Introduction to Machine Learning HW5

Wen-Yuh Su
UIN: 671937912

I. Purpose

I chose the option 3 which is a text classification task held by the Quora. The purpose of the competition is to use machine learning method to help Quora on handling the toxic or divisive content on its platform. Hence, the training data includes the problem users posted and the target whether the question is identified as insincere.

II. Method

First of all, I chose Bernoulli Nave Bayes as baseline method. For the nature language processing problem, I did search which model is fit for the NLP problem, and try some classification method on scikit-learn as well, but it turns out that SVM and Random Forest would spend too much time to complete the task with not good performance. Also, it is a binary classification task, so it is suitable for using Naive Bayes model. Therefore, I selected the Naive Bayes which is not only good for text classification task but also efficient. In addition, I read some papers which is related to text classification problems by using LSTM and with self-attention that can significantly improve the results of the traditional machine learning method.

In paper [1], the author proposed a method called self attentive method which is able to help model to focus on the crucial part in the sentences and reduce the influence from the common words. Therefore, I implemented it into code according to their formula.

$$A = softmax(W_{s2}tanh(W_{w1}(H^T))).$$

H is the hidden states from the bidirectional LSTM, and W_{s1} and W_{s2} are the fully connected layer in the neural network.

Finally, multiplying A back to the Hidden states, so we can have hidden states with attention(kinds of weight).

III. Experiment Setting

For the data setting, splitting the validation dataset from the training dataset by using scikit-learn split_train_test function with same random state, and data setup refers to the table I. In addition, for the model setting, according to the validation result, I setup the α equals to 1.0, and hidden states dim=128, learning rate=0.001, dropout rate=0.1, and 2 epoch in the LSTM and self-attention model based on the learning rate figure 1.

TABLE I. Data Setup

Dataset	Percentage	Shape
Training Data	80%	(1044897,3)
Validation Data	20%	(261224,3)
Testing Data	Default	(56370,3)

IV. Software

By using pandas, I can process csv file more efficient. To pre-process the text data, I used NLTK package to include the stop words dictionary. For the Bernoulli Nave Bayes model, I used scikit-learn package. Furthermore, I implemented the LSTM model and self-attention model by using Keras.

V. Results

The result shows that for the natural language processing problem, it is important to do data pre-processing so that the model can learn the most critical parts of sentences. Moreover, the deep learning model present usually present better than traditional machine learning model, but it has more hyperparameters resulted in difficulty to select the good ones. With self attentive LSTM model, it is supposed to be better than LSTM, but it might have some details that I did not notice so that I cannot surpass the LSTM model. More explanation is on my Github repository[1].

TABLE II. Experiment Results

Model	Setup	Validation Accuracy	Testing Accuracy
TF-IDF with Naive Bayes	no processing data	93.20%	52.2%
TF-IDF with Naive Bayes	remove stop words, punctuation	94.24%	54.0%
WordEmbedding with LSTM	remove stop words, punctuation	95.35%	62.1%
Word Embedding with LSTM and attentive structure	remove stop words, punctuation	95.45%	61.20%

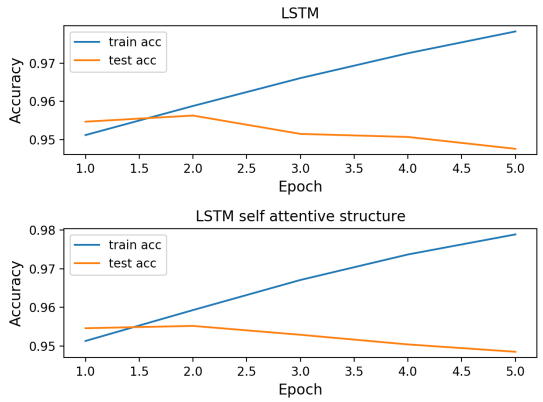

Fig. 1. LSTM Model Learning Rate Figure

[1] https://github.com/suwenyu/Intro_ML_HW5

REFERENCES

[1] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *CoRR*, vol. abs/1703.03130, 2017. [Online]. Available: http://arxiv.org/abs/1703.03130