Effect of President's tweets on the S&P 500 index

Sid Assawaworrarit*
Department of Electrical Engineering
Stanford University
sca@stanford.edu

Abstract

Social media platforms like Twitter enables national leaders to communicate directly with the public. The information released potentially drives the movements in the stock market in real time. Here we build machine learning models that predict the likely movement of the S&P 500 index-tracker SPY price in response to a tweet released by President Trump based on word encoding with LSTM and sentence encoding with neural network. We train the models on President Trump tweet record from 2017 to November 2019 and find that the trained models offer no significant difference in the prediction error compared to a baseline model that always predicts no change. The lack of accuracy improvement from the machine learning models likely stems from the weakness or non-existence of any direct relationship between tweet text and SPY movement.

1 Introduction

The recent rise in popularity of social media has afforded national leaders instant and direct communication channel to the world's audience. With a few keystrokes on a smart phone, the president can send tweets on a myriad of subjects whose contents can have far-reaching consequences to companies, markets, and trades. Since taking office in 2017, President Trump has been a prolific twitter user, sending out on average more than ten tweets per day. Given the administration's nationalistic and at times interventionist stance on trade, it is reasonable to expect that what the president tweets influences the movements of the stock market index which itself serves as a barometer of the country's economic sentiment. Machine learning (ML) tools, particularly in the area of natural language processing (NLP), can be trained to predict how the stock market indices such as the S&P 500 index will move in response to a given twitter message. In this project, we seek to find out if such a model can be constructed and, if so, how accurate are its predictions. The implication could be profound if accurate predictions are achieved as questions of stock market manipulation would be inevitably raised. The input to our full model consists of a Trump tweet text message and a history of minute-by-minute SPY (which tracks the S&P 500 index) price history of the past hour before the tweet is sent. The full model consists of two separate LSTM networks for text and SPY history data. The outputs of these networks are then combined with a fully-connected neural network to output a predicted SPY change in percentage point 15 minutes after the tweet is sent relative to its value before the tweet is sent. Comparison with models with only tweet text input and one with only price history input is also given.

*Github: https://github.com/physicszid/cs-230-2019-trump-tweet-spy-project

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

2 Related work

While there are various studies on the relationship between Twitter sentiment analysis and the stock market performance [1, 2, 3], the amount of work regarding President Trump tweets is rather limited, and findings so far either tend to be rather inconclusive or found minuscule association of the president's tweets on the stock market index movement. Vox [4] found correlation between Trump tweets using certain words such as 'China' or 'billion' to volatility in the Treasury bond market. Barron's [5] found slight negative movements of 12 and 5 basis points of the S&P 500 index during the course of one trading day for tweets about tariffs and the Federal Reserve, respectively. (One basis point (bp) equals one-hundredth of one per cent.) In a similar study [6], the authors constructed a classification model based on LSTM which outputs predicted label $y \in -1, 0, +1$ of the likely movement of the S&P 500 index where -1 is a decrease, 0 is no change, and +1 is an increase. Although doing so may make the prediction task simple, it leaves out the possibility of being able to quantitatively predict how much the index will move in response of a tweet, a quantity which can be a matter of significant interest. In this work, we choose to construct a regression model whose output is the predicted percentage change of SPY so effect of Trump tweet on the stock market could be quantified.

In addition, recent advances in natural language processing (NLP) which enables machine understanding of unstructured data such as written texts, and reccurent neural network (RNN) which enables effective machine learning of sequence data, are applied in this work. Specifically, pre-trained word vectors from Global vectors for word representation (GloVe) [7] are used to transform tweet words into encoded numeric vectors. And, long short-term memory (LSTM) [8, 9], a type of RNN architecture that uses hidden cells to learn sequence data, is used extensively in this work.

3 Dataset and Features

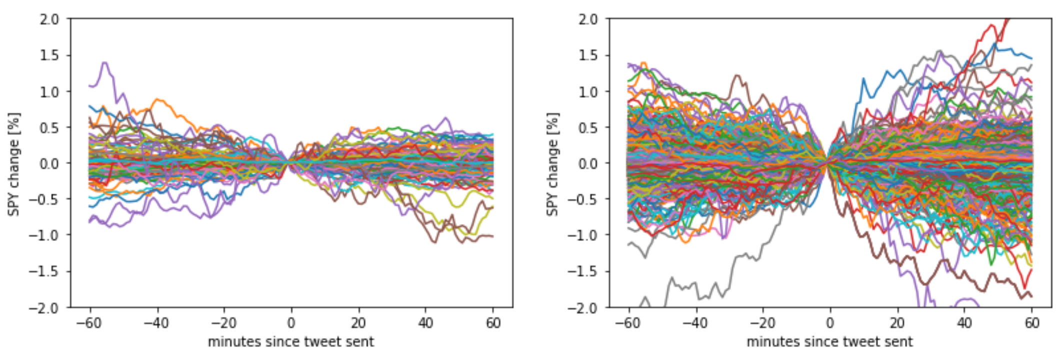

Figure 1: Plots of SPY percentage movements around the time President Trump sends out a tweet. Left panel: 100 random tweets; right panel: all tweets.

Two datasets are identified and collected: President Trump's tweet record and a record of stock market index. The record of President Trump's tweets can be downloaded from www.trumptwitterarchive.com [10] which contains the two main data that are relevant to this work: (i) the time stamp and (ii) the tweet's text, among other fields such as favorite and retweet counts. We decide not to use the favorite and retweet counts because they occur after the tweet is sent and therefore would not allow the model to work in real-time. We select only Trump own tweets and not his retweets. The record for stock market indices, at the detailed level needed for this work, is hard to find. Public, freely-accessible databases such as Yahoo and Google keep track of only the daily open and close prices which are not frequent enough to finely correlate the market movement with the publication of President's tweets. Luckily, Wharton Research Data Services (WRDS) [11] offers limited free access to its stock market database for students. The data for SPDR S&P 500 ETF Trust (ticker: SPY) which tracks the S&P500 index is chosen as it represents the state of the overall stock market. The data from WRDS contains all the transaction data (price and volume) being traded typically at microsecond frequency.

Time (ET)	Tweet (x)	y (%)
2019-08-13 09:38:44	Thank you Steve! https://t.co/4qwvkUTg5t	+1.065
2019-08-01 13:26:11	We look forward to continuing our positive dialogue with China on a comprehensive Trade Deal, and feel that the future between our two countries will be a very bright one!	-1.272
2019-08-01 13:26:10	buy agricultural product from the U.S. in large quantities, but did not do so. Additionally, my friend President Xi said that he would stop the sale of Fentanyl to the United States – this never happened, and many Americans continue to die! Trade talks are continuing, and during the talks the U.S. will start, on September 1st, putting a small additional Tariff of 10% on the remaining 300 Billion Dollars of goods and products coming from China into our Country. This does not include the 250 Billion Dollars already Tariffed at 25%	-1.267
2018-12-24 09:31:50	Virtually every Democrat we are dealing with today strongly supported a Border Wall or Fence. It was only when I made it an important part of my campaign, because people and drugs were pouring into our Country unchecked, that they turned against it. Desperately needed!	-0.7112
2018-11-27 14:05:39	Very disappointed with General Motors and their CEO, Mary Barra, for closing plants in Ohio, Michigan and Maryland. Nothing being closed in Mexico & China. The U.S. saved General Motors, and this is the THANKS we get! We are now looking at cutting all @GM subsidies, including for electric cars. General Motors made a big China bet years ago when they built plants there (and in Mexico) - don't think that bet is going to pay off. I am here to protect America's Workers!	-0.0281
2019-08-05 12:00:14	China is intent on continuing to receive the hundreds of Billions of Dollars they have been taking from the U.S. with unfair trade practices and currency manipulation. So one-sided, it should have been stopped many years ago!	-0.0218

Table 1: A sample look at the training dataset. Data points with large changes are specifically included.

The following preprocessing steps are applied. URLs, hashtags, mentions and emojis are removed from tweet text with tweet-preprocessor python package. Then, the text is converted into lower case and split into a list of individual words. Pre-trained GloVe encoding (glove.twitter.27B.50d) specifically trained on tweet text data is applied to convert each word into a 50-dimension encoding vector. The SPY raw data contains data points that are too frequent and noisy. To prevent this from affecting the model, we clean it up by averaging over a fixed time interval (ΔT). From trying a few choices ΔT and find that ΔT of 1 minute offers a good compromise between averaging out the noisy spikes while maintaining up-to-date SPY movements. Next we correlate the price data with the tweet time stamps. For each tweet sent during open market hours, we crop out the resampled SPY price data one hour prior and after the tweet time. The price data is then converted to percentage change with respect to the value before the tweet is sent. One input data point then consists of the clean tweet text embedding vector and the minute-to-minute price history in percentage point for 60 minutes before the tweet is sent (but not after). The expected output is the percentage change at 15 minutes after the tweet is sent. The lag time of 15 minutes is chosen for it is long enough to not correlate with short-term trend but short enough to filter for the direct effect of Trump tweet. Figure 1 shows relative SPY movements around the time each of Trump tweet is released. We find that the SPY trajectories are close to those- of random walk. This is to be expected as stock markets, and financial markets in general, are notorious for being difficult to predict. It is difficult to discern any effect Trump tweet from this visualization. In Table 1, we take a look at some of the outliers in the dataset where large (1%) change occurs. We can see that for the tweets that lead to negative movements there are mentioning of 'China' and reference to trade war. However, other tweets that also mention 'China' and have reference to trade war do not lead to any out-of ordinary changes in SPY.

The number of the total data points collected is $m = 1645$ which is the number of Trump tweets from 2017 to November 2019 that are released during open market hours. Of these, We randomly split the data into 1145 training data, 200 validation data, and 300 test data.

4 Methods

Our objective in this project is construct a model that predicts SPY percentage movement from Trump tweet and SPY price history data. We seek a model that minimizes the squared loss function:

$$L(W) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

where W represents all the model parameters, y_i is the true value of SPY movement, and $\hat{y}_i$ is the model prediction value.

The skeleton of the first two models is inspired by the model used in the Emojify assignment for CS230 at Stanford. Figure 2 shows the schematics of the models used in this work. The text-only model splits the tweet text into a list of word encodings, which are then passed on to two tiers of LSTMs before a small neural network to output prediction. The full model consists of two separate LSTM networks, one for the tweet text encoding input and one for the price history input. Each LSTM network consists of two layers of LSTMs. The final outputs of two networks are combined by concatenation and fed into a two-layer fully-connected neural network with linear activation for the final unit to output the predicted value. Central to LSTM is the memory cells whose contents are passed from one time step to the next and therefore facilitate learning of sequence data. Dropouts are used as a regularization tool to avoid the problem of overfitting the training data. The validation dataset serves to determine whether overfit occurs. The final model shown is based on universal sentence encoder [12] which gives out a 512-dimensional vector that encodes an entire tweet text. The vector is then fed into a neural network to output prediction. Finally, Adam optimizer [13] is used in the back-propagation step to minimize the loss function. Open-source machine learning platforms [14, 15] are used.

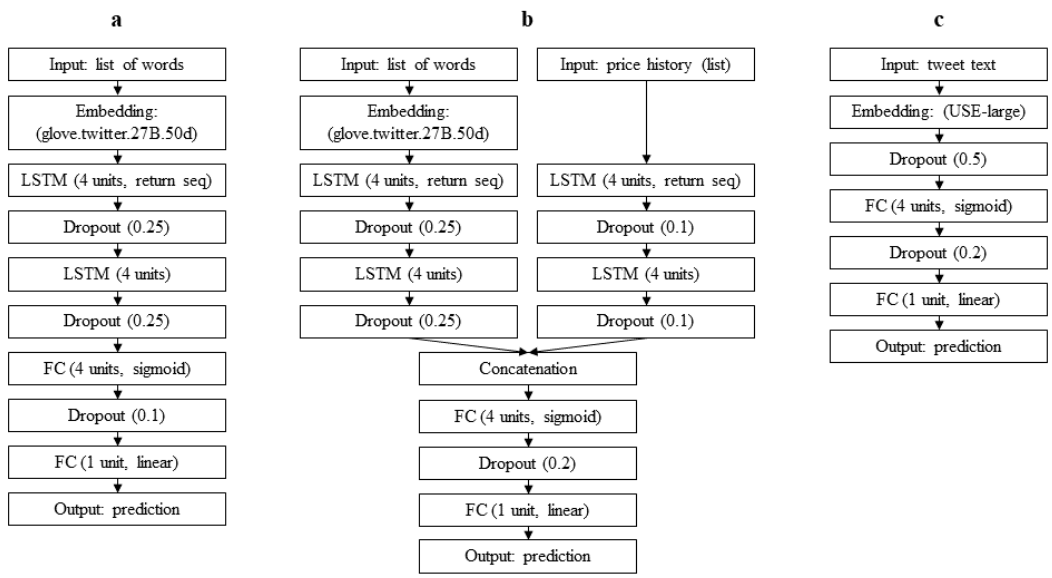

Figure 2: Models used in this work. **a**, Text-only model. **b**, Full model with text and price history inputs. **c**, Sentence-embedding model.

5 Experiments/Results/Discussion/Future Work

For hyperparameters, we vary the number of memory cells in the LSTMs and the number of units for the final fully-connected layer prior to the output, as well as the dropout probability in the dropout units. We tune the model hyperparameters by the following trends: an increase in the number of units in LSTMs and fully-connected layer tends to help with reducing training error but also leads to overfitting of the training data; and an increase in dropout probability helps lessens overfitting at the expense of higher training error. We note here that using sigmoid activations in the fully-connected network parts of the models leads to somewhat considerable gain in training and validation accuracy. This is likely because sigmoid functions can fit nonlinearity in data better. By testing the model on the validation dataset, we can monitor overfitting. We end up with 4 memory cells for all LSTM units, 4 neuron units for the final fully-connected layer, and dropout probabilities of 0.25 for the text LSTM, 0.1 for the price history, and 0.2 for the fully-connected layer. Despite trying several models (including ones not shown), we find that in all models we are unable to use higher number of hidden cells and neuron units without suffering from overfitting.

The final results of the models are shown in Table 2. For comparison, the result of a baseline model which always outputs no-change prediction (i.e. $\hat{y}_i = 0$) is included. Although there seem to be small reductions in the validation errors for all models, the test errors do not reflect this. The baseline

	Prediction error ($\sqrt{\text{MSE}}$, bp)		
Model	Train error	Validation error	Test error
Baseline (no change)	15.13	15.25	15.26
Tweet-text with LSTM	15.06	15.20	15.34
Full (text and price hist) with LSTM	13.91	14.89	15.38
USE with neural network	15.09	15.20	15.31

Table 2: Summary of results.

prediction test error is 15.26 bp. All of the trained models yield relatively similar test prediction errors.

None of the models constructed in this work are able to yield predictions with accuracy significantly above the baseline model. In fact, all trained models output predictions that are typically within ± 2 basis points. The differences in the test errors obtained are statistically insufficient to determine whether prediction errors indeed become worse under the full model. It is known that even a tiny reduction in prediction error can be of benefit in an automated trading algorithm. For us to determine whether statistical significant prediction error reduction of below one basis point would require much more data points.

It is worth pointing out that accurate predictions is not the primary goal of this project. Rather, we try to find out whether a model can discern a SPY-index moving trend based on a restricted set of data inputs consisting of just the tweet text and price history which are available at the time the tweet is released. While it is possible that a larger, more complicated model with additional inputs such as tweets and news from all sources could produce lower prediction errors, it would be difficult to separate the effect belonging to Trump tweet for such a model.

While our results suggest that Trump tweets do not significantly affect SPY movement at a level detectable by the machine models, the effect of president's tweets may show up in the movements of other stocks or sector-specific indices which tend to be more volatile than the S&P 500. This is one area worth exploring in future work. Additionally, a large portion of Trump tweets that occur outside open market hours and all of the retweets are not included in the datasets. This is done because large movements overnight or weekend do regularly take place and are difficult pinpoint the source that causes it. However, finding a framework to bring in these into our dataset while recognizing that several other uncontrolled factors are responsible for the price movement should be helpful for improving the model.

6 Conclusion

In this project, we investigate out whether a machine-learning model can predict the likely movement in percentage point of the SPY price from Trump tweets that occur during stock market hours. We constructed various machine-learning models that take the tweet text and optionally SPY price history during the 60 minutes before the tweet is sent to predict the future movement of SPY price. While the trained models show a slight decrease in prediction error compared with the baseline model, testing with the test dataset reveals the models do not perform better compared to the baseline model with any statistical significance. From this result and our earlier visualization, we conclude that it is unlikely for a machine learning model to yield a prediction much more accurately than the baseline model just from Trump tweet content and price history during previous hour. This suggests that Trump tweets has no significant effect on the S&P 500 stock market index movement.

References

[1] Alec Go, Lei Huang, and Richa Bhayani. Twitter sentiment analysis. *Entropy*, 17:252, 2009.

[2] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! In *Fifth International AAAI conference on weblogs and social media*, 2011.

[3] Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 system demonstrations*, pages 115–120. Association for Computational Linguistics, 2012.

[4] Emily Stewart. The volfefe index, wall street's new way to measure the effects of trump tweets, explained, 2019.

[5] Evie Liu. Yes, trump's tweets move the stock market. but not for long., 2019.

[6] Yuxin Yang Tong Yang. Predict effect of trump's tweets on stock price, 2017.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[10] Trump twitter archive, 2019.

[11] Wharton research data services, 2019.

[12] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[15] François Chollet et al. Keras. `https://keras.io`, 2015.