Stock Price Prediction via Financial News Sentiment Analysis

Lizi Chen
New York University
New York, United States
lc3397@nyu.edu

Chun-Yi Yang
New York University
New York, United States
chunyi.yang@nyu.edu

[As of the deadline of paper submission, not all proposed work or results have been finished. Further modification is needed.]

***Abstract*—**

There has been a steady trend in using big data and machine learning techniques to analyze how daily media affects the stock market. Upon takeaways from researches in macro market analysis and short twitter text analysis, we conduct a more granular stock and news data analysis. With by-minute U.S. stock market prices in addition to Reuters and Wall St. Journal news scrapped in full details for 45 days, we have a stock price predictive model trained from historical stock trend, news keyword, news content, and news title. As of August 7th, 2017, our result from elementary experiments shows 84% accuracy in predicting a 1% rise or drop on stock price from news publication time to seven minutes after. More results and models will be added in the future work.

Keywords—spark, big-data, sentiment-analysis, stock-prediction, supervised learning.

I. INTRODUCTION

The Big Data era has prospered the financial industry for many years. Banks and hedge funds now can fully leverage their huge amount of data to construct their robust strategies. Likewise, as the software and hardware technology springing up in the big data field, the capability of using data rocks even further. The 2014 released Apache Spark framework has already been gradually taking the preeminence over its predecessor Hadoop. Its efficiency and capability have been canonized from technology to finance industries and even helps the evolution of Fin-Tech industry and Machine Learning.

The busy industry Fin-Tech and many of its upholding hedge fund companies have been pouring effort to improve the integration of big data and machine learning technology. In this project, we find the relationship between news that are published in popular business media and the stock price trend. With the drastic improvement of data process capability from Spark, we collect full text news and intra-day stock prices for all the U.S. stocks traded from NASDAQ, NYSE, and AMEX. By having news and stock prices mapped by exact minute timestamp, we provide high-granularity insight into the complex relationship between these two.

We construct our model according to popular algorithms: such as TF-IDF, Logistic Regression. *Conclusion on how these models perform in prediction will be added in future work.*

II. MOTIVATION

1. Being able to selectively consume a large amount of information efficiently can be a major dominance in financial market. There are thousands of newly-published articles from various websites every day, and that can be millions of words in a lot of ways of narration. Investors have to choose from resources, such as Reuters website; then search for reports and articles that may relate to his or her portfolio positions and investment plan, then read and understand the information from these articles. This process will take a lot of time and confuses the investors in many ways. Although there are investors who trade mostly based on stock data, charts and statistical analysis, getting a gist from the most recognized media adds more security to foresee a stock trend.

2. Most of news come with a topic, or a simple narration of an event, *[A subject] [B predicate] [C object]*. Such pattern helps train algorithmic models efficiently. News will provide more detailed information yet the theme can be established in a much shorter paragraph. Although there are better machine learning algorithms that can provide read-able abstract paragraph based on a long article, we believe that a more concise indicator in a pattern of *[A stock] (will) [Drop / Rise]* is more efficient.

3. There have been a lot of stock prediction researches leverage social media, such as Twitter and Yelp, yet the training data were mostly EOD (End-Of-Day) stock prices. Many events have shown direct impact on related stocks after several hours if not minutes in the same trading day. Our hypothesis believes that by having the news and stock trend matched by-minute level can add versatility in model training as well as provide better predicative result.

4. Although social media have been increasingly reflecting and influencing behavior of other complex systems; such as the stock market, the collected data is very scattered. News website provide more main stream comprehensive narration from a less objective point of view.

III. RELATED WORK

The past researches on stock prediction can be categorized into two major approaches. In technical analysis, investigators use only the stock price data series itself to predict future price [4]. In fundamental analysis, macroeconomic information such as the sector a company is in, the information of a company itself, is fully considered to predict stock price [5].

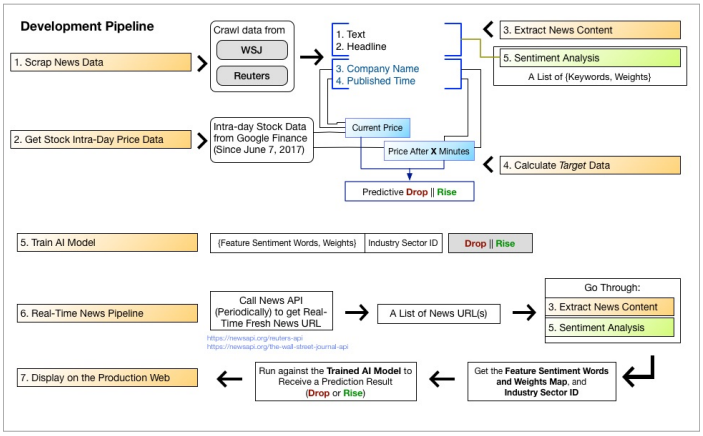
The tones and words used as content in the financial news are correlated to the performance of stock markets in several aspects (Robert et al. 2012). For example, through sentiment analysis on the financial articles, the negative sentiment articles perform better on positive ones on price direction trend (50.9% versus 50.0%). Moreover, an interesting finding is that the negative articles have 52.4% of time accompanied with the price increase as well as the positive articles have 53.5% of time accompanied price decrease. The authors concluded that it might be caused by the contrarian trading strategy of traders. [2]

A full-around survey about sentiment analysis algorithms, which was conducted by Walaa [3] provides insight on how each algorithm fits to certain scenario and which model provides better results. Within 9 models, the Neural Network with Deep Learning gives the best result.

IV. DESIGN

Since the detailed data; full content news and intra-day minute-level historical stock prices, is neither public nor free throughout the web, we collect all our data from scratch. Below is the development process diagram, which contains 7 major steps.

1. Scrap News Data:



2. Get Stock Intra-Day Price Data:

To construct the model, we need two sets of data: Intra-day stock price data, and News from WSJ and Reuters. The use of it is described in the following.

3. Extract News Content: For each scrapped news, we have to extract useful information from the meta data by using Python's Beautiful Soup package. One piece of news will provide the following six feature data:

- Published Date
- Title
- Keywords *(May not be shown from all webs)*
- Sector *(May not be shown from all webs)*
- Content
- URL

Not all websites have these meta data written in the web HTML, but if so, our code will scrap them.

4. Calculate Target Data: For each news, there can be one or many related stocks. We will first extract the company name text from the news, then map to the ticker symbol used for its stock. With a ticker symbol and news publish time, we can call the historical stock price database to query the stock price at the time the news was published, in addition to prices at any time after that, 7 minutes, 15 minutes, 30 minutes, 1 hour or further. By comparing the current price of the stock and its future prices, we can see the trend of it after the publication of news, either *'Rise'* or *'Drop'*.

5. Train AI Model: Now that we have all target values calculated, we will have a pre-training set of data, of which the schema is shown below:

TITLE	CONTENT	DROP_RISE

After the sentiment analysis process (which is described in the next paragraph), the training data set schema is shown below:

Weights of Word Feature (80k attributes)	DROP_RISE

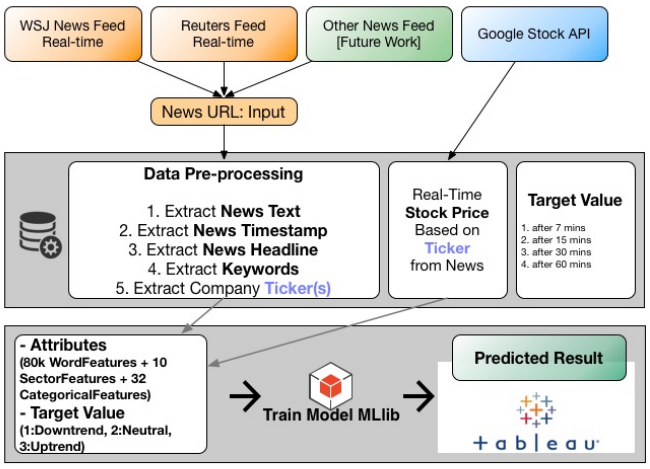
Because we experiments the range of time that a news may need to incite so as to influence a stock price, the actual training data set includes four DROP_RISE columns, each represents the trend after 7 minutes, 15 minutes, 30 minutes, and 60 minutes.

[…]	Trend7mins	Trend15mins	Trend30mins	Trend60mins

We will consider the news title and body content altogether in this case, since most of the news have both of these two attributes. The publication time, keywords, sector, and related company names and stocks are not part of the training data set but to determine the target value and to do the sentiment analysis job.

Sentiment Analysis: as part of the model building process, we need to analyze the sentiment for each piece of news. *As of July, 2017*, the quantity of unique words from all the collected news are 80,000. Therefore, we construct our training data set by having 80,000 attributes for each entry. Each attribute represents the weight of a word.

After having the model trained, we can feed the model with news pieces; however, it's better to provide real-time function that helps future investors to access the indicators directly from a web browser. Thus, we add steps 6 and 7 as shown in below:



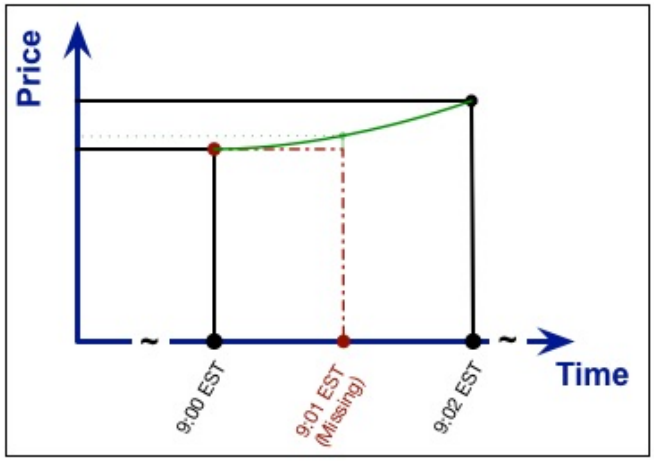
6. Real-Time News Pipeline: Adding real-time news and stock data streaming as well as automate the whole data collection and model training process will help eliminate post-production maintenance work. This process begins with periodically calling the newsapi.com asking for new JSON data from target websites. For each newly published article webpage, a webpage extractor will collect information for the trained model so as to display a real-time predicative result. After the ground-truth comes out after 20 minutes, 1 hour, or 2 hours, we will compare it with the predicative result generated previously so as to improve our model.

7. Display on the Production Web: Below is the diagram for the components of the final product. We provide visualized web interface for clients to analyze and customize according to their portfolio interests.

V. EXPERIMENTS

1. Data collection: The data is divided into information data and updated data by different functions. The information data such as company attributes are collected from Intrinio.com as well as the words of company alias are gathered from CityFalcon.com. As for updated data like stock prices and financial news articles, the stock prices data are fetched on the minute basis, starting from 10:00 to 16:00 because we wait for one hour from 09:00 to 10:00 let the stock price digest the over night information. On the other hand, by utilizing the News API, a multi-media headlines metadata API, we check the WSJ (Wall-Street-Journal) and Reuters API every 5 minutes and apply crawlers to scrap the website content and timestamp we need.

2. Data processing: We first dealing with the missing value. Companies lacking information attributes are marked as Null. Next, to trace the influence of the articles to the stock price, we mark the company ticker to financial articles by pairing the articles keywords to bags of words of company alias names. All data are categorical and stored in JSON format. The intra-day stock price collected from Google Finance API has its limitations. Although most of stock prices have changes every minute, Google will eliminate the ones that do not change for a continuous minutes. As illustrated in the sample diagram below, the stock price at 9:01 EST is the same as its previous record, 9:00 EST. Google will eliminate all the same records; in this case, the one at 9:01 EST. Thus, we have to mock up the missing-minute stock price. Current mock-up approach is to have all the missing record be the same as its previous one, as shown in the red dotted lines. Future work has to optimize such mock up data by creating a linear-fashion fake data, as shown in green line in the diagram. This will simulate the real-world situation more closely.



3. Data generation: Before generate data features, we need to know what company information does each article contain and used it to map stock price. There are 2 data mapping processes in our dataset. The first is article mapping with company ticker. A keyword list of each article from internet combined with a 4-gram word list from it's title is generated. Then we utilized a company alias name dictionary from CityFalcon database to mark company ticker to each article. The second the mapping the articles to stock price. The company ticker mark from last process is used to position its own stock price.

4. Feature generation: The data input attribute consists of context features and related company information. The TF-IDF (Term frequency-inverse document frequency) in Spark MLlib is applied on article content to generate weights of word features. As the result, a 80k feature vector from TF-IDF is represented as article property. The other features such as the company category, group, and sector fields information come from intionio.com database. The output is divided as 3:4:3 fold by the definition of price difference of target time and current time. If the price difference is less than 1% than we mark it as 1 standing as downtrend, greater than 1% we mark it as 3 standing as uptrend, and others marked as 2 standing as neutral. Therefore, a supervised algorithm can be applied.

5. Modeling: Total of 3,552 sample were collected in this experience. They are split by 80/20 as training and testing set. We applied the logistic regression as the training model and a 5-fold cross-validation. We choose F1 score as the evaluation baseline because it take a good balance of *precision* and *recall* of the test to compute the score.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

6. Result: The comparison of different target's F1 score is list as below.

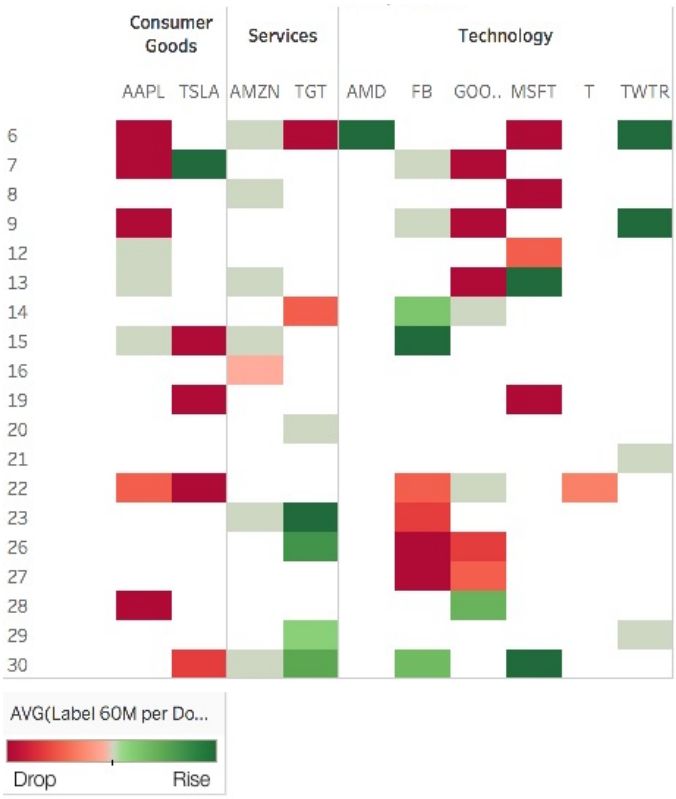
	7 mins	15 mins	30 mins	60 mins
F1	0.8426	0.7498	0.7853	0.6804
Precision	0.8623	0.7823	0.7882	0.6352
Recall	0.8623	0.7823	0.7882	0.6352

The best parameter setting of logistic regression in MLlib are ElasticNet: 0.0, regParam: 0.1

VI. CONCLUSION

From the comparison table shown in the experiment section, the 7 mins target has the best F1 score of 0.8426 among the four. This results indicates that the stock price reflects immediately after the press is released, which authenticate our assumption of direct impact of financial article through stock price.

By using Tableau, we can visualize our stock prediction with a more user-friendly interface, as shown below. Tableau takes in result JSON file and read its attributes by names. The sample heat map diagram below shows some of the company tickers that has highest exposure along many news feed. In the diagram, the level of red implies tendency of the stock to drop, and the level of green-ness; in the opposite, implies a potential of rising. By connecting Tableau with remote Spark server, Tableau enables visualizing near real-time prediction head map.



Our hypothesis has been affirmed with preliminary experiments, further complex models are yet to be constructed in further work; apart from which some other variations of trials can be added to improve precision of the models.

1. Current target value in the training dataset is computed from the average price of the open, close, highest, and lowest. We have not take the volume into the training.

2. Current Spark MlLib does not support PCA with over 65,535 features; our training set has a word vector which contains over 80,000 weights, we will come back to PCA if needed when Spark MlLib supports so.

ACKNOWLEDGMENT

1. CityFalcon - The company information to create Alias of Tickers dictionary
2. Intrinio - The company's basic information such as sector, category, and group
3. NYU Hight Performance Computing Group - Thanks for resource and technical support
4. Suzanne Mcintosh - Special thanks to class instructor Suzanne Mcintosh's guidances and support of this project.

REFERENCES

1. T. White. Hadoop: The Definitive Guide. O'Reilly Media Inc., Sebastopol, CA, May 2012.
2. Robert S, Yulei Z, Chen-Neng H, Hsinchun C. 2012, Evaluating sentiment in financial news articles. Decision Support System 53 : 458-464
3. Walaa M., Ahmed H., Hoda K., 2013, Sentiment Analysis Algorithms and Applications: A Survey.
4. Edwards, R. D., Magee, J., and Bassetti, W. H. C. Technical analysis of stock treads. CRC Press, 2007.
5. Thomsett, M. C. Getting started in fundamental analysis. John Wiley & Sons, 2006.