**CS274P Project Report -**

**Twitter sentiment analysis during the pandemic**

Yifei Li, Hanyan Wang, Yukun Shi

**Project Overview**

There are more than 100M COVID-19 cases around the world until March 2021, and many people are suffering from it. Though several vaccines have come out, there is still a long way to go before humans conquer it.

Twitter is one of the most popular social networking sites with millions of users and posts every day on a variety of topics. The tweets can be a powerful indicator of the trend and concerns from the public. Performing sentiment analysis on the recent posts, we can find out the user’s idea and attitude during the pandemic period.

Sentiment analysis is a popular research topic in natural language processing, many models have been developed to carry out such tasks, such as support vector regression(SVR), random forest, and decision tree. In 2019, Saad and Yang [1] extracted features from tweets and performed ordinal regression on them to achieve high accuracy. Kumar [2] proposed a hybrid deep learning model which handled real-time data for predicting sentiment in 2020.

In this report, we aim to apply what we learned in class (inner and outer approach) and what we learned out of class (Bert, FastAi) to explore sentiment analysis technique. In particular, we implemented three basic models and two advanced models to analyze sentiment in tweets during the pandemic.

**PipeLine**

The goal is to use three basic models and two advanced models to analyze sentiment in tweets during the pandemic. The task involved are the following:

1. Download and preprocess the Corona Virus Tagged Data
2. Train a feedforward neural network to set up the baseline model
3. Implement two models that are introduced in CS274P and two advanced models that learned from our independent study
4. Graph the confusion matrix and compare the results between models

From the results, our two advanced models have the best performance on analyzing the sentiment of each twitter text.

**Data**

Our data comes from a Kaggle competition:

<https://www.kaggle.com/datatattle/covid-19-nlp-text-classification>.

The dataset collects over 40,000 tweets during March and April last year, not only from the United States but all over the world. There are 4 columns, namely Location, Tweet At, Original Tweet and Label. It divides all the tweets into 5 categories: extremely positive, positive, neutral, negative, extremely negative. Their distributions are as follows:

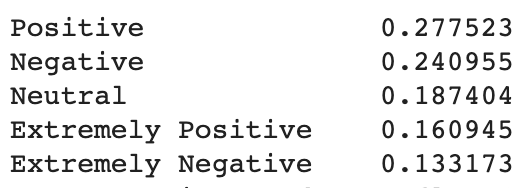
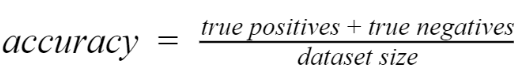


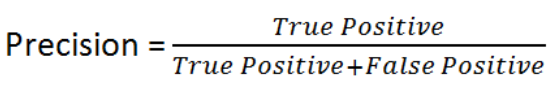
Figure 1: distribution of sentiment labels in the dataset

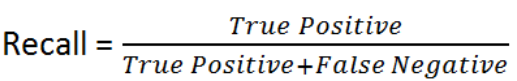
We extract the Original Tweet and Label columns from the data, the former is used as input while the latter is output. In order to preprocess the data, we remove useless symbols in the original tweet such as html tags, urls, mentions and digits. Then we tokenize the data by building a dictionary of it. Each distinct word is assigned an index, more frequent word has a lower index. After we convert each sentence into a list of indexes, we pad them to the same length to feed into our models.

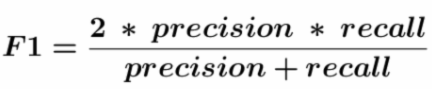
**Metrics**

We use various metrics to measure our models’ performance - accuracy, precision, recall, and F1 score, each is explaining below.

****

****

****

****

**Methods: Architecture Design and Training**

We tried 5 models, 3 basic models: feedforward neural network, bidirectional LSTM, convoluted neural network and 2 advanced models: Bert and FastAI.

Model 1: Feedforward neural networks contain an embedding layer with 574 K parameters and 4 dense layers with size 128, 64, 32, 16 and an output layer.

Total parameters: 696 K.Training Time:

Model 2: Bidirectional LSTM model contains the same embedding layer and a bidirectional LSTM layer with 559k parameters, a max pooling layer, a dense layer and an output layer.

Total parameters: 1.16 M

Model 3: Convolutional neural network contains three convoluted layers with the same kernel size (kernel size = 4), a dense layer and an output layer.

Total parameters: 580 K.

Model 4: Bert [3]. A pre-trained Bert layer with parameter 110M, a dense layer and an output layer. Pre-trained model link: <https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/2>.

Total parameters: 110M.

Model 5: FastAI. Freeze the layers of a pre-trained AWD-LSTM and only trained on the last layer to capture in-depth information of the dataset, and unfreeze all other layers for the second training.

Total parameters: 24M.

**Results**

Table 1 has the detailed result of our five models. As you can see, the feedforward neural network, which is our baseline model, has an accuracy of 65%. All other four models have more or less improvement over the baseline models, with Bert and FastAI stand out and have accuracy around 84% and 83%.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Precision Recall** | **F1-Score** | **Accuracy in %** |
| Feedforward neural network | 0.68 0.65 | 0.65 | 65 |
| Bidirectional LSTM | 0.76 0.75 | 0.76 | 75 |
| Convolution layers + Dense layers | 0.68 0.67 | 0.68 | 67 |
| Bert | 0.85 0.84 | 0.84 | 84 |
| FastAI | 0.84 0.83 | 0.83 | 83 |

Table 1: Test result of our 5 models

Figure 2 shows a visualization of accuracy of our five models.

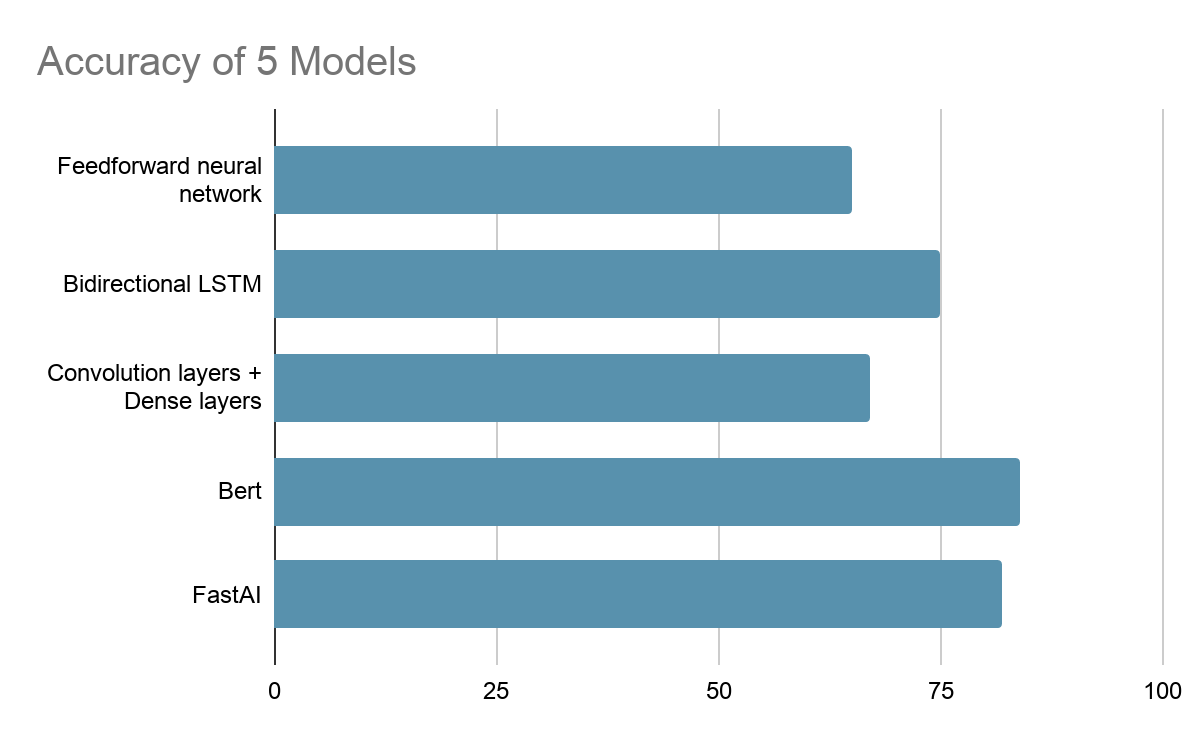
****

Figure 2: Comparison of testing accuracy between 5 models

**Conclusion and Future Work**

Sentiment analysis is a hot topic in the field of text classification. In this report, we collected tweet data from Kaggle and applied 5 models to analyze sentiment. In this project, we explored sentiment analysis techniques from five different approaches with five distinct models. With the inner and outer approach learned in class, we are able to produce desirable results. With the two advanced models, Bert and FastAI, **we succeed to produce the State-of-Art result** among the Kaggle submitted projects (The average accuracy on Kaggle is around 75% and some top competitors have accuracy around 85%, compared to our accuracy around 84%).

Moreover, there are three main fields in the domain of Natural Language Processing - sentiment analysis, semantic analysis, and syntactic analysis. Currently, our best sentiment analysis model has an accuracy of 84%. We can further improve it by applying the other two techniques, including but not limited to parsing, relationship extraction, and keyword extraction to do more **in-depth** analysis. In addition, we can also use our trained model in this project to perform **in-breadth** tasks, for example, to generate information from other covid-related texts using the transfer learning technique (FastAI) we learned in this project.

**Reference**

[1] S. E. Saad and J. Yang, "Twitter Sentiment Analysis Based on Ordinal Regression," IEEE Access, vol. 7, pp. 163677-163685, 2019.

[2] AkshiKumar, KathiravanSrinivasan, ChengWen-Huang, and Albert Y.Zomaya, "Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data", Information Processing & Management, vol. 57, no. 1, January 2020.

[3] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

**Contribution**

Hanyan Wang: preprocessed data+trained the baseline model

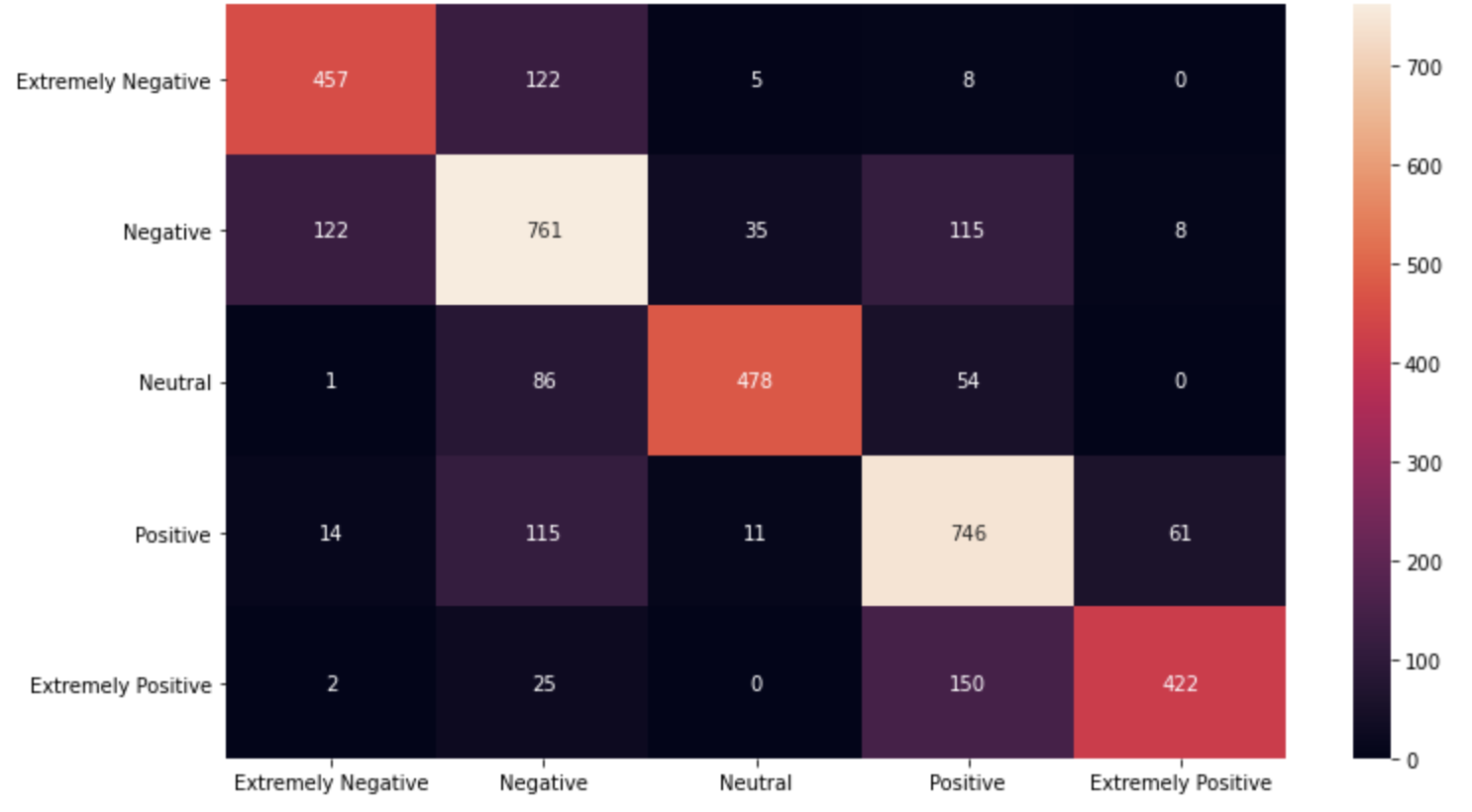
Yukun Shi: designed and trained model 1-4 (feedforward NN, convolutional NN, Bert)

Yifei Li: designed the FastAI implementation to apply transfer learning on AWD\_LSTM model. Evaluated and compared each model’s testing result.

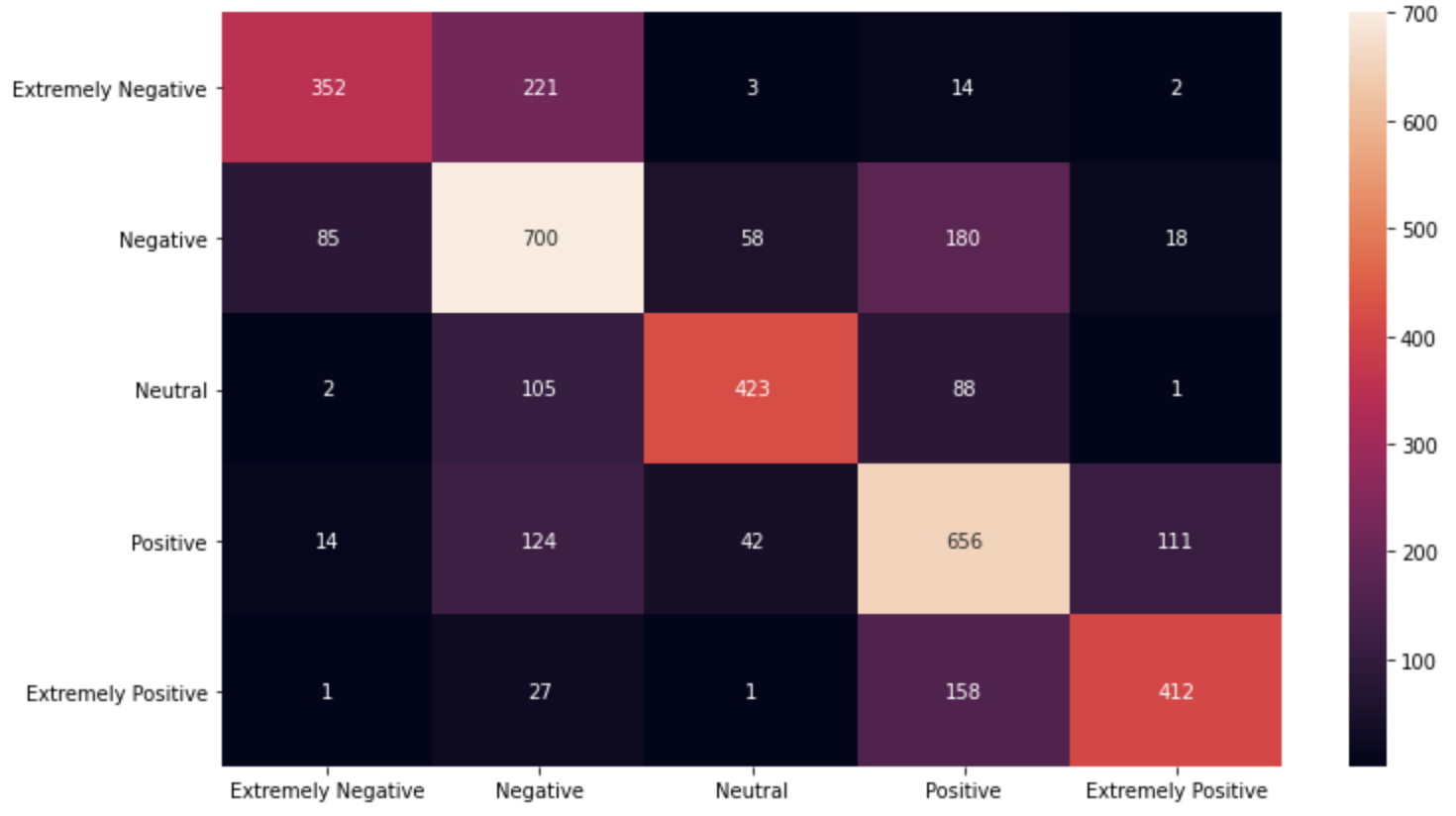
**Appendix**



Feedforward neural network



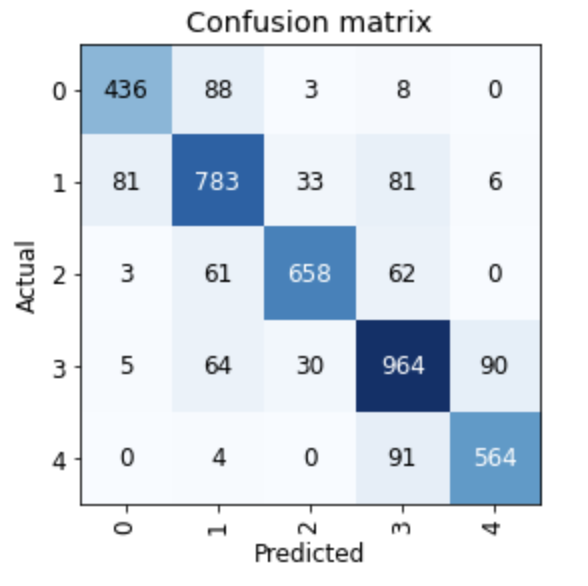
LSTM



Convolution layers + Dense layers



Bert



FastAI