CS230

Twitter Sentiment Analysis for Predicting Stock Price Movements

Kevin Hu, Daniella Grimberg, Eziz Durdyev *
Department of Computer Science
Stanford University

1 Introduction

1.1 Problem and Motivation

One of the central focuses of financial engineering is predicting stock prices, however, this task has proven to be challenging due to the various sources of information that could impact the prices and the complexity of the task. In this paper, we test the hypothesis that stock prices are correlated to and can be predicted using public sentiment expressed on social media using Deep Learning tecniques. Specifically, we investigate twitter as a source. The influence of social media forums in stock markets and potential insight they offer was demonstrated through recent events such as the short squeeze on GameStop's stock in January 2021 due to a Reddit thread and the massive decline in Bitcoin prices in June 2021 following a tweet by Elon Musk suggesting a 'breakup' with the cryptocurrency. In this project we propose leveraging twitter data using Deep Learning tecniques focusing in the field of NLP to predict stock prices. Using Twitter API we extract daily tweets pertaining to a company, and use a pre-trained BERT model to generate embeddings and classify tweets as positive/negative, we then combine daily averages of sentiment with metadata for the tweets as input to a classifier model that predicts whether stock prices will increase or decrease for a certain company.

1.2 Literature Review

There is extensive literature using twitter data to predict stock prices. Silbersdorf et all [1] proposes using sentiment analysis on tweets combined with data on frequency of tweets as input to an LSTM for stock price prediction. With promising results, this work suggests publicly available twitter data can be very useful for stock prediction. Bansal et all. [2] proposes using a BERT model for the sentiment analysis and combined with stock price movements data. There are many other works attemting to predict stock prices using historical stock market data(without the use of social media data), for example Lin et. all [3], combining SVM with Arima models for predicting stock prices. Overall, most proposed solutions to predicting stock prices use stock data and sometimes combine them with information obtained from social media. We propose a different approach using only Twitter data as features, testing the hypothesis that public sentiment expressed on twitter can be used to predict stock prices.

1.3 Approach

To tackle the problem of stock price prediction using sentiment analysis, we built a pipeline with two main components. The first component is a sentiment classification model that is used to represent the

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

public's opinions and attitudes that are displayed through tweets. The second component is a binary classification model that utilizes the sentiment feature representation, along with other metadata of the twitter posts, to predict the up or down movement of a given stock.

2 Sentiment Analysis

In the first stage of the pipeline, the goal is extract public sentiment from social media posts. For a given stock at a given date, we will scrape Twitter prior to the time period to find a batch of one hundred tweets that are relevant to the company. Here, we are making the critical assumption that these tweets is a good representation of the public sentiment (and thus valuation) towards the company. We choose the batch size to be 100 due to the data constraints from the Twitter API. Ideally, it would be better to work with a larger batch size, which gives us a more accurate representation of Twitter sentiments, as well as reduce the impact of extracting meaningless or empty tweets. However, to ensure that we can scrape enough data for a sizable collection of stocks to train the model, we decided that 100 is reasonable choice for the batch size. The batch of tweets will be inputted into a BERT model that will extract their associated sentiment values (positive or negative), and compute an average measure of these values.

2.1 Dataset for Training Sentiment Classification

The dataset used to train the sentiment classification model is the sentiment140 dataset, which contains 1.6 Million tweets with tagged sentiment as positive or negative. We can download it and use it freely. From this dataset, we have the feature of tweet text and the label of sentiment as Positive/Negative. We will use this feature and label for the BERT model.

2.2 Sentiment Classification Model

The high level structure of our model is shown in Figure 1. The input is a batch of m tweets (preprocessed with a BERT tokenizer, which is not shown in the diagram). For each tweet, we run a pretrained BERT from Hugging Face to compute binary sentiment value (1 for positive, 0 for negative). We will be testing and tuning the sentiment model based on the results of this binary classifier. Finally, the sentiment values from the batch is averaged into an overall sentiment output for the stock. For the project, we are just taking a simple mean over the sentiment outputs; we also considered taking some form of weighted average by incorporating the meta data of the tweets. For instance, it seems to make sense to assign more weight to tweets from verified users with many followers and retweets (such as an Elon Musk tweet) than tweets that might be less influential (such as from your average user). However, for the purpose of our experiments, this weighted sum idea poses another problem. As a reference, Elon Musk has over 60 million followers on Twitter, while the average account only has about 700. This means that if we weigh the features according to the number of followers (either by taking ratio or using softmax), then this means that often, an entire batch is overshadowed by a single tweet.

This overall sentiment value can be used as a baseline predictor to the direction of the stock movement. Assuming perfect correlation between the sentiment and the price movement, we can predict the stock going up if sentiment value is larger than a threshold (e.g. 0.5) and the stock going down otherwise.

2.3 Model Results

Table 1: Fine-tuned BERT Sequence Classification Model with 100,000 samples

Epoch	Train Loss	Val. Loss	Val. Accuracy	Learning Rate	Batch Size
1	0.4	0.36	0.8462	2e-5	32
2	0.28	0.36	0.8507	2e-5	32
3	0.19	0.42	0.8492	2e-5	32

Initially, we did a quick fine-tuning with 8,000 randomly sampled samples from the Sentiment140 dataset which accurately predicted 80% of a validation set. Then, we fine-tuned the BERT sequence classification model with 100,000 random samples from Sentiment140 by running 3 epochs with

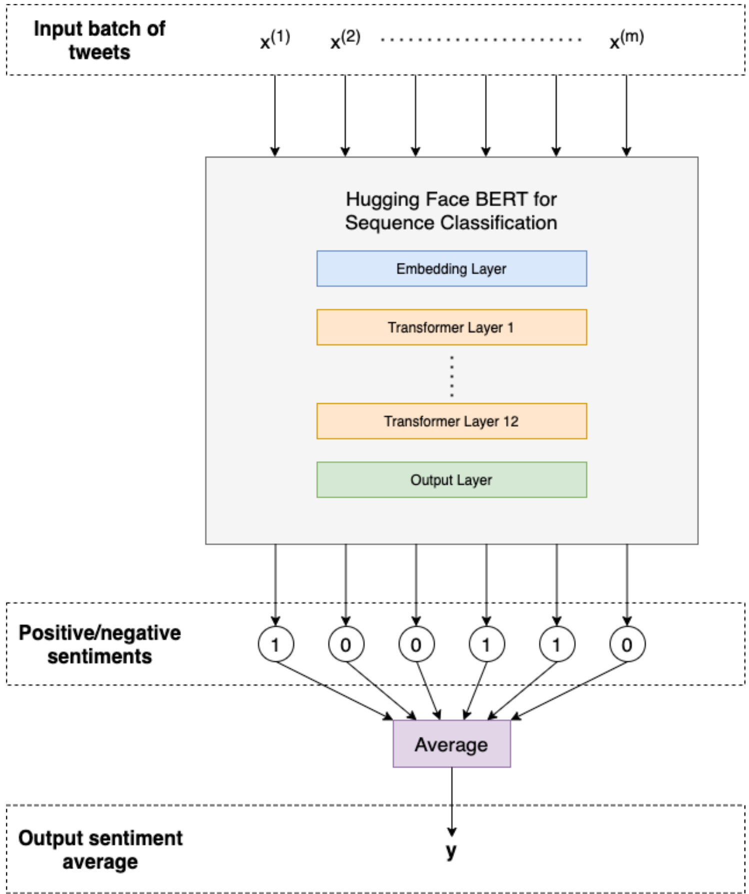

Figure 1: Sentiment classification using Hugging Face's pretrained BERT

batch size 32 and learning rate 2e-5. Results in Table 1 show that after the 2nd epoch the model starts to overfit to the training set as validation loss goes up in the 3rd epoch, so we will fine-tune the BERT sequence classification model with more samples from the Sentiment140. However, even with 100,000 samples from the Sentiment140, results are very promising. At epoch 2, the model accurately predicted 85% of a validation set.

3 Stock Movement Prediction

In the second stage of the pipeline, the objective is to build a predictor of stock market movement based on the public sentiments that is extracted from the previous step. Instead of using a single scalar value (the sentiment model output) to make the prediction, it makes more sense to apply the concept of transfer learning and use a feature representation of the public sentiment by taking the activation of the second last layer of the transformer. Starting with this encoding of the sentiments, we combine it with metadata of the tweets (including number of followers, retweets, whether the account is verified, etc.) to form the input vectors. The input is passed through a fully-connected neural network, which outputs a prediction of whether the stock price will go up or down.

3.1 Dataset for Stock Market Prediction

The sources we used are the publicly available Twitter Developer API for the input sentiment data, and Yahoo Finance for the output label (daily price movement for various S&P 500 stocks). From the Twitter API we obtain the raw text of the tweets as well as metadata such as number of likes, retweets, number of followers of user, verification of user, to be used as feature. The text from the tweet will be pre-processed (see next section) and converted into a sentiment embedding using the previous part of the pipeline. From the Yahoo Finance API we extract the labels (of whether the stock price for the relevant companies increased or decreased in a certain day). For each stock symbol on a given trading day, we aggregate 100 related tweets prior to that date according to Twitter's cashtag functionality and join the data with the labels from Yahoo Finance.

3.2 Pre-processing the Dataset for Stock Market Prediction

In this section, we describe the process of tokenizing twitter posts and extracting the embeddings for the stock price prediction model.

3.2.1 Tokenizing Tweet Text

We tokenized the raw text from each tweet by calling Hugging Face transformer API's BertTokenizer. BertTokenizer has a unique way of tokenizing text sequences. First, it breaks each word into tokens. For example, the word "rumination" is broken down into two tokens: "rum" and "##ination". Then, the tokenizer appends special tokens [CLS] to the beginning and [SEP] to the end of each text sequence. Finally, the tokens are replaced by their corresponding ids in the BERT Model vocabulary. At this point, the tweet texts are ready to be fed into the BERT Model to extract their feature representation. Figure 2 illustrates the BERT tokenizer workflow.

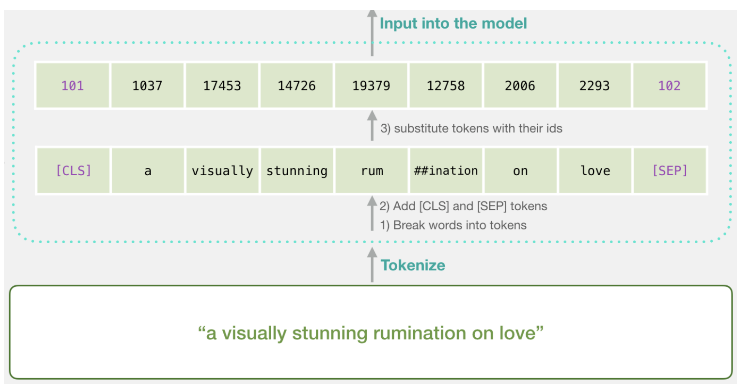

Figure 2: BERT Tokenizer. Image source: https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

3.2.2 Extracting Tweet Text Embedding

After obtaining the ids for each tweet, we utilized Hugging Face transformer API's BertModel as the following:

```
with torch.no\_grad():
    last_hidden_states = model(ids)
```

where last_hidden_states is 3 dimensional tensor. Its first axis is the batch size, the second axis is the length of the text sequence, and the last axis is the hidden unit output for each token of the text sequence. We extract the hidden units for the first token, [CLS], of each tweet text because in the BERT Model, hidden units of the first token of each text represent the whole text sequence.

After extracting a BERT vector representation for each tweet text, we aggregated the Twitter data for a stock on a given day. During the aggregation, we took means of the tweet data features. Finally, we concatenated the sentiment features of the tweets with numerical features such as is_verified, num_followers, and num_retweets to create a pytorch tensor. The size of the concatenated tensor is [871, 771], where 871 is the size of the dataset and 771 is the size of the input features. 771 is the concatenation of 768, 1, 1, and 1.

3.3 Limitations and Challenges

3.3.1 Dataset Construction and Preprocessing

Building the dataset was a major challenge for the project, as the Twitter API has daily usage limits and only allows us to search for tweets in the past 7 days. This becomes even more challenging as we can only use data for the 5 trading days that are relevant to the stock market. Another problem was finding enough stocks that had a substantial number of related tweets to build a robust dataset. There are thousands of financial instruments out there, and only the most common or "hot" stocks (such as Apple and Tesla) garner enough attention on Twitter. Moreover, the twitter data needs extensive cleaning. As the search relies on string matching, we may obtain irrelevant tweets (for example, many tweets simply listing the name of various stocks without further information). In the end, although we were able to obtain over 50 thousand tweets, many were useless and when compiled together (tweets per company per day) there were not enough "good" data points which could have caused the model to not learn as well.

3.3.2 Model Inputs

Besides the lack of abundant quality data, it could be the case that our input does not contain enough information to make accurate inferences on stock market movements. One of the goals of this project is to investigate whether Twitter sentiment features have a strong correlation with the daily price movements. Empirically, it seems like the answer is no. Even if we inserted additional hidden layers in the binary classification neural network, the model stops learning after one epoch, which suggests that the issue is not with the expressiveness of the network, but rather with the data or the optimization problem itself. As an extension to the project, we think there are two ways to improve the prediction model.

The first idea is to collect more features to increase the predictive power. For sentiment features, it may be better to scrape financial news articles and blog posts instead of tweets. And in terms of other features, it might be helpful to collect a variety of financial data (such as historical prices and related signals) and economic indicators (such as index movements, interest rates, etc.). The second idea is to re-frame the prediction problem. Currently, we are trying to predict the daily movement of stock prices (which is highly volatile) using public sentiments (which generally changes at a much larger time scale). Within a short time frame, the sentiments on Twitter may not change much, but the stock price can go either way. Perhaps a better prediction problem is to look at the up and down price movements over a longer period of time (e.g. over weeks, months, or even years) and use sentiment features in these time frames. This would require much more data and computation, but could lead to much better results.

4 Conclusion

For this project, we explored two aspects of sentiment analysis. The first is fine-tuning a sentiment classification model based on a pretained BERT. We are able to achieve good results by training on the Sentiment140 dataset. The second is building a prediction model for stock price movements. Due to the inherent difficulty of the problem as well as the challenges discussed in the previous sections, we have been unsuccessful in our attempts to solve this task, and recognize that there are many paths for future improvement.

References

[1] Marah-Lisanne Thormann, Jan Farchmin, Christoph Weisser, Rene-Marcel Kruse, Benjamin Safken, Alexander Silbersdorff. Stock Price Predictions with LSTM Neural Networks and Twitter Sentiment. https://pdfs.semanticscholar.org/9f91/57bbc

[2] Priyank Sonkiya, Vikas Bajpai, and Anukriti Bansal. 2021. Stock price prediction using BERT and GAN. In Proceedings of ACM Conference (Conference'17). ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/ nnnnnnn.nnnnnnn

[3] . P.-F. Pai, and C.-S. Lin, A hybrid ARIMA and support vector machines model in stock price forecasting, Omega, vol. 33, no. 6, pp. 497–505, 2005.