Stock Price Prediction with BERT and XGBoost using Twitter data

1st Chenyu Huang
SEAS
Columbia University
New York City, US
chenyu.huang@columbia.edu

2nd Shangzi Xie
SEAS
Columbia University
New York City, US
shangzi.xie@columbia.edu

***Abstract*—In recent years, many scholars are using methods based on machine learning or deep learning to predict stock price movement using web-based social data. However, the growing volume of opinionated text and complexity of the market caused by chaotic event interactions, makes it almost impossible to come up with a precise strategy for decision making in the stock market. So as to fix this problem, we proposed an event aggregation model based on BERT and Sentiment Analysis to acquire a better feature representation of the stock movement. By eliminating the redundancy of features and the necessity of iterative computation, our model is evaluated to perform better than several traditional models.**

***Index Terms*—Market Intelligence, Event Linkage, BERT, Sentiment Analysis, XGBoost**

I. Introduction

Our goal of this research is to draw a correlation between web-based events and the movement of the stock price so that we can forecast the stock price with historical data.

As there has been a huge number of literatures studying on algorithms predicting the stock price with either classification or regression, one of our main focuses would be on feature extraction. In other words, how to correctly aggregate similar events so that we can compute the impact of the events as features. We come up with an advanced feature extraction method: Sentiment Analysis Filtering (SAF), which balanced the dataset with respect to sentiment scores so that the successive models can be trained within a rich input space to become more robust.

The paper is organized as follows: Section 2 discusses some related works in Event Modeling, Event Linkage and Impact Prediction. Section 3 illustrated the whole data pre-process. Feature extraction with BERT, POS, etc. are introduced in section 4. Model training and evaluation are discussed in section 5. Section 6 concludes the whole research and section 7 makes some further discussion on the web application details.

II. Literature Review

A. Event Modeling

First of all, we have to model our events so that we can apply a certain metric to find the similarity of events or evaluate the correlation between an event and the market. Graph, vector representation and dimension reduction methods are widely used in the related literatures.

Graph representation is one of the most popular and classical methods to model an event. Disjoint sets [1] have been used to predict the stock price and traded volume time series. N-gram graph is also used along with metrics including value similarity or value ratio [2] to facilitate later text clustering or classification.

A great number of vector-based representations are springing out in the recent times. High-dimensional word embeddings [3] are used to compute the distances between documents so as to represent the similarity. Pre-trained vector model of BERT [4] are proved to be extraordinarily effective in recent studies.

Dimension reduction methods are also widely used to convert web-based social events into a set of labels [5] [6], which represents the classification of the events. Some statistics methods like logarithmic differentiation [7] are also used to map the original event to lower dimensional representation.

B. Event Linkage

From the econometric perspective, some researchers are using Granger causality [7] to draw a relationship between web-based social data and the S&P 500 Index. However, this can only be used to naively accept or reject a relationship. In order to come up with a stronger correlation, K-means clustering [8] can be applied on some graph based representations because of their intrinsic distance property. Cosine similarity [9] is extremely widely-used in vector representations to describe the similarity between events. Semi-supervised learning techniques can even be applied to automatically generate labels according to the cosine similarity metric.

C. Impact Prediction

With a great number of previous studies, impact prediction is considered as a classification problem in the market intelligence field. We are to provide a predictor such that it can tell whether the stock price of the next day will increase or decrease. Classifiers such as SMO [10] and XGBoost [11] are popular in this task because of their outstanding performance. Of course, a more challenging perspective is to consider impact prediction as a regression problem, which requires a much higher accuracy and computation frequency. Support Vector Regression (SVR) are coming into our eyes in the recent times. Among them, a variant of SVR with brain storm optimization [12] outperformed 3 other traditional modals in all evaluation criteria. Other variants of SVR [13], [14] are also providing a feasible and effective option to forecast stock price “continuously”. What’s more, some scholars are thinking out of the time domain and come up with a minute-level price forecasting approach with spectrum analysis [15].

III. Data Preprocessing

The tweet dataset is collected with Twint and stock price dataset is acquired from Yahoo! Finance. We used stock price of $GOOGL. Both datasets share the time slot from 2019/02/06 - 2020/02/06.

The preprocessing procedure of the tweet dataset consists of: 1. UTF-8 Filter, 2. English Filter, 3. Acronym Substitution, 4. Emoji/Emoticon Substitution and 5. Twitter Specific Symbol Substitution.

UTF-8 Filter and Twitter Specific Symbol Substitution (including: URL, hashtag, target, repeated sequence) are implemented with Regex in Python.

Only tweets written in English in considered valid in our dataset, we used a probablistic library pycld2. Emoji’s and emoticons are substituted with alternative texts with demoji library.

Acronym dictionary [16] is reproduced to convert latest slangs into universal readable vocabularies.

Note that all twitter-specific features are not directly removed but maintained temporarily for later use. URL, hashtag, and target are substituted with $\|U\|$, $\|H\|$, $\|T\|$ respectively.

TABLE I
Regex Table

Pattern	Regular Expression
non UTF-8	`[^\x00-\x7f]`
URL	`(http\S+)|((...)*.com(...)*)`
#hashtag	`\#[a-zA-Z]+`
@target	`@[a-zA-Z]+`
repeated sequence with ’a’	`a{3,}`

IV. Feature Extraction

A. BERT

BERT [4], a pre-trained model proposed by Google in 2018 is used by us to do the feature extraction job. We set up BERT as a local service with 1 worker generating 768-dimensional vectors.

Each tweet is converted into a 768-dimensional vector, which denotes an entity of an event or a projection of an event on a specific user. However, different tweets of different users might be describing the same event. To eliminate the redundancy of events represented in our dataset, SVD is applied and only event entities with singular values higher than 15 can be selected for successive analysis.

The validity of the feature generated by BERT is illustrated by the correlation matrix where no single pair of dimensions have a pearson correlation coefficient greater than 0.35.

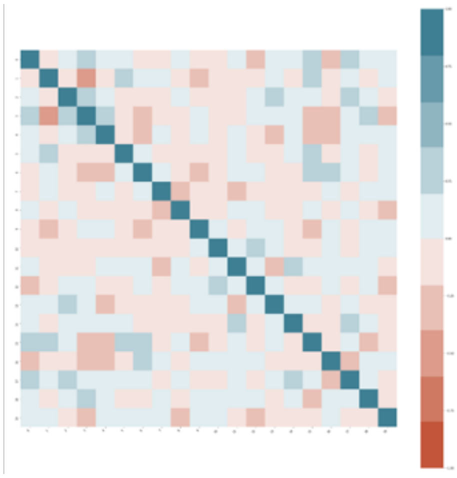

Fig. 1. Correlation Matrix of BERT dimensions (part)

B. Twitter-specific Feature

Due to the 140 words limitation of length of a single tweet, twitter users are active in developing internet-based slangs which can express the same meaning as long phrases with fewer characters (e.g. atst → at the same time). Other similar forms like emoticons and emoji’s are also widely used.

A emoticon dictionary [16] is reproduced so that the occurance of emoticons can be counted as sentiment scores as a feature.

Meanwhile, some twitter-specific symbols are created for interactivity including: #hashtags and @targets. These twitter-specific elements are hard to interpret for traditional machine learning models though they contain a lot of useful information.

C. POS Feature

Among all Part-of-Speech elements, nouns, verbs, adjectives and adverbs are believed to contain rich sentiment polarity in a sentence.

Also, exclamation words and Capitalized words are the words which the author want the readers to pay more attention to, they are also counted along with the sentiment scores to generate part of the features.

TABLE II
Hand-crafted Feature Table

POS	***Twitter-specific***	***Others***
num (+/-) capitalized	num hashtags	num (+/-) words
num (+/-) exclam	num targets	sentiment score
num (+/-/total) noun	num urls	num negations
num (+/-/total) verb	num (++/+/-/–) emoti	num newlines
num (+/-/total) adjective	emoticon score	
num (+/-/total) adverb		
$\sum$ noun sentiment		
$\sum$ verb sentiment		
$\sum$ adjective sentiment		
$\sum$ adverb sentiment		

$^{+}$positive, $^{++}$extremely positive, $^{-}$negative, $^{--}$extremely negative

D. *Price-related features*

Besides, the features discussed above are all related with the orientation of the price movement, but they are not able to interpret the magnitude of the future price. So some price-related features are necessary scale the orientation to a correct value. Those features include: adjusted close price, range of highest and lowest price, range of opening and closing price and stock volume.

E. *Lagged features*

To take full use of the historical data, previous features within a short time window are concatenated horizontally to form a complete feature vector. Here a time window N = 2 is proved to achieve the best performance.

F. *Sentiment Analysis Filtering*

Traditional machine learning models for classification require the dataset to be balanced in terms of labels, but here we balanced the dataset with respect to sentiment class as well so as to ensure the robustness of the regression model. Since we want to learn the stock movement orientation from the sentiment polarity generated from tweets, if the sentiment score of the whole dataset is extremely biased towards positive or negative, or mainly gathered within neutral range, the model will very likely to be overfitting. Originally, the dataset is distributed such that the stock price movements are composed of 55% increments and 45% decrements and over 80% of the m have neutral sentiment score.

Therefore, SMOTE is applied to the dataset so that number of samples of each sentiment class are equal. The process of oversampling the dataset with respect to sentiment class is defined as Sentiment Analysis Filtering.

By running a classification model with XGBoost where stock price increment is labeled as 1 and decrement is labeled as 0, the validity of Sentiment Analysis as a filter is demonstrated.

TABLE III
CLASSIFICATION WITH / WITHOUT SENTIMENT ANALYSIS FILTERING

	Acc(%)
Original	61.2
+Sentiment Analysis Filtering	69.7

Along with 768 dimensions generated by BERT, 32 hand-crafted features and 4 price-related features are used to train the model for predicting the next day stock price. Since lagged features are applied, we have 1,608 features in all.

V. STOCK MOVEMENT PREDICTION

We are to predict the adjusted close price of $GOOGL with all the features provided above.

XGBoost is selected as the regression model with moving average method as a baseline.

In order to revamp the computation speed and stablize the model, price related features are normalized before training and the predicted prices are recovered with the standard deviation and mean value of the previous N prices.

Metrics used to evaluate the performance of the model are RMSE and MAPE. RMSE describes error with respect to the magnitude of the original data while MAPE measures the relative absolute error, combined to be a effective pair of metrics.

A. *Hyperparameter Tuning*

TABLE IV
HYPERPARAMETERS BEFORE AND AFTER TUNING

Param	**Before**	**After**
learning_rate	0.1	0.3
n_estimators	100	270
max_depth	3	3
min_child_weight	1	13
subsample	1	0.4
colsample_bytree	1	1
colsample_bylevel	1	1
gamma	0	0.1
RMSE	13.068	**12.571**
MAPE	0.833	**0.762**

Tuned XGBoost Model provides a better RMSE and MAPE as opposed to the original one. Note that $GOOGL stock price is above $1,000 during the period we selected so the RMSE appears to be large.

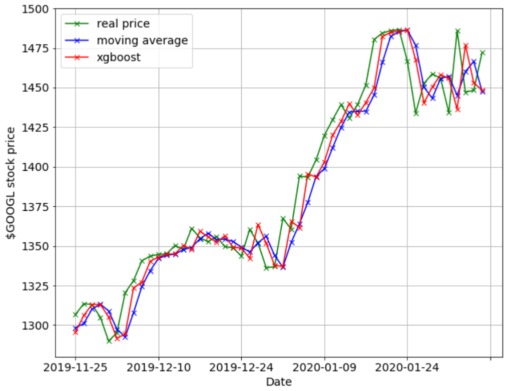

Fig. 2. Predicted Stock Price with XGBoost and Moving Avg.

TABLE V
XGBOOST PERFORMANCE V.S. MOVING AVG.

Method	**RMSE**	**MAPE(%)**
Moving Avg.	17.569	0.919
XGBoost	**12.571**	**0.762**

VI. CONCLUSION

Based on BERT and Sentiment Analysis, this paper proposed a novel feature extraction method: Sentiment Analysis

Filtering (SAF). We validated that SAF is a good tool to investigate stock market with Twitter data. Our research reveals that prediction with 1,608 features filtered with SAF appears to be more stable and accurate than prediction with only price-related features (moving average). However, this paper only analysed the scenario of predicting with historical data, predicting upcoming stock price based solely on historical data would lead to a continuous lag which further leads to a slow response to external impacts. we will conduct a comparative analysis with the introduction of news related to the company happening in real-time to make further understandings on the nature of the stock market in our future research.

VII. Web Application Implementation

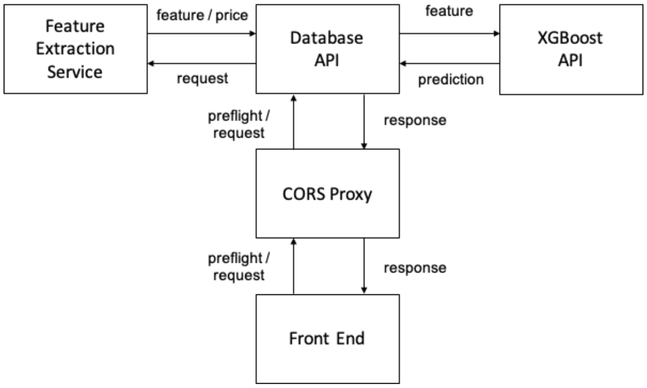

Fig. 3. Web Application Architecture

4 modules are converted into API endpoints. Feature extraction service, database api and XGBoost api are implemented with python flask library while CORS proxy server is developed in node.js.

As a bridge in the architecture, the database api transmits data (feature, real price, predicted price) among all other sections.

The front end is requesting data from the back end once a day through a CORS proxy server which is echoing requests and generating preflight headers. Once a request is received in database API, it either directly response with the expected data or request the feature extraction service to check if there are latest features which then will be passed to XGBoost API to predict stock price. See the github page for more details. (under development)

Acknowledgment

This paper is instructed by Ching-Yung Lin in Advanced Big Data Analytics course at Columbia University. Thanks to my friend Mingze Mao for his crutial suggestions during the research process.

References

[1] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes, “Correlating financial time series with micro-blogging activity,” in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pp. 513–522.

[2] N. Pittaras, G. Giannakopoulos, L. Tsekouras, and I. Varlamis, “Document clustering as a record linkage problem,” in *Proceedings of the ACM Symposium on Document Engineering 2018*, 2018, pp. 1–4.

[3] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, “From word embeddings to document distances,” in *International conference on machine learning*, 2015, pp. 957–966.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[5] B. Dickinson, W. Hu *et al.*, “Sentiment analysis of investor opinions on twitter,” *Social Networking*, vol. 4, no. 03, p. 62, 2015.

[6] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, “Sentiment analysis of twitter data for predicting stock market movements,” in *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*. IEEE, 2016, pp. 1345–1350.

[7] E. Gilbert and K. Karahalios, “Widespread worry and the stock market,” in *Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[8] H.-Y. Yu, S. Park, Y.-G. Cheong, M.-H. Kim, and B.-C. Bae, “Emotion-based story event clustering,” in *International Conference on Interactive Digital Storytelling*. Springer, 2019, pp. 348–353.

[9] M. A. Fauzi, D. C. Utomo, B. D. Setiawan, and E. S. Pramukantoro, “Automatic essay scoring system using n-gram and cosine similarity for gamification based e-learning,” in *Proceedings of the International Conference on Advances in Image Processing*, 2017, pp. 151–155.

[10] K. Reddy, K. L. Shiva, K. Abhilash, and Y. Yoganandam, “Database assisted automatic modulation classification using sequential minimal optimization,” *arXiv preprint arXiv:1806.07566*, 2018.

[11] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[12] J. Wang, R. Hou, C. Wang, and L. Shen, “Improved v-support vector regression model based on variable selection and brain storm optimization for stock price forecasting,” *Applied Soft Computing*, vol. 49, pp. 164–178, 2016.

[13] X. Qiu, H. Zhu, P. Suganthan, and G. A. Amaratunga, “Stock price forecasting with empirical mode decomposition based ensemble ν-support vector regression model,” in *International Conference on Computational Intelligence, Communications, and Business Analytics*. Springer, 2017, pp. 22–34.

[14] J. Zhang, Y.-F. Teng, and W. Chen, “Support vector regression with modified firefly algorithm for stock price forecasting,” *Applied Intelligence*, vol. 49, no. 5, pp. 1658–1674, 2019.

[15] S. Lahmiri, “Minute-ahead stock price forecasting based on singular spectrum analysis and support vector regression,” *Applied Mathematics and Computation*, vol. 320, pp. 444–451, 2018.

[16] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, “Sentiment analysis of twitter data,” in *Proceedings of the Workshop on Languages in Social Media*, ser. LSM ’11. USA: Association for Computational Linguistics, 2011, p. 30–38.