the testing data, we split the two datasets into 10-fold with $StratifiedKFold$ function in sklearn. The below experiments are results of an average of 10-fold cross-validation.

2.2 Machine Learning Model

We compare our model on two baseline methods: 1) Naive Bayes, 2) Support Vector Machine. For the SVM, we set the parameter C to 4000 on both the Obama and the Romney dataset. According to experiment figure 1, through tuning, we find that if C is greater than 4000, the model would start overfitting. Nevertheless, if C is less than 4000, the model would be underfitting.

2.3 Deep Learning Model

Let L denote a sentence with n words. We represent L as a sequence which contains word embedding w_i:

$$L = (w_1, w_2, ...w_n) \tag{1}$$

The LSTM model is applied to encode these word embedding into hidden states $h_t \in \mathbb{R}^h$. In this project, we set length of the hidden state to 128. The last part contains a fully-connected layer and softmax function to generate the prediction result. The model becomes:

$$Pred = softmax(Relu(W_1 h_t)) \tag{2}$$

We tune the model with a learning rate of 0.01. Based on figure 2, we can observe that the model would start overfitting after two epoch. Thus, we set the model would early stop at second epoch. The resulting Matrix $Pred$ is finally passed to $argmax$ function for document sentiment analysis.

3 Evaluation

3.1 Effect of preprocessing

To evaluate the effect of preprocessing, we conduct four different of preprocessing method: 1) removing only stopwords, punctuation and doing lowercase, 2) method 1 and removing HTML tags and URLs, 3) method 1, 2 and deconstruct, and 4) method 1, 2 and correct misspelling. We run four different preprocessing methods on Naive Bayes and SVM respectively. Table 1 tabulates results of those experiment on Obama and Romney. As we can see that the most effective method is to remove the HTML tags and URLs. Besides, the deconstruct and correct misspelling still have some improvement to increase the accuracy of the classification task.

3.2 Deep learning result

For the deep learning model, first, we conduct the experiment with the same pre-processing setting with the machine learning method. Thus, the deep learning model would learn the word embedding during training. As we can see the effect of LSTM in Table 3, LSTM does improve in the Romney dataset; on the other hand, it is not able to do well on the Obama dataset.

Table 2: Pre-process for deep learning

Method	Coverage of vocabulary	Coverage of Text
None	38.14%	68.88%
removing punctuation	58.93%	82.48%
correct misspelling	58.92%	80.89%

Table 3: Effect of deep learning

	Obama				Romney			
	Accuracy		F1		Accuracy		F1	
Naive Bayes	57.54%	50.98%	57.60%	**62.81%**	53.74%	14.92%	13.31%	69.32%
SVM	57.40%	54.86%	56.26%	60.59%	56.45%	30.95%	35.81%	69.93%
LSTM	55.29%	52.32%	57.51%	55.90%	55.44%	28.96%	44.21%	69.18%
LSTM with pre-trained word embedding	**59.89%**	**55.76%**	**60.15%**	62.79%	**59.23%**	**37.60%**	**45.36%**	**72.16%**

Besides, we add pre-trained wiki word embedding to deep learning. Instead of using the same pre-processing method, we run some tests to find out how many words are covered in the pre-trained model. Table 2 indicates that the percentage of the words in the dataset which are covered by the pre-trained model. We can observe that it reaches high accuracy when removed the punctuation.

Based on the pre-processing mentioned above, we train the LSTM model with wiki new pre-trained word embedding. As a result, table 3 shows that with the pre-trained model, LSTM improves the accuracy significantly.

4 Conclusion

In this project, we implement several machine learning models with specific pre-processing methods. We show that a deep learning model with pre-trained word embedding is able to achieve a better prediction. We also demonstrate the importance of pre-processing based on different models.

References

[1] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.

[2] Marti A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, July 1998.

[3] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization, 2014.

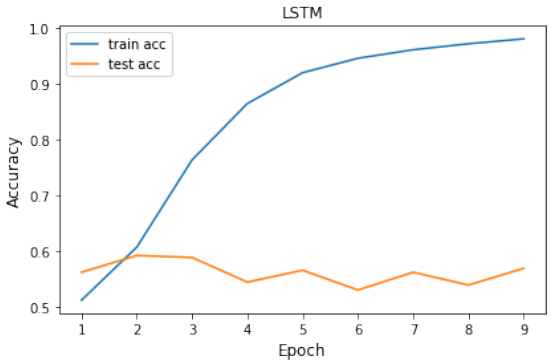

Figure 1: LSTM tuning

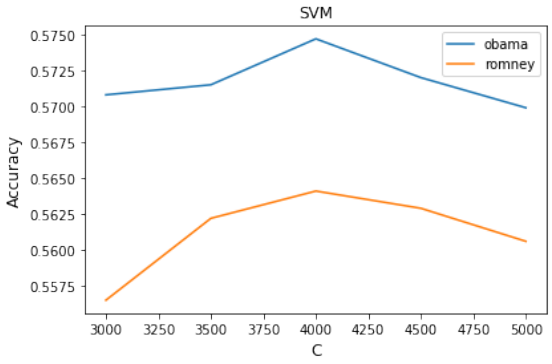

Figure 2: SVM tuning

[4] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.