# 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

***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. [PROVIDE RESULTS FROM THE MODEL LATER]**

***Keywords—spark, big-data, sentiment-analysis, stock-prediction***

## 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 it 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 various algorithms: TF-IDF, Word2Vec, Neural Network, [**ADD FURTHER WORK**]. After comparing and contrast with various predication times, we conclude that [**ADD RESULT**].

## II. Motivation

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 then read and understand the information from these articles. This process will take a lot of time. 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.

Most of news come with a topic, or a simple narration of an event, *[A subject] [B predicate] [C object]*. 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.

There have been a lot of stock prediction research with the help from 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. We believe by having the news and stock trend matched by minute level can add versatility in model training as well as provide better predicative result.

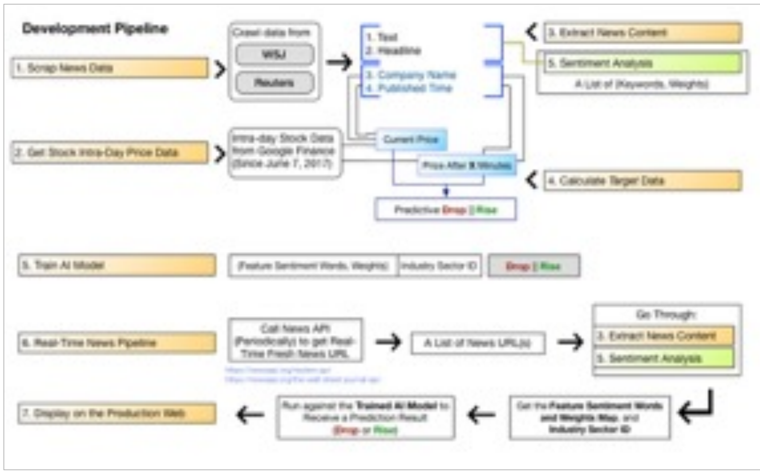
Although social media have been increasingly reflecting and influencing behavior of other complex systems; such as the stock market, the collected data is very scattered. News website provide more main stream comprehensive narration from a less objective point of view.

## III. Related Work

[**RELATED PAPER SUMMARIES WITH REFERENCES**]

## 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, 20 minutes, 1 hour, 2 hours 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 |
|---|---|---|

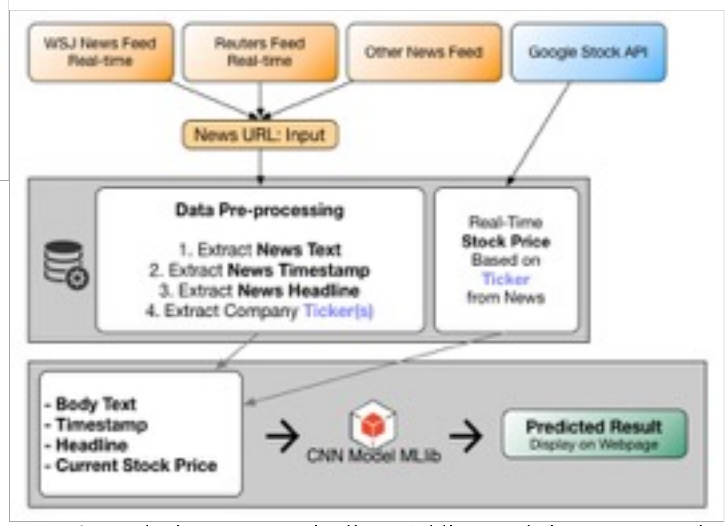
After the sentiment analysis process (which is described in the next paragraph), the training data set schema is shown below:

| Weights of Word Feature | DROP_RISE |
|---|---|

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. **[DESCRIBE THE FINAL SENTIMENT ANALYSIS APPROACH HERE AND REASONS]**

After having the AI Model training, 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

(In this section, you can describe: *Your experimental setup, problems with: data, performance, tools, platforms, etc. Discuss your experiments, describe what you learned.* Discuss limitations of the application. Discuss what you would do to expand it given time - how would you improve it, etc.)

## VI. CONCLUSION

(One paragraph about the value, results, usefulness of your application.)

## ACKNOWLEDGMENT

(This section is optional. It can be used to thank the people/companies/organizations who have made data available to you, for example. You can list any HPC people who were particularly helpful, if you used the NYU HPC. List Amazon if you used an Amazon voucher.)

## REFERENCES

1. T. White. Hadoop: The Definitive Guide. O'Reilly Media Inc., Sebastopol, CA, May 2012.