Correlation Analysis Between Multiple Resource and Stock price

Weihao Bi

December 2019

Abstract–As we all know, emotion plays a tremendously important role in decision making. Does this also impact trading? Especially stock trading. Trader in investment bank may can control their emotion to avoid effect from it, but most of participant in the market are not trader. This project aims to analysis Multiple resources from several open data sources and find out if stock price can be reflected from all of them or part of them. This project may include: NLP (sentiment analysis), Machine Learning, some basic finance concept, spam filter
Index Term– Stock Price – Twitter – Google Trend – sentiment analysis

1 Introduction and motivation

1.1 Twitter Analysis

1. Twitter offers full-scale API to obtain tweets, both for historic tweets and real-time tweets stream, it is convenient to fetch reasonable tweet.

2. For historic tweets, we can set proper filter to fetch tweets with particular keyword or language within a time range.

3. For real-time tweets stream. It established a connection to the streaming APIs, which means making a very long lived HTTP request, and parsing the response incrementally. Conceptually, you can think of it as downloading an infinitely long file over HTTP.

4. Since stock market has recess each day and each week, tweets within the closed time also play an important role for upcoming trend.

1.2 Google Trends Analysis

1. The trends should have the same scale with twitter and stock price

2. Trends don't have direction, which means when trends rise up, that only can reflect the stock may have a incident happened, but we cannot determine how it impact the market.

3. Google Trends won't give a fixed value for determined time. On the contrary, it return a relative number within a period, this may constrain us to locate only top-k time nodes rather a threshold for it. Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means there was not enough data for this term. -Google

1.3 Addition of Universal Factor

1. As we consider the world as a complex data set, we want to predict everything base on past data and pattern, we need more and more data no matter if it is related to our goal due to some under layer correlation between not the least concerned things.

2. In this case, to produce an easy tool to add any kind of data into predictor only if they are in same time series with market price,

3. Because we could use deep learning to arrange different weight for different feature, we also have a bunch of way to pruning trivial feature and select combination of features. So the question can be narrow down to find a way to automatically format data set with market price.

2 Background of Topic

Many economist have argued that the stock market is random because it is governed by random events.

There are two articles to proven it: Efficient Market Hypothesis and Random Walk Theory

But researchers have a strong faith that stock market can be predicted and know how the market will go. One of the mainstream paper is Twitter mood predicts the stock market by Huina Mao and Quantifying Trading Behavior in Financial Markets Using Google Trends by Tobias Preis, Helen Susannah Moat & H. Eugene Stanley.

In that paper, Huina used public tweets and prove that there was a correlation between the moods of public expressed on twitter and the way the stock market performs.

To understand this topic more comprehensive, we need to clear several things:

2.1 What is stock market?

The stock of a corporation is all of the shares into which ownership of the corporation is divided.[1] In American English, the shares are commonly called stocks.[1] A single share of the stock represents fractional ownership of the corporation in proportion to the total number of shares. This typically entitles the stockholder to that fraction of the company's earnings, proceeds from liquidation of assets (after discharge of all senior claims such as secured and unsecured debt),[2] or voting power, often dividing these up in proportion to the amount of money each stockholder has invested. -Wikipedia
Stock all be defined as:

A stock market is the aggregation of buyers and sellers (a loose network of economic transactions, not a physical facility or entity) of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange, as well as stock that is only traded privately. -Wikipedia

As we all know, there are dozens stock exchanges in the world, in this case, we focus on New York Stock Exchange(NYSE).

2.2 Twitter Sentiment Analysis

Twitter is an American microblogging and social networking service on which users post and interact with messages known as "tweets". Twitter was created in March 2006 by Jack Dorsey, Noah Glass, Biz Stone, and Evan Williams, launched in July of that year. The service rapidly gained worldwide popularity. In 2012, more than 100 million users posted 340 million tweets a day, and the service handled an average of 1.6 billion search queries per day. -Wikipedia

Thanks to this huge platform, we could easily find correlated tweets with certain stock. For example, key word: 'AAPL', which is a stock code for Apple inc. You can use advance search to search tweets and get more than one thousand tweets for each day.

So, with those tweets, can we do the sentiment analysis by ourselves? The answer is absolutely no. But can machine know the mood for each tweets? Partially no, we now can use nltk to get a rough mood for particular text or article, but is it correct all the time? No. Using machine learning we can get a fine model to analyze the text. So, to refine the accuracy of sentiment analysis, we can transform this question to how to make a reliable sentiment analysis model.

In the paper Twitter mood predicts the stock market, they used The Dow and general tweets for market, which means they focused on macro stock market, and how all participants in the market react for it. They get a conclusion that individual's emotion or individual's invest can affect The Dow, which means this is a positive correlation between individual's bid or ask. With this conclusion, this project are going deeper and explore the correlation between individual's motion for particular company and the price of that company.

2.3 Google Trends Analysis

Google Trends is a website by Google that analyzes the popularity of top search queries in Google Search across various regions and languages. The website uses graphs to compare the search volume of different queries over time. Google Trends also allows the user to compare the relative search volume of searches between two or more terms. -Wikipedia

Nowadays, Google allows people to get information they interested in. That is, in a huge scale, a society trends, which can reflect a motion among whole internet. In the paper Exploring the relationship between Google Trends data and stock price data by Dartanyon Shivers, it has a good correlation between certain word with stock price. More precisely speaking, some stock correlated to some keyword rather than his company's name, for example, Hasbro(HAS) has a strong relation between word 'toxic' in certain time period. Due to some bad news for toxic toy report for Hasbro, their stock price has dropped sharply within four weeks. At the meantime, 'Hasbro toxic' started to get a top search word in google trends.

Back to this project, the target for this project is not to find the word that impact company's stock price hugely, but find a general pattern for detect a trends and investigate it.

3 Dataset Description and Collection

As I mention in the title, focus on intraday data is a cost way to analyze, since mainstream finance platform usually only provide daily stock price data rather intraday data. The solely way to get it is purchasing from some website. Fortunately, one post his data to Kaggle, but only for a range of data and limited stock. Because analyzing real-time data is a challenge for now, and do research for past data also can product a meaningful result. So it can summarize two main source for dataset, one is:

3.1 API

1. Twitter API By creating a developer account in twitter, we can use restricted API and rate from Twitter.
2. Google Trends API There is no official API for Google Trends, we can find, however, plenty of third-party library for it.
3. Other API Because we want to build a extendable model, hence, we may add more API later.

3.2 Existing dataset

1. Intraday price data Download from Kaggle. The data contain eight columns: 'Date,Time,Open,High,Low,Close,Volume, OpenInt'.
2. other data

Until now, we have all access to fetch all data both from Kaggle and those official website.

4 Analysis Methodology

Analysis happened in distinct stages which we have explained below. The idea is to explore the data lifecycle from collection to practical usage in insight generation

4.1 Stock Price Data

4.1.1 Data collecting

Download from Kaggle the owner is Boris Marjanovic and gave access to all to use this data.

The data The dataset (last updated 12/06/2017) is presented in CSV format as follows:

- Intraday data: Date,Time,Open,High,Low,Close, Volume,OpenInt
- Daily data: Date,Open,High,Low,Close,Volume, OpenInt

It offers us three scale data, 5 minutes, hourly, daily. Although we choose to use hourly data as our time scale, we still need to use 5 minutes data to compress the hourly price change. To be more specifically, if we use hourly data, we will get open high low close price each hour, what can be compute is change in each hour, the first hour has 10 as high 9 as low, so the maximum change is 1, but the second hour has 8.5 as low 9 as high. If we simply use hourly data, we cannot seek the maximum change as it is possible 1.5 from 10 to 8.5 during one hour.

It can be seen from this, we should choose 5 minutes data and do a roll window for data to locate the maximum change within one hour.

We also count volume as an important index for interpreting the market. Use the same principle, we just sum each 5 minutes volume in past 60 minutes and get the total volume for past one hour.

4.1.2 Data Cleaning

Data Cleaning is performed in order to remove inaccuracies and make the data consistent in format. Some of the steps we undertook are:

- Standardization
  The date time for the data has 6 hours offset, so we need to minus 6 hours for every index Change in past 60 minutes will return a exact number for change per share, but different stock has different price per share, so this requires to transform the absolute value to relative percentage value.
- Normalization

  Columns with numeric values are highly susceptible to a flaw in machine learning algorithms which give higher priority to columns containing larger numbers. This automatically increases their importance and leads to a fundamental flaw during the modelling phase. We use standard scaling to center the data around the mean and transform it to have unit standard deviation.

  This allows our algorithm, during the modelling phase, to better understand feature importance and interactions with the target variable
- Missing Values

  Because we use rolling window to detect change in price, so at beginning of data will return NaN due to no previous data to compute past change. So we just remove all NaN.
- Remove and Combine Columns

  The original data set contain eight columns, for future convenience, we have to combine 'Date' and 'Time' to a new column 'DataTime'. The column 'OpenInt' denote the interest before the market open, at this time, we do not take account of this element and eliminate this column.

4.1.3 Data Modeling

To generate a appropriate model, this require to find a proper label for model. We would like to find correlation between some factors and the price of stock, so the label should be the change of price.

There is another question appeared when to transform price to direction, we collect the min-max price within one hour, but the difference between those cannot give expression about the trend of price. At here, using difference value between close price and maximum price in previous can reflect the direction. The principle is:

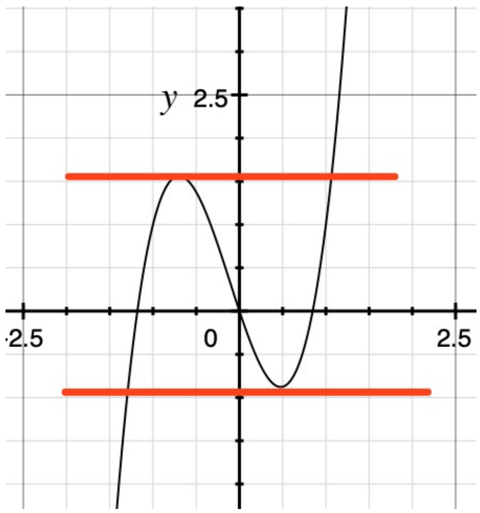

Figure 1: Price trace

The close price only can plot within two red line if we point maximum value and minimum value as two local peak. And in this case, any value at right line always lower than maximum price, hence, the overall change is down because the hugest change among two red line is from local highest point to local lowest point.

The next thing need to be modeled is to deal with Closing market. As is well-known, the stock market will close everyday and open in next day, so the day has a significant gap between two days. We have to consider it as a specific pattern, the reason is most of financial report or market news happen randomly and usually not in the opening time of market. So at that time, emotions will build up and be released at the moment when the market is open. That why we can see most of huge changes happen at the beginning of day. Thus, we create a new column called 'if_gap'. With this feature, we can classify that if this change happened between two days or it is intraday.

At this point, we created a label for stock price data. The next step is process other training data.

4.2 Twitter Sentiment

4.2.1 Data collection

python-twitter is a extension library for twitter API, it is feasible to only use official API to fetch data from Twitter, but not the best approach to achieve that. With python-twitter, we could simplify the code and speed up the fetch time.

Twitter only allow people to fetch recent 7 days tweets, but we only keep the price data from 2017, the alternative way to achieve this is to use premium search. The free subscription barely allow each Application to download 5,000 tweet per month. And each request can only return 100 tweets. So far, this could be a fatal issue that impact the amount of training data and may cause the hourly analysis become unpractical.

4.2.2 Data Preprocessing

The twitter API return a bunch of information related to the tweets, it includes text of tweets, the created time, the username, how many likes, how many retweet, how many comment, etc. To get better result with limited data set, the alternated way to utilize those tweet is to set weight for them, we could give a higher weight for highly liked, with many comments, with many retweet, those tweet will be assign a significant weight. After that, we could continue to do sentiment analysis.

4.2.3 Data Modeling

Using only one model/library to analyze the mood of particular tweet is weak. Among internet, there are several open source sentiment analysis libraries. We could try to using three or four of them to generate both emotional polarity (positive or negative) of sentences or multi-dimensional structure of human mood (detailed mood). Then using major vote to get final result.

So the training data should only contain two columns, one is binary set (positive and negative), another is a label of mood.

4.3 Google Trends

4.3.1 Data Collecting

There is a unofficial library called pytrends to get data from google trend. By using it, we can fetch specific period of trends. For example, we could get trend data for APPLE (AAPL) within Jan 1st 2018 through Jan 2ed 2018. With return dataset, we could easily find maximum value , which means at that time, AAPL reach the peak search trend. To demonstrate it, plot the daily trend:

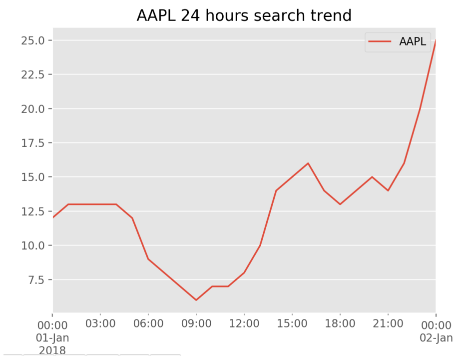

Figure 2: Daily Trend Sample

As you can see the figure above, That might be weird that the highest value is at midnight, the reason for that is because the Google trends use UTC time zone, which has three hours time difference. So the figure with correction should be:

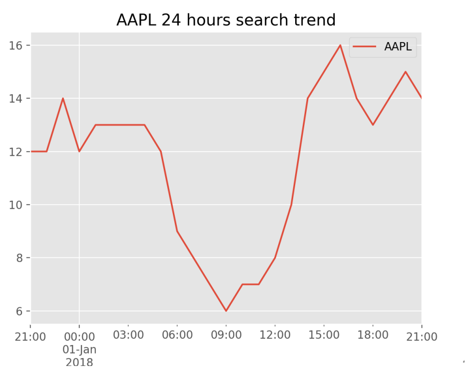

Figure 3: Daily Trend Sample correction

The reason why the values for the trends are so low is that January 1 is holiday, so when we shrink the scale to week size, we can see this trends:

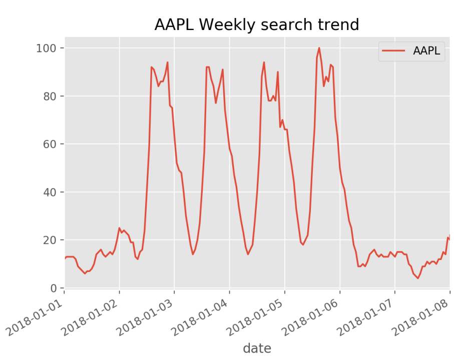

Figure 4: Weekly Trend Sample

At this time, the trends give us a brief spike on four week days, which and a hollow on weekend.

4.3.2 Data Preprocessing

Now, with a daily graph of trends, we could easily find some local maxima and locate the time. We can record the value of those peak, after that, to make it as a feature to train.

5 Data Processing

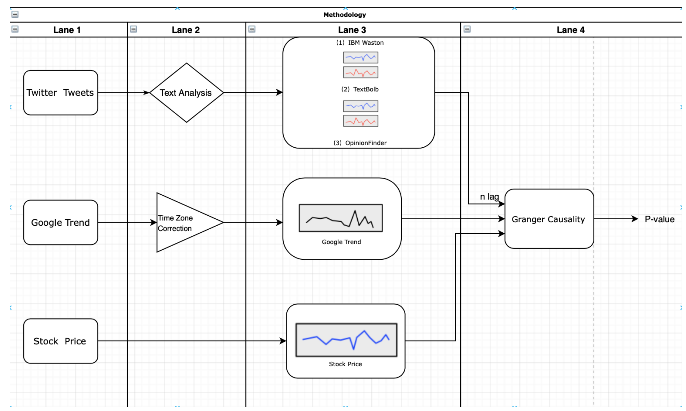

Figure 5: Data Processing

5.1 Sentiment Analysis

We are using three model or API to classify the mood of twitter, they are: IBM Watson,

5.1.1 IBM Watson

IBM Watson offer convenience API to analyze the sentiment and mood of web page or short sentence, which is very suitable for our project. To understand how IBM Watson Natural Language Understanding works:

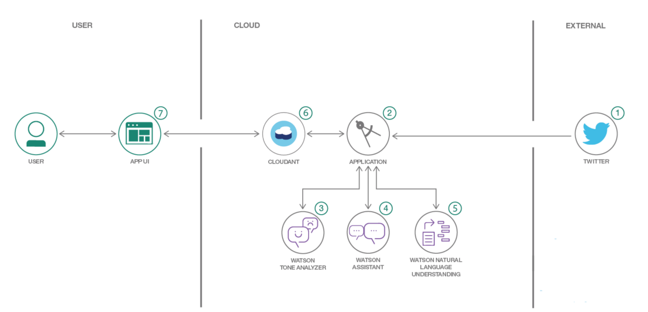

Figure 6: How it works

By deploying a real-time monitor tool, we could monitor the emotion over twitter in real-time. To make the whole processing, we used several service at IBM Cloud:

- Watson Assistant:

  Watson Assistant is a robust platform that allows developers and non-technical users to collaborate on building conversational AI solution.

- Watson Tone Analyzer:

  Uses linguistic analysis to detect communication tones in written text.

- Watson Natural Language Understanding

  Natural language processing for advanced text analysis.

- IBM Cloudant:

  A managed NoSQL database service that moves application data closer to all the places it needs to be — for uninterrupted data access, offline or on.

To generate Web page of interface, Node.js was been used here and deployed at local since we don't need to open it to public.

The home page of monitor is shown above.
Cause the main goal of this project is not visualization, so we use some existing library to quickly build this page and visualization.

Feature generating

With two of services IBM offer us, we could generate both binary sentiment and emotion category. The output of features are supposed to be


```
{     ”_id”:”04bebec790cc71d89d7ef06fa6a0e07b”,
          ”_rev”:”1−6
    e3d43a60ba9229dc8976f13299e11af”,
    ”text”:”@JetBlue Do I need to come to the
    Emirates counter first or JetBlue?”,
    ”post_date”:”2017−06−08T22:37:52.844Z”,
                  ”post_by”:”system”,
                  ”source”:”system”,
                  ”tweet_id”:13,
    ”enrichments”:{
        ”nlu”:{
            ”usage”:{”text_units”:1,
                     ”text_characters”:68,
                     ”features”:4},
            ”sentiment”:{
                ”document”:{
                    ”score”:0,
                    ”label”:”neutral”}
            },
    ”language”:”en”,
    ”keywords”:[{”text”:”Emirates”,
                ”relevance”:0.672861,
                ”count”:1},
                {”text”:”JetBlue”,
                ”relevance”:0.644458,”count”
    :1}],
    ”entities”:[{”type”:”Company”,”text”:”
    JetBlue”,
                ”relevance”:0.33,”count”:1},
                {”type”:”TwitterHandle”,
                 ”text”:”@JetBlue”,
                 ”relevance”:0.33,
                 ”count”:1}],
    ”emotion”:{”document”:
               {”emotion”:
                   {”sadness”:0.414856,
                   ”joy”:0.18409,
                   ”fear”:0.049613,
                   ”disgust”:0.056152,
                   ”anger”:0.136122}}}},
    ”tone”:{”document_tone”:
               {”tones”:
                   [{”score”:0.716301,
                     ”tone_id”:”tentative”,
                     ”tone_name”:”Tentative”
    }]}},
                     ”intents”:[]}}
```


In this document, we store the some redundant feature for future use, and also record the emotion and tone. For emotion, sentiment score under nlu(stand for natural language understanding) give a confident value for sentiment. we also get 5 categories for specific emotion which same as (GPOMS) Google Profile of Mood States: sadness, joy, fear, disgust, anger. with specific emotion, we could do deeper analysis. For tone, it will return a classification of tone and a confident score for that, we could count it as a independent feature for each tweet.

5.1.2 TextBlob

TextBlob is the python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

TextBlob offer more than two analyzer, classifier, for sentiment analysis, Naive Bayes gave a much better result when taking sentiment analysis, so we choose Naive Bayes as the classifier.
This is where naive Bayes can help. Naive Bayes extends Bayes' theorem to handle this case by assuming that each data point is independent. The formula looks like this:

$$P(y|x_1, ...x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(y|x_1, ...x_n)}$$

This is saying "the probability that classification y is correct given the features x_1, x_2 , and so on equals the probability of y times the product of each x feature given y, divided by the probability of the x features". To find the "right" classification, we just find out which classification:$P(y|x_1, ...x_n)$ has the highest probability with the formula.

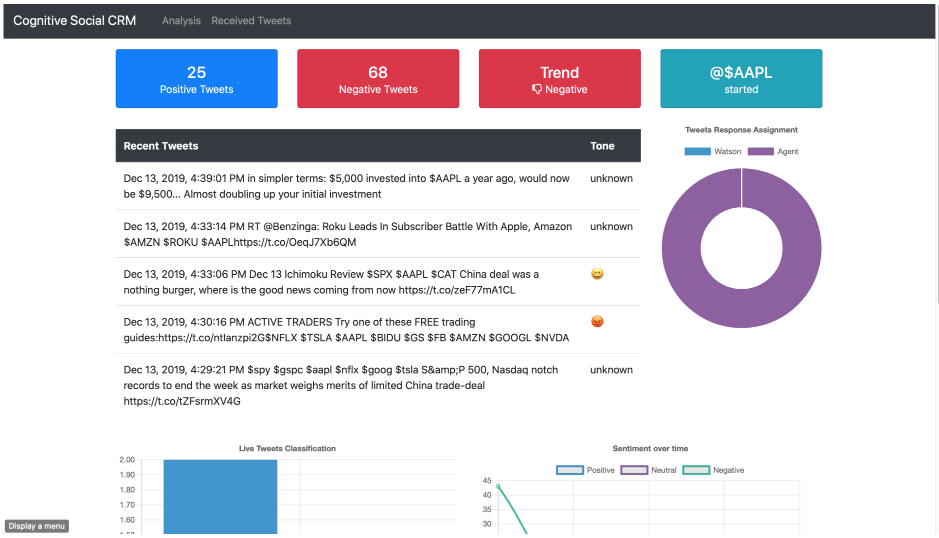

Figure 7: Monitor of $AAPL at Twitter

Feature generating

The textblob give simple three attributes as results.

$$classification = pos, p_{pos} = 0.799620, p_{neg} = 0.200379$$

So this could be additional features for tweets.

5.1.3 OpinionFinder

OpinionFinder (OF) is a publicly available software package for sentiment analysis that can be applied to determine sentence-level subjectivity, i.e. to identify the emotional polarity (positive or negative) of sentences. It has been successfully used to analyze the emotional content of large collections of tweets by using the OF lexicon to determine the ratio of positive versus negative tweets on a given day.

OpinionFinder can return so many feature for sentence. Within here, we only consider result at exp_polarity.txt. It contains mood word in the document and the what kind of mood it is (polarity). In the end, the final feature generated by OpinionFinder is a fraction between 0 and 1. Which is 0 represents negative, 1 represents positive.

Feature Generating

Same as TextBlob, OpinionFinder returns a polarity score, we could set two threshold to classify the output. 0.4 and 0.6 were chosen to be threshold, and all score below 0.4 will denote negative and between 0.4 and 0.6 is neutral, otherwise is positive.

5.1.4 Comparing with price

To get appropriate comparing, the first thing need to be done is set a common time series. For price, we use close price as value in every time scale. Which can be denoted by formula: X_t^j Where $0 \leq t \leq n$ denote time point and j denote different feature.

To get a more reasonable input data, using a slide window to normalize z-scores on the basis of a local mean and standard deviation of k hours before and after the particular time. For example, the z-score of time series X_k, denoted $\mathbb{Z}_{X_k}$, is defined as:

$$\mathbb{Z}_{X_k} = \frac{X_t - \overline{x}(X_{t\pm k})}{\sigma(X_{t\pm k})}$$

Where where $\overline{x}(X_{t\pm k})$ and $\sigma(X_{t\pm k})$ represent the mean and standard deviation of the time series within the period $[t-k, t+k]$. This normalization causes all time series to fluctuate around a zero mean and be expressed on a scale of 1 standard deviation.

5.2 Google Trend

5.2.1 Comparing with price

To compress Trend data and price data, we could simply set them as same time series, notice we could consider the market has recess but the trend does not have, so we need to complement data for those gap, we use Pandas built-in function called *interpolate()* which can automatic complement the gap using linear value between two ends. So we now could output a quite basic figure

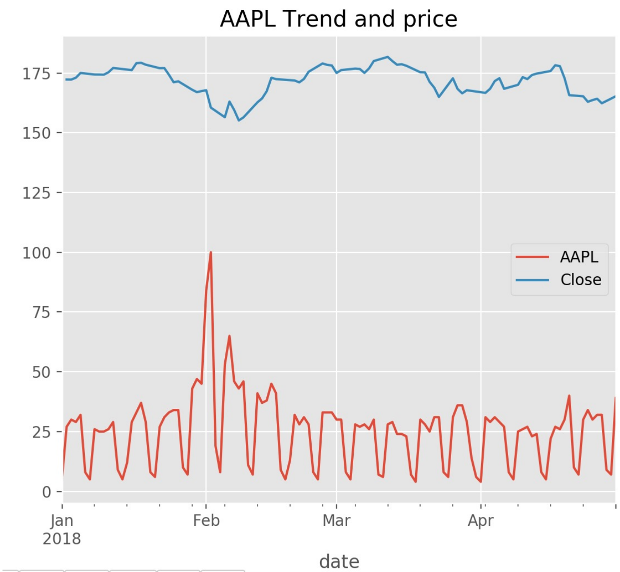

Figure 8: Daily comparison

Obviously, this is not enough for find relevant between them. Because it have a highly relevant with daily time, which means the open and close of market impacted it tremendously. Two methods to deal with this issue, if the time frame was reducing to daily, or

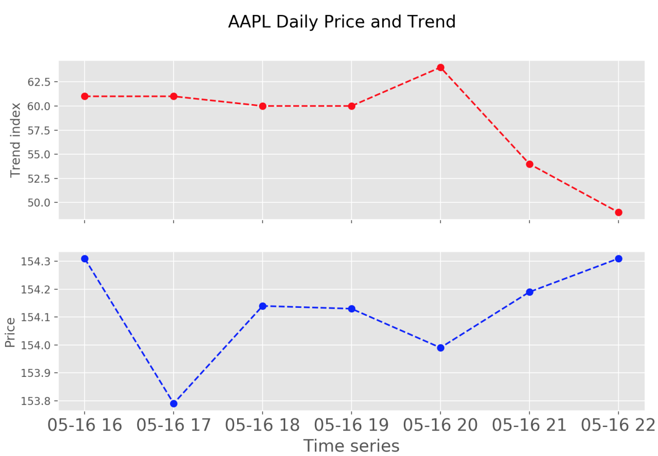

Figure 9: Trends and Price

5.3 Granger Causality

The impact from all of resource can lead two directions, which make it impossible to calculate the correlation between them and price. At here, we introduce the Granger Causality to deal with this issue. Granger causality analysis rests on the assumption that if a variable X causes Y then changes in X will systematically occur before changes in Y . We will thus find that the lagged values of X will exhibit a statistically significant correlation with Y . Correlation however does not prove causation.

Let's understand how a multivariate time series is formulated. Below are the simple K-equations of multivariate time series where each equation is a lag of the other series. X is the exogenous series here. The objective is to see if the series is affected by its own past and also the past of the other series.

$$Y_{1,t} = f_1(Y_{1,t-1}..., Y_{k,t-1}...Y_{1,t-p}, ...Y_{k,t-p}....X_{t-1}, X_{t-2}....)$$

$$Y_{k,t} = f_k(Y_{1,t-1}..., Y_{k,t-1}...Y_{1,t-p}, ...Y_{k,t-p}....X_{t-1}, X_{t-2}....)$$

This kind of series allow us to model the dynamics of the series itself and also the interdependence of other series. We will explore this inter-dependence through Granger's Causality Analysis.

The result of Granger's Causality test with 1,2,3 lag shown in table 1. In table 1, we could find that when lag equal to 3, the p value of trends and stock price can be proven that have a relative connection. In this case, the stock price can be affected by past 3 time length. After

Table 1:
Granger's Causality test

	number of lags (no zero) 1	number of lags (no zero) 2	number of lags (no zero) 3
ssr based F test	F=0.0545 p=0.8154 denom=988 num=1	F=0.2671 p=0.7657 denom=985 num=2	F=2.2071 p=0.0857 denom=982 num=3
ssr based chi2 test	chi2=0.0547 p=0.8151 df=1	chi2=0.5369 p=0.7646 df=2	chi2=6.6686 p=0.0832 df=3
likelihood ratio test	chi2=0.0547 p=0.8151 df=1	chi2=0.5367 p=0.7646 df=2	chi2=6.6462 p=0.0841 df=3
parameter F test	F=0.0545 p=0.8154 denom=988 num=1	F=0.2671 p=0.7657 denom=985 num=2	F=2.2071 p=0.0857 denom=982 num=3

that, we move to analyze how sentiment among tweets impact the stock price. We first use major vote to classify the polarity of emotion with TextBolb, Opinion Finder and IBM Waston. At this time, we only take account of polarity of emotion. The result can be show on table 2

Table 2:
Emotion Granger's Causality test

lag(days)	OF	IBM	TB	Major Vote
1	0.0851	0.1203	0.1331	0.9847
2	0.0675	0.0524	0.8512	0.0609
3	0.0964	0.1091	0.8830	0.1013

In the table above(shown in Table 1), we could prove that the tweets sentiment can reflect the price changing partially. In the table, we could see that with lag=2, the result shows the best correlation between them, for all of three tool to generate polarity sentiment, the method of major vote cannot improve the p-value significantly, but it can reduce the variance among these three method.

6 Discussion

In this paper, we investigate the tweets mood and google trends as factors that can affect the stock price. Our results show that both emotion over tweets and google trends can be indicators for stock market. But the results still maintain some defect, most of them due to the amount of resource and more and more sophisticated model. The twitter limited the number of retrieve data from it, which reduce the accuracy and constrain the time scale into daily rather intraday analysis. We could filter most of spam data and set up the appropriate time series. But some stocks still trapped with the issue about how to filter the common tweets. Like the stock Macy's, whose stock code is M. Even through, we can use advance searching tool which can restrict the keyword to $M. Nevertheless, we will lose part of tweets which only contain M. So, more elaborate filter for tweets should be implement in the future.

The monitor of tweets is a prototype, which can supervise the real-time tweets and emotion. But also subject to twitter's API, the streaming cannot last for a long time with a free subscription. To refine and make it modularization, the most important part is serialize the feature from real world, in another word, all article, tweets, blogs, even posting on Reddit, all of those places are resource of data, to monitor them at same time also means monitor the emotion within market. We cannot build a wonderful model up to predict price and trends. But knowing the mood from market can make trading more confident.

The future researching should focus on finding an easier way to normalize data and visualize it. In addition, some of sentiment analysis tool offer more specific classifier, to observe how different mood can impact market also can make this topic more detailed. Another thing need to be figure out is the causative mechanisms that may connect public mood states with stock market in this manner.

References

[1] IBM Watson Document
`https://www.ibm.com/watson/developer`

[2] Twitter API Document
`https://developer.twitter.com/en.html`

[3] Johan Bollen ,Huina Mao ,Xiao-Jun Zeng. *Twitter mood predicts the stock market.* rXiv:1010.3003v1 [cs.CE] 14 Oct 2010

[4] Esther Hsu, Sam Shiu, Dan Torczynski *Predicting Dow Jones Movement with Twitter*

[5] Wikipedia Granger causality
`https://en.wikipedia.org/wiki/Granger_causality`

[6] Preis, T., Moat, H. & Stanley, H. *Quantifying Trading Behavior in Financial Markets Using Google Trends. Sci Rep 3, 1684 (2013) doi:10.1038/srep01684*

[7] Dartanyon Shivers *Exploring the relationship between Google Trends data and stock price data.*

[8] RAN AROUSSI *Reliably download historical market data from Yahoo! Finance with Python*

[9] OpinionFinder, MPQA Resources, University of Pittsburgh
`http://mpqa.cs.pitt.edu`

[10] Werner Vanzyl *Monitor Twitter feeds to better understand customer sentiment using Watson Assistant, Tone Analyzer, and Natural Language Understanding*