Stock Market Trends Prediction Using StockTwits Tweets Analysis

Report For NLP (SE316)

Delhi Technological University

Submitted By:

Ayush Gupta
2K15/CO/040

Jatin Thareja
2K15/CO/069

Rishabh Kumar
2K15/CO/104

Submitted To:
Ms Minni Jain

Table Of Contents

Summary

In this project, we aim to predict the bullish (increasing) or bearish (decreasing) nature of stocks of three companies namely Apple, Amazon and Google by performing sentiment analysis on the stocktwits tweets for their stocks and gather the prevailing trend for them and using the result, along with other factors like previous actual sentiment, change in tweet volume and cash flow to predict their bullish or bearish nature.

We applied various approaches of sentiment analysis ranging from the lexicon-based approach of SentiWordNet to supervised learning based approach of RNN, DNN and Naive Bayes classifiers to a labelled corpus of 1.5 lakh tweets, collected from stocktwits website. Ultimately we combined the lexicon-based approach with supervised learning approach for best results. We achieved an accuracy of 73% for sentiment analysis on our test data.

However, the actual prediction of share market movement is so uncertain and comprises of a large number of variables that a 50% accuracy is considered satisfactory while an accuracy greater than 60% is considered significant in this domain. Using our approach we were able to achieve an accuracy of 57% in predicting the actual trend of the market.

Introduction

Stock Market is one of the most dynamic and volatile places which is highly dominated by general sentiment of traders. It is characterized by high uncertainty factor and fast-paced changes in trends. Building a system that can predict the movement of the stock market has posed a major challenge for researchers since such a system has to deal with a high noise to signal ratio. Moreover, the movement of the stock market is mainly determined by the sentiment of traders which is not easily captured. These traders are influenced by a large number of factors like monetary reports, news, general opinion about the company as well as the opinion of the fellow traders. Targeting the events that do have an effect on the prices of stocks and predicting the exact effect they cause has largely remained unsolved till date. Many researchers have focused on capturing this input through means such as news articles, company's press releases, budget reports etc. In this project, we focused on the prevailing sentiment of stocks of a company through StockTwits tweets.

StockTwits is a website, dedicated towards discussions and comments about stocks. It attracts and allows a large number of people which includes a fair number of professionals to share their views and sentiment about particular stocks. Hence, it proves to be a great tool for analysing the mood of the traders. We aimed to gather the prevailing sentiment of the stock of a company by analysing tweets about them and using the result, along with other parameters like the previous actual sentiment, change in tweet volume, and cash flow to predict the bullish (increasing) or bearish (decreasing) nature of the stock. Since stock movements are so uncertain and depend on an uncountable number of variables, a 50% accuracy is considered satisfactory while an accuracy greater than 60% is considered significant in this domain.

Here, we have focused on stocks of three major companies Apple, Amazon and Google for which plenty of data is available on stocktwits website. The main beneficiaries of this project are the traders who wish to know the general sentiment of specific stocks to aid their own instincts.

Problem Definition

To predict the bullish or bearish nature of stocks of three companies namely Apple, Amazon and Google by performing sentiment analysis on stocktwits tweets of their stocks and using the result along with other parameters to predict the trend of the market.

Background Study

The problem of predicting stock market movement through social sentiment analysis has been studied for a long time.
Yet extensive research has not been published on this topic as:

1. Stock markets are too unpredictable
2. Social analysis alone does not influence the markets, other factors like economics, politics, natural calamities also affect the prices
3. The accuracy currently achieved is not at par with those of the other topics.

Existing works looking into this topic have found correlations between bullish sentiment on Twitter and short-term price anomalies of stocks [8], as well as message volume peaks and abnormal price action [9]. However, a critical issue commonly found in previous research is statistical inaccuracy and over-fitting, including the selection of equities for which the results are demonstrably favourable.

Also, research has been going on topics which are similar though not exactly identical to this including:

1. Predicting Stock Movement through Executive Tweets [6]
2. Stock Market Trends Prediction after Earning Release [7]

These research works are similar in the fact that they make use of natural language processing to predict stock movements, though they focus on a different kind of available data, which is more peculiar and directly related to stock markets, but its lesser in volume.
These techniques though more accurate are not applicable to a large number of stocks due to the scarcity of data.

Proposed Methodology

Our approach has the following stages:

1) Data Collection:

Stocktwits has a facility where the user can tag his/her message with a bullish or bearish sentiment. This tagging helped us to gather a labelled corpus of 1.5 lakhs tweets comprising of 66% bullish messages and rest bearish. This data will help us to train our classifiers for supervised learning.

Another set of stock specific tweets were gathered from stocktwits API comprising an unlabelled set of 3 Lakh messages each ranging over an average period of 2 years for the companies Apple, Amazon and Google. So in total, we had 9 Lakh tweets for stock trend prediction and 1.5 Lakh tweets for sentiment analysis.

We also gathered each day stock's opening and closing price along with cash flow for a period of 3 years using Intrinio API.

2) Data Preprocessing:

Since all the data we collected were from a source meant specifically for share market related discussion, it had very little noise. However, the messages themselves needed preprocessing since they contained links, tickers, tags, digits, special characters, HTML entities etc. which would play no role in our computations. Hence we stripped the data of these attributes and reduced it to bare textual message in lower case.

However, we did not remove stop words from the data since many researches have shown that they affect the sentiment of the text. We also left emoticons as it is since they can provide a good idea about the sentiment.

3) Sentiment Analysis Phase:

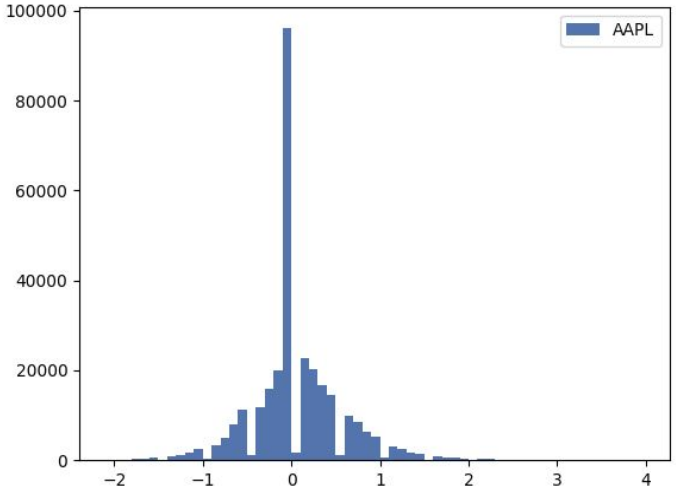
We tried a number of approaches for sentiment analysis which used both lexicon based namely SentiWordNet scoring and supervised learning based namely RNN, DNN, Naive Bayes classifiers. Finally, according to our results, we used a combination of Naive Bayes along with SentiWordNet scoring for best results.

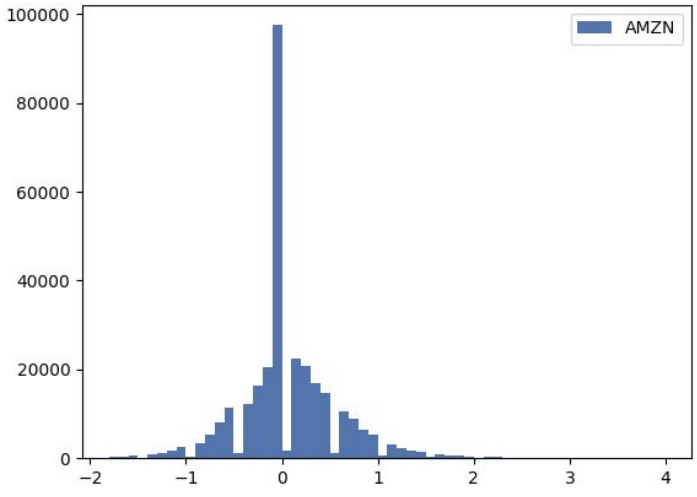
a) SentiWordNet approach:

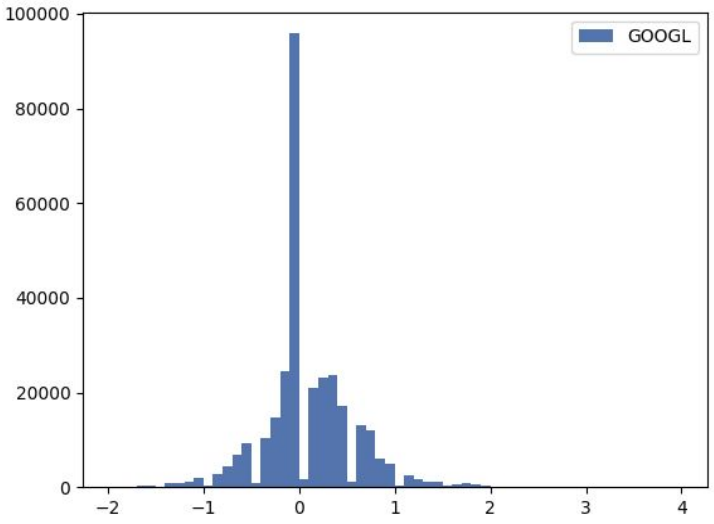
SentiWordNet is a sentiment analysis lexicon built over WordNet. It assigns each of the synsets in WordNet with three scores: the objectivity score, the positivity score and the negativity score, sum of which comes out to be 1. Hence a word in a synset can be objective, positive and negative at the same time depending on the weights. We used NLTK which comes with SentiWordNet corpus to score our data of 9 Lakh tweets. It involved the following steps:

i) Tokenization of the messages.
ii) POS tagging using TreeBank corpus.
iii) Picking nouns, verbs, adjectives and adverbs from the text.
iv) Lemmatizing the above words.
v) Determining the senses of words and using lesk algorithm for word sense disambiguation in case of multiple senses.
vi) Getting the SentiWordNet score for the sense.

All the above tasks were carried out using NLTK. Also, we gathered a list of negation words from WordStat financial dictionary, and whenever we encountered a negation word while scoring a sentence we reversed positive and negative scores for all subsequent words in the sentence.







Sentiwordnet scores for stocks of Apple, Amazon and Google

b) RNN / DNN classifier-based approach:

DNN is a class of convolution neural networks. They take a fixed size input and generate fixed-size outputs. It is a type of feed-forward artificial neural network - are variations of multilayer perceptrons which are designed to use minimal amounts of preprocessing. They use connectivity pattern between its neurons is inspired by the organization of the animal visual cortex, whose individual neurons are arranged in such a way that they respond to overlapping regions tiling the visual field. They are ideal for images and videos processing. On the other hand, RNN can handle arbitrary input/output lengths. Unlike feedforward neural networks they can use their internal memory to process arbitrary sequences of inputs. Recurrent neural networks use time-series information. I.e. what is spoken last will impact what will be spoken next. RNNs are ideal for text and speech analysis.

To run sentiment analysis on the data we first need to convert words to their respective dense vector representation.

Word Embedding

Word embedding is the collective name for a set of language modelling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of

real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with much lower dimension. Word embeddings provide a dense representation of words and their relative meanings.

They are an improvement over sparse representations used in the simpler bag of word model representations. Word embeddings can be learned from text data and reused among projects. They can also be learned as part of fitting a neural network on text data. Two popular examples of methods of learning word embeddings from text include:

- Word2Vec.
- GloVe.

Word2Vec

Word2vec is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. While Word2vec is not a deep neural network, it turns text into a numerical form that deep nets can understand. The output of the Word2vec neural net is a vocabulary in which each item has a vector attached to it, which can be fed into a deep-learning net or simply queried to detect relationships between words.

Word2Vec trains words against other words that neighbour them in the input corpus.
It does so in one of two ways:

1. Continuous Bag of Words(CBOW): It uses context to predict a target word
2. Skip-Gram Model: It uses neighbouring words to predict a target context

GloVe Embedding

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. It is common in the field of Natural Language Processing to learn, save, and make freely available word embeddings.
The smallest package of embeddings is 822Mb, called “*glove.6B.zip*“. It was trained on a dataset of one billion tokens (words) with a vocabulary of 400 thousand words. There are a few different embedding vector sizes, including 50, 100, 200 and 300 dimensions.

Both RNN and DNN were implemented using TensorFlow and the word embeddings were learned along with the training part of the respective models and gave an accuracy of 70% on our labelled training data.

c) Multinomial Naive Bayes approach:

Naive Bayes classification has been known to give good results for sentiment analysis. It is fast to train and implement. We created a vocabulary of more than 65000 words from our labelled data and use the frequency of appearance of each word as the feature vector for our classification. We calculated term frequency of each word in the tweet. However, since the size of tweets is restricted to 140 characters there was no need to calculate inverse document frequency. We used both unigrams and bigrams of words for our classification. Naive Bayes classification was implemented using scikit-learn and gave us an accuracy of 72% on the test data.

Finally, for best results, we combined the lexicon based positive and negative scoring of a word as a feature of our sample to get an accuracy of 73.15%

4) Prediction Phase:

The final stage of our project was to use the result of sentiment analysis to predict bullish or bearish nature of stocks. We used a Multi-Layer Perceptron of two hidden layers each of size 2, using ‘lbfgs’ optimizer which is an optimizer in the family of quasi-Newton methods. The input for the neural network consisted of the total number of bullish and bearish messages each data for a particular stock, the actual previous day trend, change in volume of messages about the stock, and cash flow in the market for a particular stock. The output was either a bullish or a bearish label for next day. Before classifying we scaled our data to have a unit standard deviation. The classifier was implemented using scikit-learn and gave an accuracy of 57% on our test data.

Implementation Example

Consider a list of tweets for the stock of Apple.

['$APPL is bound to increase!!!!!!',
'@rK buy buy :) buy buy :) $APPL',
'I am bearish for *** $APPL ***',
'Unpredictable, for $APPL see this https://www.apple.com']

1) Preprocessing phase changed the set of messages to following

[' is bound to increase',
'buy buy :) buy buy :) ',
'I am bearish for ',
'Unpredictable, for see this ']

2) Each sentence is now given a positivity and a negativity score. (The objectivity score is equal to 1 - (positive + negative) scores), which is denoted by a tuple (positive score, negative score)

[(0.125, 0.125), (0.875, 0.0), (0.0, 0.5), (0.0, 0.625)]

3) The Sentiment of each sentence is calculated using a combination of lexicon and supervised learning approach given by

['Bearish', 'Bullish', 'Bearish', 'Bullish']

4) Now the above results, along with other features are fed to MLP classifier. Let us say that the previous day actual sentiment for Apple was Bullish, the previous day tweets volume was 7, and the cash flow for today is 7000000. Thus the feature vector will become

[Cash Flow = 7000000,
Previous actual sentiment = 1,
Calculated bearish sentiment = 2,
Calculated bullish sentiment = 2,
Change in tweet volume = -3]

The above feature vector is scaled for unit standard deviation and fed to the MLP classifier to get the final predicted label as 'Bullish'.

Results

Our approach to sentiment analysis using a combination of lexicon-based and supervised learning using Naive Bayes gave an accuracy of 73.15%. The results were as follows:

	Precision	Recall	F1-score	Support
Bearish	0.53	0.78	0.63	1559
Bullish	0.89	0.71	0.79	3738
Avg/Total	0.78	0.73	0.74	5297

Results for sentiment analysis

The results for predicting the actual trend of the market, using Multi-Layer Perceptron model gave an accuracy of 57%. The results were as follows:

	Precision	Recall	F1-score	Support
Bearish	0.59	0.12	0.20	82
Bullish	0.59	0.94	0.72	109
Avg/Total	0.59	0.59	0.50	191

Results for prediction of stock market trends

Conclusion And Shortcomings

An accuracy of 57% for stock trend prediction is in general considered satisfactory. Our model has a high recall for Bullish trends whereas quite low recall for Bearish which is a direct consequence of less data available for bearish sentiment as well as the bearish movement of prices. The sentiment analysis model also suffers from this cause and thus shows a high precision rate for bullish classification as compared to bearish. Another improvement that can be made during prediction is that our model gives equal weights to the tweets of all users, however, in the real world, some users are more influential over others while some are good in the prediction of the stock trend than others. Messages from such users can be detected and given a preference over others. Also, the lexicon we followed as one of the features of our classification is not optimized for stock related messages. A custom scoring lexicon might give better results than SentiWordNet.

References

1. Predicting Stock Price Movement Using Social Media Analysis, Derek Tsui
2. SENTIWORD NET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining, Stefano Bacci anella, Andrea Esuli, and Fabrizio Sebastiani
3. Opinion Mining Using SentiWordNet, Julia Kreutzer & Neele Witte
4. Contextual Semantics for Sentiment Analysis of Twitter, Hassan Saif, Yulan He, Miriam Fernandez, Harith Alani
5. On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter, Hassan Saif, Miriam Fernandez, Yulan He, Harith Alani
6. Predicting Stock Movement through Executive Tweets, Michael Jermann
7. Stock Market Trends Prediction after Earning Release, Ran An, Chen Qian, Wenjie Zheng
8. Huina Mao, Scott Counts and Johan Bollen. Quantifying the effects of online bullishness on international financial markets. European Central Bank, 2015
9. Gabriele Ranco, Darko Aleksovski, Guido Caldarelli, Miha Grcar, and Igor Mozeti ˇ c.ˇ The Effects of Twitter Sentiment on Stock Price Returns. PLoS ONE, 2015.