MSBD6000I Text mining
Tencent Stock Opening Price Prediction

Agniva Debnath (20647050),
Cheng Man Ho (20545628),
Hui Kwat Kong (20123133)

Abstract

In this report, the importance of text analysis for stock price prediction is investigated, by introducing the prediction of Tencent company's stock price changes (UP, DOWN). Our results indicate that by leveraging the text information with the text mining techniques, such as BERT pretrained model and sentiment analysis, machine learning model can boosts the prediction accuracy by 5% (relative), comparing with only applying traditional technical indicators.

1. Introduction

Tencent is the biggest stock with the biggest capital and volume in the Hong Kong stock market. This would be highly valuable if one can anticipate the price trend of Tencent. However, as being the largest stock in the Hang Seng Index, there are tremendous factors from Hong Kong, China and the US market affecting the stock price, it is reasonable that we narrow the issue down from predicting the exact future stock price to whether the T+1 opening price will be greater than the T closing price with the effects by news related Tencent (A binary classification).

Our solution is to combine the traditional technical indicators with the market news around the world about Tencent, encoded by BERT and sentiment analysis. We conducted different machine learning model experiments and the final accuracy in testing is 73.6%.

2. Literature review

As suggested from the Efficient-market hypothesis, stock prices reflect from all the available information and their changes react based on new information. Over the years, many researchers try to examine this hypothesis and predict the stock changes by analysing information from text through sentimental analysis or NLP. On one hand, some believe the stock is a random-walk and thus not predictable; while on other hand, some suggest it can be predicted. For those who believe that can be predicted, degrees of directional accuracy at 56% are often considered as a satisfying result for stock prediction (Nguyen et al, 2015 [7]).

Lee et al. (2014) [4] tried to predict the stock price changes in a more traditional way. They formulated the stock trend to 3 classes (UP,DOWN,STAY) and tried to utilise textual information from the 8-K report, which is a financial document legally required in some areas. The document contains information about several kinds of predefined news. Unigram and some matrix factorisation methods are applied to the textual data. The accuracy of the experimental is around 0.55.

In recent years, the research interest starts moving from a purely bag-of-word word-based approach to a content-based approach and tries to analyze the meaning of the whole text. Nguyen et al. (2015) [7] collected data from social media and applied topic modelling methods such as LDA to analyse the topic behind the textual data. In their model, they applied t-1 and t-2 data to predict the t stock price.

Attention also becomes a new focus area in sentiment analysis. Liu et al.(2017) [5] collect news data from websites and apply attention-based mechanisms to generate different weighting to different news features generated from LSTM. The attention mechanism is on an event-based level and comes into play when there are multiple events for a single stock at a single time period. Hiew et al. (2019)[2] applied BERT with LSTM and VAR to weibo post and tried to predict the stock return. They compare the performance for the VAR and LSTM model and analyse the importance of different features.

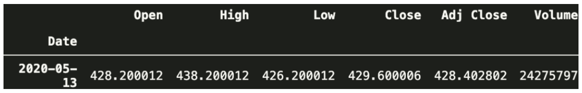
In this paper, we will try to incorporate some financial news articles from different data sources, and apply BERT, an attention-based NLP model, and traditional sentiment analysis method to predict the stock trend changes.

3. Dataset

Based on our topic, both the stock data and the news data are necessary.

3.1. Stock data

We adopted Yahoo Finance API to get the stock data of tencent, HSI, US stock index (NASDAQ, DOW J, S$P 500). The API can extract the opening, closing, highest, lowest price and the volume for every trading day. The following is the data sample of tencent stock price on 13rd May 2020 (Figure 1):

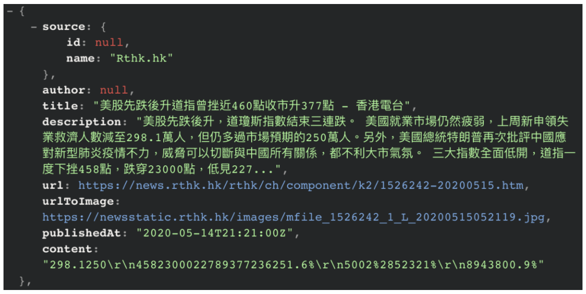


	Open	High	Low	Close	Adj Close	Volume
Date						
2020-05-13	428.200012	438.200012	426.200012	429.600006	428.402802	24275797

Figure 1. Stock data from Yahoo API example

3.2. News data

For the news data, we purchased the newsapi to extract the news data from Nov 2017 to Apr 2020. The news information is extracted by newsapi.get_everything function. Here is a sample data of the news in JSON format (Figure 2):



```
- {
  - source: {
        id: null,
        name: "Rthk.hk"
    },
    author: null,
    title: "美股先跌後升道指曾挫近460點收市升377點 - 香港電台",
    description: "美股先跌後升，道瓊斯指數結束三連跌。 美國就業市場仍然疲弱，上周新申領失
    業救濟人數減至298.1萬人，但仍多過市場預期的250萬人。另外，美國總統特朗普再次批評中國應
    對新型肺炎疫情不力，威脅可以切斷與中國所有關係，都不利大市氣氛。 三大指數全面低開，道指一
    度下挫458點，跌穿23000點，低見227...",
    url: https://news.rthk.hk/rthk/ch/component/k2/1526242-20200515.htm,
    urlToImage:
    https://newsstatic.rthk.hk/images/mfile_1526242_1_L_20200515052119.jpg,
    publishedAt: "2020-05-14T21:21:00Z",
    content:
    "298.1250\r\n45823000227893772362 51.6%\r\n5002%2852321%\r\n8943800.9%"
 },
```

Figure 2. News data from Newsapi example

The API supports many languages and countries, while only english is selected for our analysis. Here the "title", "description" and "content" would be extracted for text analysis.

4. Data Transformation

4.1. BERT Pre-train model

To encode the sentence meaning, BERT is used to provide dense vector representation of each financial news. BERT is currently one of the best models in NLP tasks published in 2018 by Google [1]. The key idea of BERT is to build the representation for natural language by using a bidirectional deep neural network with the Transformer architecture.

With the help of the open-source project, we do not build and train the BERT model from scratch. Instead, we adopt the off-the-shelf model and the pre-trained weight by TensorFlow Hub. According to the official Github, there are multiple variations of the BERT such as different number of layers, hidden and head etc. The model we used is the basic BERT-Base Uncased, with 12-layer, 768-hidden, 12-heads. The weight is pre-trained for English on Wikipedia and BooksCorpus.

Regarding the input financial news, there is not much data preprocessing done to the BERT model's input. Since all the news data are from news articles, it is believed that the sentences should be grammatically correct and each word should be correctly spelled with no special characters such as emoji.

FullTokeniser is provided in the Github repository and applied to each news. The key function of the tokenizer is to convert the words to lowercase strings and to deal with the words that are not in the pre-trained vocabulary list. For the words not in the vocabulary list will be separated by a greedy search. For example: if 'good morning' is misspelled as 'good morning', the greedy search with tokenize the words to 'good', 'mor', '##nni', '##ng'.

The main problem is about the word "tencent". "Tencent" is one of the most important and frequent words in the news, but unfortunately it is not in the pre-trained vocabulary word list. If this word is not handled, the greedy match algorithm will match 'tencent' as 'ten', '##cent'. To minimize the negative effect from this important but out-of-vocabulary list word, we suggested replacing 'tencent' to 'company'. We believe the word 'company' is a good replacement because this word is neutral (not positive nor negative), in the vocabulary list and similar to "tencent".

Due to the input size limitation of BERT (maximum input length of 512), only the title and the description is used for BERT input. It is believed that title and description should be the proper summarisation of the full news content and adequate for the model to encode the relevant information. Also, the title and description are much shorter in terms of token length, this will save some computational time.

As BERT input allows 2 sentences, title and description are treated as two sentences to the model and they are separated by the [SEP] special marker.

BERT's standard output consists of two kinds of vectors: the pooled output and the sequence output. The pooled output is a vector for each input news with the size of [1,768] and it is the representation of the whole news. On the other hand, the sequence output is a vector for each word in each news, i.e. size is [number of tokens of a single input news, 768]. Since our task is about converting the whole news into a dense vector, we used the pooled output from BERT as the input of the later stage model.

Since the output dimension is greater than thousand, PCA is applied to further encode them as 10 principle components.

4.2. Sentiment Analysis

In addition to implementing BERT, we experimented with sentiment lexicons, which were shown to be effective in previous research (Bollen et al., 2010).

We experimented with TextBlob, which has become one of the go-to libraries for performing NLP tasks. It inherits from class 'BaseBlob' is meant for larger text bodies containing many sentences. Due to the input size limitation of BERT, we were not able to perform analysis over the content of financial news. This gave us the opportunity to analyze the sentiments of the content along with title and description for each news.

We used the TextBlob sentiment analyzer with Naive Bayes analyzer, which has a corpus trained on movie review, to fetch the polarity of each part namely title, description and content for every news. We aggregated the polarity of individual news to get the overall polarity of the day. Since the number of news per day was not equal we had to normalise each day's Polarity score, by dividing with the number of news articles for that day. This also removes the biases, as the days with more number of news articles will otherwise have a higher cumulative polarity score.

However, incorporating these sentiment analysis data obtained using TextBlob with technical indicators and BERT output did not improve performance significantly. We had achieved an accuracy of 66% which is even lower than without news data.

We further experimented with class SentimentIntensityAnalyzer of nltk sentiment VADER (Valence Aware Dictionary and SEntiment Reasoner). VADER has been found to be quite successful when dealing with social media texts as its corpus trained on social media text. VADER not only tells about the Positive or Negative categorisation but also tells us about how positive or negative a sentiment is by assigning scores to the categories which makes it quite successful when dealing with social media texts.

We followed a similar process of analysing each component of the news and fetching the Positive and Negative scores for each of them. Since we are categorising the news only into positive or negative sentiments, to account for the neutral component score we half the neutrality and add it to the positive component, since 61% T+1 day opening price is larger than the T day closing and neutral sentiments are prevalent. We aggregated the scores for all the news from a day to get the cumulative score for each day. To handle the unequal number of news of each day and prevent creation of bias, we normalise the same way as discussed previously.

The sentiment analysis data combined with technical indicators and BERT output had shown a significant improvement in performance when modelled with XGBoost classifier was able to achieve an accuracy around 73.6%.

After transforming each news into BERT (PCA) and sentiment outputs, they still cannot be applied to machine learning directly. The number of news is different everyday. Therefore, these BERT and sentiment outputs are aggregated as "mean", "sum", "count" everyday.

4.3. Technical indicators

It is believed that the traditional indicators such as Bollinger Bands, Relative Strength Index (RSI) etc., are also very important in forecasting the stock price. In order to capture the interaction of text news with different technical indicators, we also apply different technical indicators by ta.add_all_ta_features function. To capture the other stock effect on Tencent, we all include the HSI, NASDAQ, DOW J, S&P 500.

4.4. Other transformations

Noticed there are some holidays in Hong Kong that have no stock trading these days. Due to the other market effects, we believe the opening price pattern after long holidays will be different, hence we create a label to capture there is holiday(s) in T and T+1 days. If so, the "holiday" label will be set as 1, otherwise it is 0. The news in the holiday will be passed into the previous trading day (T day), and they will be aggregated to predict the T+1 day. Besides, we also create the month, weekday, day columns to see the seasonality effects.

To capture the trend, we also create more columns from T-1 to T-4 including the information of previous days, it is implemented by pandas.DataFrame.shift method. All in all, we will use the T-4 to T days (in total 5 days, a typical week) to predict the opening price in T+1 day.

5. Modeling Experiments

5.1. Model set-up

Data is split into training (80%) and testing data (20%). The hype-parameters are selected by roc auc score during a randomized search process. Here the "Area Under the Curve" of "Receiver Operating Characteristic curve" (roc auc) is selected because the target (y) is not balanced. (61% T+1 day opening price is larger than the T day closing price). Randomized search is more preferable than grid search in searching the best set of hype-parameters for important features.

Then different model experiments are conducted in the training, while the best set of hype-parameters is selected by cross validation of 3 folds stratified by the target y: (See following table 1)

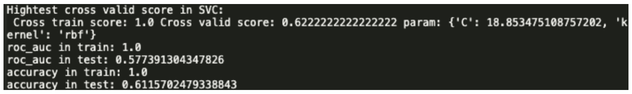
Model Metric	SVM	Voting	LSTM	XGBoost (With text)
ROC AUC (Training)	1	0.799	0.743	0.858
ROC AUC (Testing)	0.577	0.723	0.633	0.738
Accuracy (Training)	1	0.702	0.671	0.781
Accuracy (Training)	0.612	0.669	0.633	0.736

Table 1. Metric evaluation of different models under best set of hype-parameters

5.2. Model Results

5.2.1 SVM

It is extremely easy to over-fit for SVM algorithms even when we are using soft-margin with high penalty as regularization (Figure 3):

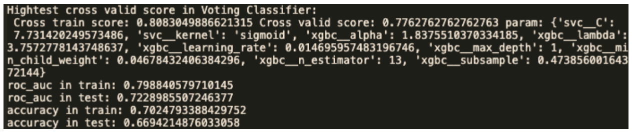


```
Hightest cross valid score in SVC:
 Cross train score: 1.0 Cross valid score: 0.6222222222222222 param: {'C': 18.853475108757202, 'k
ernel': 'rbf'}
roc_auc in train: 1.0
roc_auc in test: 0.577391304347826
accuracy in train: 1.0
accuracy in test: 0.6115702479338843
```

Figure 3. Best hype-parameter & Cross-validation/Training/Testing Evaluation of SVM

5.2.2 Voting Classifier (SVM + XGBoost)

We also try to combine different algorithms, XGbsoot and SVM by soft voting mechanisms. The result is even worse, it may be due to an inefficient observed period. So we do not conduct the experiment of stacking modelling approach (Figure 4):

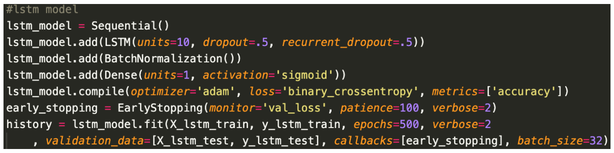


```
Hightest cross valid score in Voting Classifier:
 Cross train score: 0.8083049886621315 Cross valid score: 0.7762762762762763 param: {'svc__C':
 7.731420249573486, 'svc__kernel': 'sigmoid', 'xgbc__alpha': 1.8375510370334185, 'xgbc__lambda':
3.7572778143748637, 'xgbc__learning_rate': 0.014695957483196746, 'xgbc__max_depth': 1, 'xgbc__mi
n_child_weight': 0.04678432406384296, 'xgbc__n_estimator': 13, 'xgbc__subsample': 0.473856001643
72144}
roc_auc in train: 0.798840579710145
roc_auc in test: 0.7228985507246377
accuracy in train: 0.7024793388429752
accuracy in test: 0.6694214876033058
```

Figure 4. Best hype-parameter & Cross-validation/Training/Testing Evaluation of Voting Classifier

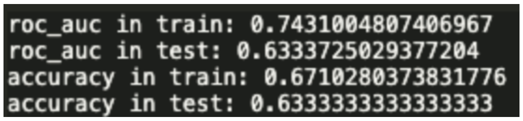
5.2.3 LSTM

When it comes to time series related prediction, LSTM (Long Short-Term Memory) is believed to outperform among different algorithms. However, We found that it is very difficult to develop a sophisticated LSTM model, especially when data is not enough to support a deep learning model (Figure 5 and 6):



```
#lstm model
lstm_model = Sequential()
lstm_model.add(LSTM(units=10, dropout=.5, recurrent_dropout=.5))
lstm_model.add(BatchNormalization())
lstm_model.add(Dense(units=1, activation='sigmoid'))
lstm_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
early_stopping = EarlyStopping(monitor='val_loss', patience=100, verbose=2)
history = lstm_model.fit(X_lstm_train, y_lstm_train, epochs=500, verbose=2
    , validation_data=[X_lstm_test, y_lstm_test], callbacks=[early_stopping], batch_size=32)
```

Figure 5. Hype-parameter of LSTM model setup

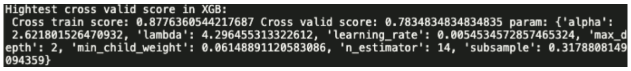


```
roc_auc in train: 0.7431004807406967
roc_auc in test: 0.6333725029377204
accuracy in train: 0.6710280373831776
accuracy in test: 0.6333333333333333
```

Figure 6. Training/Testing Evaluation of LSTM

5.2.4 XGBoost (with text)

Noticed that XGBoost classifier performs the best since it is a powerful algorithm with early stopping which can avoid over-fitting. Overfitting is easily observed in our experiments since we only have around 500 trading days, which is pretty short in terms of economic cycle (Figure 9):



```
Hightest cross valid score in XGB:
 Cross train score: 0.8776360544217687 Cross valid score: 0.7834834834834835 param: {'alpha':
 2.621801526470932, 'lambda': 4.296455313322612, 'learning_rate': 0.0054534572857465324, 'max_d
epth': 2, 'min_child_weight': 0.06148891120583086, 'n_estimator': 14, 'subsample': 0.3178808149
094359}
```

Figure 7. Best hype-parameter & Cross-validation/Training/Testing Evaluation of Voting Classifier

6. Evaluation of XGBoost

In last section, it is found that XGBoost outperforms the rest of the models. We would like to further evaluate it in this section.

6.1. Feature importance of XGBoost

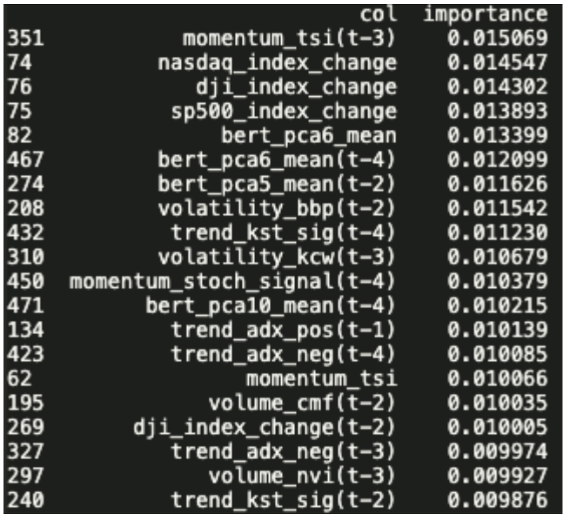
The XGBoost importance shows that some of the important features come from the text related columns (BERT output). It is interesting that the T-1 day columns in text are more important than the T day one. It is believed that the news needs some time (maybe a day) to spread (Figure 8):

6.2. Effect of Text

One may wonder what are the effects without the text data? We also conducted one more experiment without the text information. Noticed that the final accuracy in both training and testing data decreases among 5% while roc auc score decreases in both training and testing data as well: (See following table 2)

6.3. Individual case study

To further investigate the effects from the news, we adopted the “shap” [6], the python package that aims to explain the output of machine learning models by a game



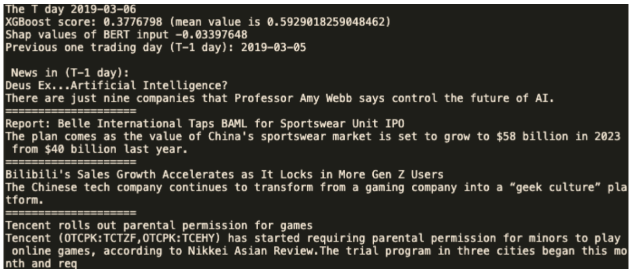
```
                          col  importance
351            momentum_tsi(t-3)    0.015069
74            nasdaq_index_change    0.014547
76               dji_index_change    0.014302
75             sp500_index_change    0.013893
82                  bert_pca6_mean    0.013399
467            bert_pca6_mean(t-4)    0.012099
274            bert_pca5_mean(t-2)    0.011626
208           volatility_bbp(t-2)    0.011542
432             trend_kst_sig(t-4)    0.011230
310           volatility_kcw(t-3)    0.010679
450     momentum_stoch_signal(t-4)    0.010379
471           bert_pca10_mean(t-4)    0.010215
134             trend_adx_pos(t-1)    0.010139
423             trend_adx_neg(t-4)    0.010085
62                    momentum_tsi    0.010066
195                volume_cmf(t-2)    0.010035
269          dji_index_change(t-2)    0.010005
327             trend_adx_neg(t-3)    0.009974
297                volume_nvi(t-3)    0.009927
240             trend_kst_sig(t-2)    0.009876
```

Figure 8. Feature importance of XGBoost model

Model Metric	XGBoost (With text)	XGBoost (Without text)
ROC AUC (Training)	0.858	0.796
ROC AUC (Testing)	0.738	0.725
Accuracy (Training)	0.781	0.740
Accuracy (Training)	0.736	0.686

Table 2. Metric evaluation of XGBoost with/out text information

theory approach. In the training data, we calculate the shape values for all the records. We extracted the lowest (most-negative) shap values in the T-1 day BERT output, that is to say, the news is negative for the stock price predicting, corresponding relatively low XGBoost score. One may check that the news look negatively for Tencent:



```
The T day 2019-03-06
XGBoost score: 0.3776798 (mean value is 0.5929018259048462)
Shap values of BERT input -0.03397648
Previous one trading day (T-1 day): 2019-03-05

 News in (T-1 day):
Deus Ex...Artificial Intelligence?
There are just nine companies that Professor Amy Webb says control the future of AI.
====================
Report: Belle International Taps BAML for Sportswear Unit IPO
The plan comes as the value of China's sportswear market is set to grow to $58 billion in 2023
 from $40 billion last year.
====================
Bilibili's Sales Growth Accelerates as It Locks in More Gen Z Users
The Chinese tech company continues to transform from a gaming company into a “geek culture” pla
tform.
====================
Tencent rolls out parental permission for games
Tencent (OTCPK:TCTZF,OTCPK:TCEHY) has started requiring parental permission for minors to play
 online games, according to Nikkei Asian Review.The trial program in three cities began this mo
nth and req
```

Figure 9. Shap demostration of lowest shap values

Definitely, the final news "Tencent rolls parental permission for games" jeopardizes the game industry and hence its stock price.

7. Conclusion

In conclusion, the texting with our approaches can successfully predict the next trading opening price pattern as accuracy around 73.6%. There are still several limitations and future improvement areas:

- Limited news data: It is better to extract as many years as possible since normal time series related analysis requires several years to see the long term effects. Also, it is desirable to include the chinese news or different comments from stock reviewers. It requires the engines to combine the Chinese and English news together.
- Sentiment analysis and BERT encoding is just an unsupervised approach. That is to say, the meaning “positive” in sentiment analysis does not necessarily mean “going up” for the stock price. It is desirable to train a special encoding approach more related to the pricing issue.
- Finally, all machine learning models require that the past can somehow reflect the future, however it is not always true for stock prediction. For example, during the COVID19 pandemic, the US stock market triggered the circuit breaker several times, following a huge drop for Hong Kong stock, which is extremely hard to predict even with the news data.

References

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[2] Hiew, J. Z. G., Huang, X., Mou, H., Li, D., Wu, Q., & Xu, Y. (2019). BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability. arXiv preprint arXiv:1906.09024.

[3] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[4] Lee, H., Surdeanu, M., MacCartney, B., & Jurafsky, D. (2014, May). On the Importance of Text Analysis for Stock Price Prediction. In LREC (Vol. 2014, pp. 1170-1175).

[5] Liu, J., Chen, Y., Liu, K., & Zhao, J. (2017, August). Attention-Based Event Relevance Model for Stock Price Movement Prediction. In China Conference on Knowledge Graph and Semantic Computing (pp. 37-49). Springer, Singapore.

[6] Lundberg, S.M., Erion, G., Chen, H. et al. From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2, 56–67 (2020). https://doi.org/10.1038/s42256-019-0138-9

[7] Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. Expert Systems with Applications, 42(24), 9603-9611.