CS 410 Progress Report

Michael Bencsik
bencsik2@illinois.edu

Team Members

Number	Name	ID
1	Michael Bencsik	bencsik2

Project Summary

The goal is to perform stock market sentiment analysis by classifying text from news headlines to stock market prices to determine if there is a strong relationship between the direction of stock price changes and human emotion and if this outcome is predictable.

Tasks Completed

I have been working with one dataset initially to implement examples of text classification from TensorFlow and PyTorch [1]. I was able to clean up a lot of the text data, but more cleaning could be done. I am unsure as to how far to take this before the text begins to lose its meaning.

For TensorFlow, the easiest way to load the data from a CSV file was by using the pandas python module. The loaded data takes the form of a panda's dataframe type, which can be formatted, sliced into training and test sets, and reshaped. Then, a TensorFlow dataset was created from each dataframe. The data was then tokenized using TensorFlow Text's Unicode Script Tokenizer. The next step was to split the dataset into training and validation sets. I ran into issues here, due to my data not being formatted exactly as the example. The example was mapping a function across all elements while splitting the data. My data will need to be reformatted or the mapping function will need to be re-written for my purposes.

For PyTorch, I was able to create a subclass of the abstract class Dataset. This allows the data to be formatted, split, and indexed in any way due to overloading the operator methods. The class also used pandas to read in the data from the file as a dataframe, which is an attribute of the class. An instance of the training and test sets were created. The text data was tokenized using TorchText's tokenizer, then a vocab was created. The vocab vectorizes the words, making a bag of words model. The data was batched and ran through an Embedded Bag function, which seems to just find the mean

of the vectorized words. After the model was trained, the accuracy was approximately 50% for most runs. This gives me an equivalent to a coin toss for a correlation.

Tasks Pending

Pre-Processing Data

I still need to format my datasets more consistently and in a manner that works well with PyTorch and TensorFlow frameworks. I need to clean my data better prior to tokenization. Some of the tokenizers will separate punctuation, which means I am getting erroneous results, due to an incorrect vocabulary. Stop words need to be removed from the vocabulary, so that my models are not training based on high frequency irrelevant words. If time permits, I would like to incorporate N-grams or TF-IDF vectorization instead of a uni-gram vectorization.

Model Training

I would like to run multiple models from PyTorch and TensorFlow as well as multiple datasets for each model. Ideally, I would like to compare the accuracy of different models such as neural networks, SVM, logistic regression, Naive Bayes, and decision trees.

Documentation and Presentation

Once I have a better understanding of pre-processing the datasets and have the ability to compare multiple models, I will need to document the functionality of the code and how to use it. Lastly, a presentation needs to be created to demonstrate the installation and use of the code.

Challenges

The main challenges currently being faced are the pre-processing and formatting of the datasets. This is the first time I am using these frameworks from scratch, which takes familiarization with the frameworks and documentation. I am also running into issues working with such large datasets. It is hard to determine which issues are causing the biggest impact to the results when I am unable to manually scan the entire dataset.

References

[1] Sun, J. (2016, August). Daily News for Stock Market Prediction, Version 1. Retrieved [10/23/22] from https://www.kaggle.com/aaron7sun/stocknews.