CS410 Progress Report

Team Information
Team Name:
Team Commonwealth

Team Members:
1. Zuliang Weng / zwe
2. Zijing Chen / zijingc3
3. Liping Xie / lipingx2 (captain)

Project Topic
Competition - Text Classification

Project Progress

Tasks have been completed:

1. Made decisions based on the options in our proposal:
   - State-of-the-art neural network classifier: BERT
   - Deep learning frameworks: PyTorch
   - Programming Language: Python

2. Our testing result has passed the baseline:
   - precision =0.7333333333333333
   - recall = 0.7211111111111111
   - f1 = 0.727170868347339

3. Draft version code is ready, here is what we have done related to coding:
   We modified and fine-tuned BERT to train the text classifier. To be more specific, we tried some pre-trained BERT models. The reason why we chose a pre-trained BERT model is that the pre-trained BERT model weights already encode a lot of information about our language. As a result, it takes less time to train our fine-tuned mode.
   First of all, we installed the transformers package from Hugging Face which gave us a pytorch interface for working with BERT. Next, in order to apply the pre-trained BERT, we used the tokenizer provided by the model, and we tried “bert-base-uncased” and “bert-base-cased”. Since BERT has very specific formatting requirements, we loaded the data from the file and formatted it to match its requirements, such as added special tokens to the start and end of each sentence; truncated all sentences to the same length, etc. Then we used “BertForSequenceClassification” to train the model. This is the normal BERT model with an added single linear layer on top for classification that we used as a sentence classifier. Then we applied our model to generate predictions on the test set.

Tasks are pending:
1. Use Google Colab to train and fine-tuned our model. Training a large neural network in Google Colab can save some training time.
2. Try some other pre-trained model such as bert-large-uncased.
3. Fine-tuning the model, try different batch size, learning rate, and number of epochs.

4. Discuss how much context data will be used in training, since if we use all the context data, it will take a long time to train.

Challenges we are facing:

1. It takes too long to train the model locally since the context is very long.
2. The testing result is still not good enough, we need another way to improve the result.