Sentiment analysis of tweets and their effects on the Stock Market

Hrishikesh Mahajan*
Department of Computer Science
Binghamton University - State University of New York
hmahaja1@binghamton.edu

Jainil Parikh†
Department of Computer Science
Binghamton University - State University of New York
jparikh1@binghamton.edu

ACM Reference Format:
Hrishikesh Mahajan and Jainil Parikh. 2020. Sentiment analysis of tweets and their effects on the Stock Market. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

An individual can broadcast a brief statement in real time to some or all members of the sender's social network through Twitter. Twitter, which has large audience potential, currently attracts an estimated average of 271 million users every month. Twitter effect has been shown to be particularly relevant to experiential media products (e.g., movies, music, and electronic games); these are generally the products for which 'instant' success is essential. Twitter-based models can then be built to aggregate the opinions of the collective population. They can be used to predict future trends while gaining useful insights into individual behavior. Social networks offering micro-blogging services enable the rapid spread of user generated content (UGC) from a handful of individuals to millions of people around the world in the form of short text, images or videos. Micro-blogging platforms have grown so exponentially, that they are now perceived as indispensable sources of information and are fast gaining popularity amongst users, organizations and researchers in various disciplines. Twitter is updated hundreds of millions of times a day with content varying from individual daily life updates to worldwide news and events. Twitter allows users to create personal profiles that others may subscribe to or 'follow', publish status updates known as 'tweets' limited to 140 characters and to communicate with others through 'replies'. 'Retweeting' is a common practice whereby a user can choose to forward a tweet they find interesting to their followers whilst crediting the original author, allowing popular posts to travel well beyond the network of the original creator. It is therefore perceived that highly retweeted posts reflect the views of the global Twitter community Twitter also encourages the use of hashtags which allow tweets to be collated in a thread that can be used for following specific events and topics. Twitter tracks the most mentioned phrases and hashtags and posts them under a list of 'trending topics', which is updated regularly, and allows users to keep track of what is most popular at any given time. The purpose of this project is to create a pipeline for streaming the twitter data made available through the Twitter's official API. In addition to this, we also plan to scrape the stock market data of the trending 30 companies. We plan to perform sentiment analysis on the tweets to identify the pattern revealing how real-time events lead to an effect on stock market.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 SYSTEM ARCHITECTURE

The Figure 1 describes our System Architecture for our Project. We plan to collect the tweet data from the Twitter's official API. For every tweet that comes in, we are storing the tweet-ID and the timestamp of the tweet. We then pass the tweet to our Twitter Filter which extracts relevant tweets to our project using a list of keywords to match in the tweet data. If present, the function stores them into the Database. Else, discards the data. For the Stock market data, we hit the website every 10 minutes and the website sends us a data in HTML format. We pass this data to a Data Filter which uses the BeautifulSoup4 library to removes the HTML tags and gives us data in text format. This data is fed into a Relevant Data Extractor function which gives us the data for the trending 30 companies.

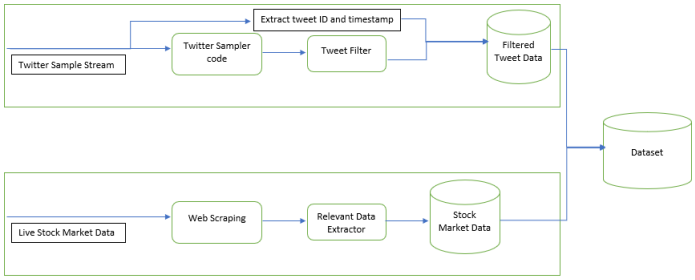

Figure 1: Pipeline Architecture

3 DATASET

Through the Twitter's official API, we were able to isolate and store relevant tweet fields for our project like hashtags, text, retweet text, retweet count, quote count, and hashtag count. Simultaneously, we are also collecting the stock market data for the trending 30 companies by scraping data of a website: https://www.investing.com/equities/trending-stocks. We have set up a web crawler which hits this website every 10 minutes and feeds the data to out Relevant Data Extractor function which extracts the following attributes:

Name, Last, High, Low, Change, Change Percent and Time. The Figure 2 represents the plot of data obtained in KB against time. Since we periodically fetch the same amount of data, we see a straight line graph.

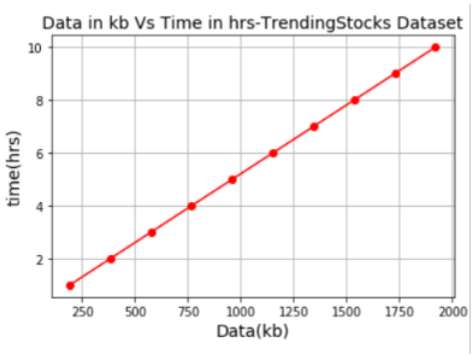

Figure 2: Stock Data Plot

4 IMPLEMENTATION DIFFICULTIES

While implementation of the pipeline, we had to update our approach because of some unforeseen problems we encountered. We tried our best to stick to the architecture we submitted in the proposal, however, the increasing complexity of code and considering the time constraints, we finalized on the design as displayed in Figure 1 This section briefly describes some of the difficulties in our implementation of this pipeline.

4.1 Design Issues

Before finalizing at this design in Figure 1 of our system, we were planning on a different approach. We tried to pull all data of every tweet and store it into a file of a fixed number of tweets (say 10,000). The issue with this design was that, we were then supposed to schedule another job which would periodically read all the JSON files and extract relevant information. In addition to this, the size of such data was huge. (About 28 MB of data in 12 minutes) With limited disk space at our disposal, we thus decided to switch to a more refined approach.

4.2 Accessing relevant fields

The twitter data is in a JSON format and it contains nested columns. In order to extract relevant features, we then planned to flatten the JSON to have individual columns and then drop non relevant columns. But this approach led to massive duplication of columns and increased the overhead. So we decided to filter the tweet fields just as they are read in the program

4.3 Storing Match Data

Our code previously focused upon hashtag fields as a match for 'relevant data'. However, many users on twitter do not necessarily post tweets with hashtags. Testing revealed that a significant amount of data was being lost. To solve this issue, we then monitored the text of the tweet along with the hashtags included in order to classify a tweet as 'relevant data'.

5 UPDATED PROJECTIONS ON DATA

Our data collection is two fold. The following subsections describe the projections on data collections for this pipeline implementation

5.1 Stock Market Data

The final testing of the stock market crawler and the pipeline was completed om October 22nd. Since we pick the data for the trending 30 companies, the size of the data does not differ significantly from its last iteration. As of now, the data has been steadily increasing with current size at 2.219 MB. The projections show that the maximum size of this dataset should not exceed over 10MB.

5.2 Twitter Data

The final testing of the twitter data pipeline was completed on November 7th. We have divided this data into 2 tables. The 'allTweets' tables stores the ID and timestamps of all the tweets from the sampled stream. This data is now at about 480MB. Our projections suggest that this data should be at least 16GB when we stop the pipeline. In addition to this, we are storing relevant tweets into the 'filteredTweets' table whose current size is about 16MB is projected to grow up to at least 2GB. Note that these numbers are an extrapolation of current data at our disposal. Since twitter is a volatile place, the number of tweets arriving at any second and their relevance to our data is subject to events in future. However keeping a tolerance of half of our current projections, we do not expect this pipeline's data collection to grow beyond 30GB.

6 CONCLUSION

In this project, we implemented a pipeline that stores data from two sources namely: Twitter and Stock Market. This pipeline implementation is the first step towards finding the interplay between sentiments portrayed on Twitter and its effect on the Stock Market. We plan to use the Stock Market data to first find for unusual spikes or drops in the prices of shares and then identify tweets posted during this time to analyse the sentiments of people which led to the deviation of market from normal trend. Thus this pipeline implementation serves an important role towards the whole analysis of the data.