PREDICTING SHORT-TERM STOCK MARKET TRENDS USING PRE-TRAINED MODEL ON FUNDAMENTAL NEWS

Shubham Agrawal
(University of Chicago, Financial Mathematics, Email: shubh@uchicago.edu)

Abstract

Stock market movement always remains a topic of research among economists and quantitative researchers. To understand the complex nature of the stock market, researchers are using a pragmatic approach and trying out alternative data and techniques [1]. With the advancement in Natural Language Processing (NLP), it has gradually become an increasingly powerful tool in the Financial world to analyze direct and alternative data. The growing volume of financial data includes companies' financial news like quarterly reports, merger & demerger news, press releases, etc.(hereafter fundamental analysis text), and stock price movement and trends data(hereafter technical analysis text) requires language processing models that can provide text classification and sentiment analysis [2]. Along with the financial techniques, the output of NLP models can be used to create automated trading strategies, portfolio risk modeling, and stock prediction and recommendation systems to name a few. In this project, we are going to focus on creating a stock recommendation system and trading strategy using fundamental analysis and technical analysis text similar to Peng et al.[3] in the Indian equities market (restricted to NSE - National Stock Exchange). NSE is currently the world's largest derivatives exchange in terms of the volume of contracts traded [4]. The problem of analyzing the enormous fundamental and technical data related to the NSE traded companies is a great topic of interest among institutional and retail investors as well as traders. Understanding the company's performance and its stock price through the numbers and training a model to learn the financial context is an exciting as well as a challenging aspect of this problem which has become an important topic of research in quantitative finance. It would be exciting to see how a trading strategy based on the NLP model performs as compared to state-of-the-art strategies for predicting price movement and indices performance which can be used as a benchmark for this task.

Introduction

The project is divided into four major parts. The first part is to get the news text which is published by NSE on its website as per the rules of the Securities and Exchange Board of India(SEBI) for fundamental analysis and get technical analysis text from the financial news sources like moneycontrol. I have used a free channel "Redbox global India" that provided summarised news for Indian equities. It is the Indian subsidiary of First Squawk, hence the quality and reliability of news are pretty good. This will serve as the input data for the model along with the stock prices based on which it is going to predict the stock movement. For the stock price, I have used the yahoo finance API for NSE tickers. The API provides end-of-the-day OLHC data with volume and is free to use. It also provides intraday data but with limited history so I restrict myself to end-of-day data. After getting the prices, I will check for the BTST strategy which is as followed. Any day if we get a signal intraday, we will buy or sell the stock based on that at that particular time and keep this position till tomorrow. Irrespective of the outcome we need to sell tomorrow. Either wait till the target to achieve or stop loss to hit or sell it by end of the day if nothing happened. Since I do not have the data I restricted myself from using the close price on that particular day as the buy price of that holding and using the close price of the next day for the selling price of that holding. However, I missed a lot of opportunities where I could have used the high price of the next day or the low price of the next day as the indicator but it will make the strategy tougher to evaluate. That's why I used the approach that most of the other papers have followed. Also, the reason behind focusing on single exchange news is that the stock prices are pretty correlated to the idiosyncratic news and training the model on the financial datasets provided by Reuters and Bloomberg for US equities like in other research papers[2][3][5] will not give good results. I am using the model of multiple GRU layers with an attention layer on the top with softmax at the end to give the final label. If the efficiency of this project is good this will show strong evidence of using news of a particular market/exchange to pre-trained the model to get better results for similar tasks. For testing, I am using a confusion matrix, accuracy, etc. The final task will be to create a backtesting framework to test whether the model output could be used in the real world to create trading strategies and stock prediction systems. This will help in future research to do a rigorous analysis of the NLP models in financial forecasting. The current analysis of most of the models does not take into account the risk associated with any strategy. These are considered important metrics to accept or reject any quantitative model in finance.

Approach

(a) Data Extraction

I am able to get the data for the stocks. For the news data, I used a free channel name "Redbox Global India" as mentioned which provides precise news from the National Stock Exchange of India Announcement portal and sometimes from other financial channels. The dataset is of more than 3000 relevant news with timestamps of the publication of that news. Some examples of the type of news will be provided below. The other dataset is the mapping of tickers with the company's name. One possible extension could be to use the company's sector included in the model.

A few examples from the news dataset (without the timestamp):

1) Indian Government Has Advanced The Target Of Achieving 20% Blended Ethanol By 2025-Govt Statement #Renuka #Balrampur

2) Balaji Amines: Q4 Cons Net Profit 1.1b Rupees Vs 845m (Yoy)

   Beat Yoy
   Beat QoQ
   Ebitda Beat
   Margins Beat

3) Marksans Pharma: Co-Signs Agreement To Acquire 100% Stake In Dubai-based Access Healthcare For Medical Products

4) Zen Tech: Co Wins Order Worth Rupees 55 Cr

As we can see that the news can be idiosyncratic sometimes which is related to a particular company. But that's not necessary, it can affect more than one company as well. Or it is related to the index as a whole. If it is related to the index, then I decided to map it with the NSE popular index Nifty otherwise I will map it with the company. However, the mapping can not be engineered as it's not structured data and we will need to use machine learning techniques to get the best guess for it. Also, a few types of news will be reported every trading day. This news can be used by the model to predict the Nifty index movement or can be a feature for each stock as well. Few examples:

1) FII Buy Net Rupees 743.22 Cr Of India Shares Today || DII Buy Net Rupees 780.94 Cr

2) NSE Index Provisionally Ends Down 0.38% Or -61.80 Points At 16,240.05

Some mapping examples from equity list dataset:

1) Dollar: Dollar Industries Limited

2) Fortis: Fortis Healthcare Limited

3) GLS: Glenmark Life Sciences

As you can see it's not always the case that the ticker is the first name is the company's name, that's why we will need a map to connect it with companies name. Whenever required, for every ticker I will query the price using google finance API by passing the timestamp for the news that we have in the first dataset. I will also query the price after a certain timestamp says hour or day depending on the time the news has arrived. If it has arrived during market hours I will like to test the hypothesis of whether the price affects the intraday price or not. If it is after the market hour, I will use the price of the next day to test the same hypothesis. There is a total of 1808 tickers including the NIFTY index ticker.

(b) Pre-Processing of the Data and Event Extraction

Using K-means Clustering to extract events

The preprocessing required a lot of effort as it is not structured data queried from a managed dataset. I used the general techniques first to clean the data like lowering the text, and removing the stopwords and punctuation wherever needed. I am not using the lemmatization for now but will use it in case it will be required. Since the news are already very precise using the lemmatization might be redundant. Also, the news contains a lot of financial jargon and abbreviation which I want my model to learn. Then I use the TFIDF vectorizer to vectorize the tweets and create the vocabulary for further pre-processing. The initial vocab size is small (7638) since there are very few words and mostly get repeated in the headlines "Beat", "buyback", "rise", "fall", etc.

I tried the event extraction model suggested by Wenjie et al[7] which is used in some of the other papers that I used for reference. But the issue with that model is it requires a large structured dataset to get trained and each document should have at least 100 words which is not the case with the dataset that I am working on. I manually label the data(it's easier since some words are repeated in the type of news) so that I can use a supervised model as well or test any of the models. After testing I found the unsupervised models like KMeans and SpectralClustering work the best. I went ahead with using k-means as this method can be used to predict the out-of-sample data as well. I checked with my manually labeled data and it turns out to have more than 95% accuracy. Since this is not engineered and a generic model other similar news headlines can also be used. For spectral clustering, I used the cluster number to be 7. The reason behind it is my dataset has more or less 6 types of news. The spectral clustering does a beautiful job to extract all these event types. The type of events(clusters) is mentioned below with an example.

1) **NSE Movement**: This news tells about the movement of NSE intraday or at the end of the day. As mentioned it will be used either to predict nifty index or as a feature for other news.
   e.g.: NSE INDEX PROVISIONALLY ENDS DOWN 0.23% OR -37.35 POINTS AT 16,202.70

2) **FII DII Investment**: Foreign and domestic investment always decide the market movement historically and are used by quant to predict the movement of stocks and index. It will be a good feature to use for predicting the movement of the next trading day as it will be always provided after market hours.
   e.g.: FII SELL NET RUPEES 3960.59 CR OF INDIA SHARES TODAY || DII BUY NET RUPEES 2958.40 CR

3) **Company wins an Order**: This is the first category of idiosyncratic news that will affect an individual company. This news can come between market hours and has huge potential to decide the movement of the stock. It contains mostly positive sentiments and can be used with NSE movement data to predict the movement of the stock.
   e.g. DEEP INDUSTRIES: CO WINS ORDER WORTH RUPEES 72 CR

4) **Merger/Stake Sell**: These are again related to a particular stock/stocks. It can have both negative and positive effects on the stock price. It has different flavors for e.g. sometimes a veteran stock investor buys or sells a stack. Sometimes government does the same. Also sometimes government makes an announcement about public(or public-private firms) about their future. Sometimes it is just some policy change by the government similar to the recent interest rate increase by Fed in the USA. In any case, it has huge potential to give a very short-term or short-term boost to the volatility of the price.
   e.g. Pennar Industries: Co Bags Large Orders Worth Rupees 498 Cr / Dynamatic Tech: Abacus Increase Stake In Co/ As Per Agreement, Ethanol Produced By Dedicated Ethanol Plants Shall Be Sold To Omcs For Blending With Petrol - Govt Statement #Renuka #Balrampur

5) **Quarterly Earning release**: This is again related to a particular company and generally have a huge effect on the movement of the stock price for a short term or sometimes even long term. The price varies a lot because of the good or bad performance of the company in the last quarter. This also has a dividend or bonus share announcement of the company.
e.g. Welspun India: Q4 Cons Net Pft 522m Vs 1.30b (YoY); 631.3m (QoQ) poor YoY poor QoQ

Both method works very well to separate this news. This even extraction will help to decide whether the news is generic enough so that it is mapped to a particular company ticker or linked to the market as a whole and given an NSE ticker.

Nearest-Neighbor to map companies' Tickers

The next task which I forgot to take into account while writing the project proposal was to map news with the stock ticker. It is a challenging task as the data is not structured. Also, it can be one-to-many mapping as we see there is news that has an effect on multiple companies. It cannot be engineered again as it won't give good performance for slightly different headlines and won't be generic enough to extend to a larger dataset. That's why I again use a machine learning approach. I tried different algorithms first like calculating the Levenshtein distance of the strings and then selecting based on that. But that has some drawbacks as sometimes the company names are very similar and it choosing a completely irrelevant company as the right tag. So I decided to use the event tag from the previous task and used it in an engineered machine learning approach. K Nearest neighbors worked out very well for this task. First, I trained the nearest-neighbor model with the company names that I have provided. Then I search in the string if the company name is already present or not, if yes then I passed the string as a predictor for the nearest neighbor and used the least distance(if the distance is lesser than the threshold) For a few events types we don't need to run this step like NSE movement and FII DII investment. We can directly map those to the NIFTY index. For others, I am relying on the output of the nearest neighbor model. This makes it generic enough to handle any type of news in the future. This works out very well and the dataset that I manually labeled with company names performed more than 98% of the time. I restricted myself to 1 company per news to make it simple.

(c) Prices of the company's ticker

Once we have the ticker I extracted the stock price of a particular ticker from yahoo finance as mentioned in the introduction. Since I am checking for BTST signals, I used today's closing price as t and tomorrow's closing price as t+1(keeping the Indian market holidays in mind) The label is basically 1 if the following expression satisfies.

$$\frac{[\ p_{t+1} - p_t\ ]}{p_t} > 0.001$$

The reason behind choosing 0.001 i.e. 0.1% threshold is to keep in mind the transaction fee if entering into the trade. If the return is in between the thresholds I am assigning it a 0 label which is removed later as there are very few occurrences of that. One possible extension is to use this label for neutral prediction with an increased threshold as it could also be very interesting, especially for the options market.

(d) Final Preprocessing

Now we have the text and label pair. We need to preprocess the text to remove all the numbers and company tickers from it along with other punctuations and stop words.

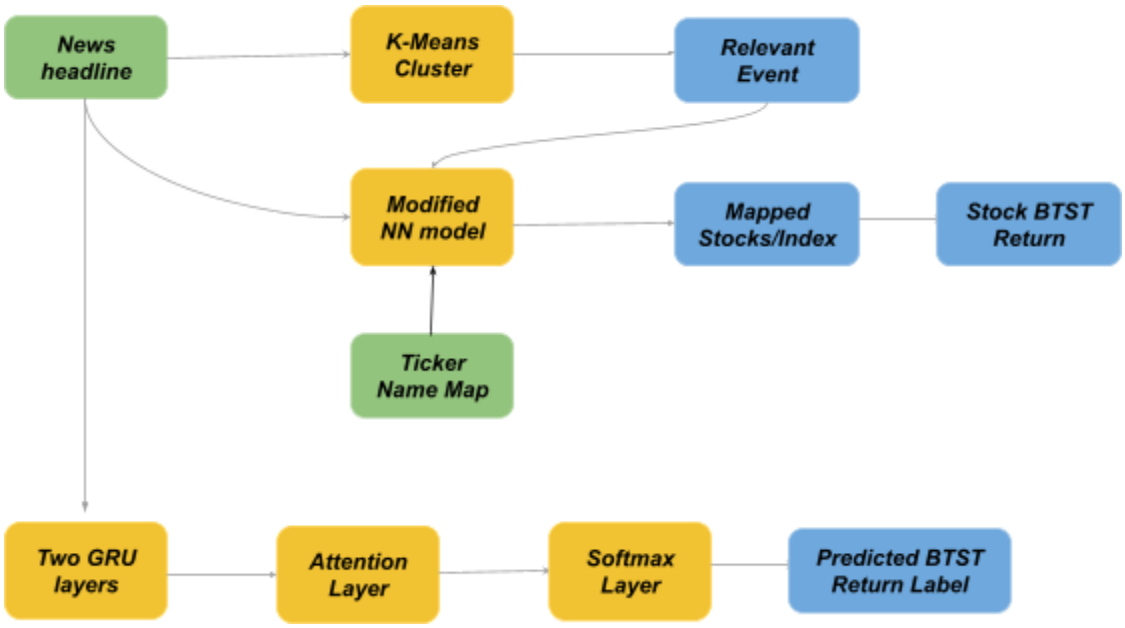
A brief statistics of the data till now:

1) Total dataset: 2509
2) Positive up movement: 45%, rest is negative movement rows.
3) Vocab size: 4538
4) Frequently repeated words: estimate, beat, down, release, up, stack, growth, etc.

(e) Prediction Model

The model that I used is a modification of the ad-char GNN model explained in Wenjie et al[7] The model uses text vector with convoluted layer along with max-pooling layer which is passed in a multiple GRU framework to increase the dimension. The output layer uses the output of GRU layers with softmax to predict the final label.

Since my dataset is pretty small, I decided to use a small model with lesser parameters to learn. The embedding size that i used is 100. I used two GRU layers to start and an attention layer with a softmax layer at the end to get the final label. The GRU has 16 and 8 neurons. The attention layer is a self-attention layer with adding max embedding in a sentence as a residual connection. The final layer has a softmax function as activation and a 0.5 threshold is used for deciding the positive and negative labels.



Summarized Model Steps:

1) Get training and testing data for the model and create a framework for future extraction
2) Preprocess the data so that it can be used for the future models
3) K-means clustering model to extract the events and decide the type of events that needs to be extracted for future use (7 possibilities)
4) Slightly modified NN model to map a news headline to a particular stock (1808 possibilities)
5) Attention layer-based GRU model to predict the sentiments based on the past movement in the stock price.

(f) Experimental Comparison and Analysis:

The model is trained on 80% of the dataset with 20% for training. Since I don't have enough data, I used a large percentage of the dataset for training. The testing and training dataset has a similar percentage of positive and negative labels as the complete dataset.

The %accuracy on the training and testing dataset after the model was trained for 10 epochs was:

Training Dataset: 92.2%
Test Dataset: 61.4%

Confusion Matrix:

	Predicted = No	Predicted=Yes
Actual=No	193	64
Actual=Yes	111	85

With just the attention layer the words with the highest norm were pretty interesting to analyze too.

"great", "growth", "beat", "sold", "Rakesh", "Dolly", "Ashish", "red", "down", "laundering", "war", "resign"

These words convey a lot of meaning related to semantics. Note: words like Rakesh, Dolly, and Ashish refer to Indian investor Rakesh jhunjhunwala, Dolly Khanna, and Ashish Kacholia respectively and the news related to him buying or selling stack in a company matters a lot. Similarly, words like war, laundering, down, and beat definitely convey positive and negative semantics and also effects the movement of stock which can be seen from the dataset accuracy.

One thing to note is that even an accuracy of 60% matters a lot on BTST signal strategy. It is because we are not holding the position for a long time and in this way can make more bets and get compounded returns easily. Also, note that the model predicts the negative sentiment better. Almost 64% which is in line with behavioral economic theory and previous studies. It is also marked by dow jones in one of their primer[9]. Investor generally wants to preserve their money from any downfall and react more quickly to negative news. For positive news, that doesn't necessarily mean having positive returns the next day. It could be a good signal for intraday trading though as generally stock rises very quickly but consolidates the next day to come down to the average value. However, I believe if I have tested it on a larger time period, the score would have improved.

Few of the rightly labeled data from the model with positive movement:

1) ['CO', 'TO', 'CONSIDER', 'INTERIM', 'DIVIDEND', 'MARCH']
2) ['KPR', 'MILLS', 'BUY', 'BACK', 'AT', 'RUPEES']
3) ['SHREE', 'CEMENT', 'COMMENCES', 'TRIAL', 'RUN', 'UNIT', 'IN', 'CHHATTISGARH']

These news mostly have effect for a short term. However, a lot of the news does have positive sentiments but with a "0" label. This shows why we can't use pre-trained models like word2vec for generating such signals. They definitely identify the correct sentiment but the positive effect on stock for short time can only be predicted after training the model.

Few of the rightly labeled data from the model with negative movement:

1) ['SL', 'NET', 'LOSS', 'RUPEES', 'VS', 'YOY', 'QOQ', 'POOR', 'YOY', 'POOR', 'QOQ']
2) ['EUROPEAN', 'GAS', 'JUMPS', 'ON', 'TENSION', 'OVER', 'UKRAINE', 'GAILINDIA', 'BREAKING']

News like this definitely have effect carried to next day as well. I also noticed that in percentage terms(also couldn't do the complete backtesting) it is profitable when we get a negative signal.

Also, a few data that is wrongly marked.

1) ['CO', 'GEARS', 'UP', 'FOR', 'FUTURE', 'GROWTH']
2) ['CONS', 'NET', 'PROFIT', 'RUPEES', 'VS', 'YOY', `BEAT', `ESTIMATES']

Model marked them positive however they turned out to be negative. It is mostly because on that particular day itself the stock rises a lot and then consolidated the next day. That's why as an extension this should be tested on smaller time horizons like minutes or hours.

(Due to lack of time, I was not able to complete my backtested as the intraday data was not readily available for all the days.)

Conclusion

Semantic neural models along with other methods can be used to create a sustainable trading strategy with the decision which mimics humans. The ability to respond quickly and accurately will definitely have added advantage. Also, trading algorithms do not have emotion and react on the basis of the score rather than any preconceived notion or biases. The model presented can be improved a lot which I mentioned in the report and can be extended for options trading strategy as well by including neutral as one of the labels along with positive and negative. Unsupervised machine learning models can be used to extract features for the model and also to create the right mapping with the stock or with the type of event.

References

[1] https://en.wikipedia.org/wiki/Alternative_data_(finance)
[2] Using Structured Events to Predict Stock Price Movement: An Empirical Investigation(https://aclanthology.org/D14-1148) (Ding et al., EMNLP 2014)
[3] Leverage Financial News to Predict Stock Price Movements Using Word Embeddings and Deep Neural Networks (https://aclanthology.org/N16-1041) (Peng & Jiang, NAACL 2016)
[4] https://www.statista.com/statistics/272832/largest-international-futures-exchanges-by-number-of-contracts-traded
[5] From Stock Prediction to Financial Relevance: Repurposing Attention Weights to Assess News Relevance Without Manua Annotations (https://aclanthology.org/2021.econlp-1.6) (Del Corro & Hoffart, ECONLP 2021)
[6] Semantic Frames to Predict Stock Price Movement (https://aclanthology.org/P13-1086) (Xie et al., ACL 2013)
[7]The Study on the Text Classification for Financial News Based on Partial Information(https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9102263) (Wenjie et al. 2020)
[8] Financial News and Tweet Based Time-Aware Network for Stock Trading(https://aclanthology.org/2021.eacl-main.185.pdf) (Sawhney et al 2021)
[9] Dow Jones primer on Sentiment analysis (https://www.dowjones.com/professional/resources/blog/a-primer-for-sentiment-analysis-of-financial-news)