

BIG DATA ANALYTICS SYMPOSIUM FALL'18

*Realtime Stock Price Prediction
Using Sentiment Analytics*

Vaibhav Gupta
Saisaketh Ramireddy



- Abstract
- Motivation
- Goodness
- Data Sources
- Design Diagram
- Results
- Obstacles
- Summary
- References

Contents



Stock price movements are heavily dependent on the popular perception of the company. The basic aim of the project is to predict whether to go long or short on a given stock based on the market sentiment. In order to infer market sentiment, we use data from social networks, namely StockTwits and Twitter. We perform sentiment analysis on this data using various techniques, including user annotations, Stanford NLP and AFINN word scores.

Abstract



- Who are the users of this analytic? Who will be the beneficiaries?

Investment Bankers and Hedge Fund Managers

- Why is this analytic important?

Many investment banks and hedge funds have dedicated R&D departments for stock prediction. In today's digital age, where social media drives the general opinion, and even affects the market performance of an organization, it is of utmost importance to gauge the sentiment toward an organization and be able to make an informed investment decision.

Motivation



We measure the correctness of our Twitter sentiment score by treating the user-annotated Bullish/Bearish sentiment (provided by StockTwits) as the baseline.

We first calculate true-positives, false-positives, true-negatives and false-negatives using hive queries and then use it to measure accuracy, recall and precision of our analytics.

Goodness



Another hypothesis of the project is that the (online) market sentiment affects the stock price.

The goodness of this assumption can be verified by finding the correlation between hourly stock price data and our derived sentiment. However, the mentioned data is not available and needs to be accumulated manually.

Besides, there are numerous examples to prove this correlation. Remember when Kylie Jenner tweeted negatively about Snapchat and its stock fell by 10%?

Goodness



Data Sources

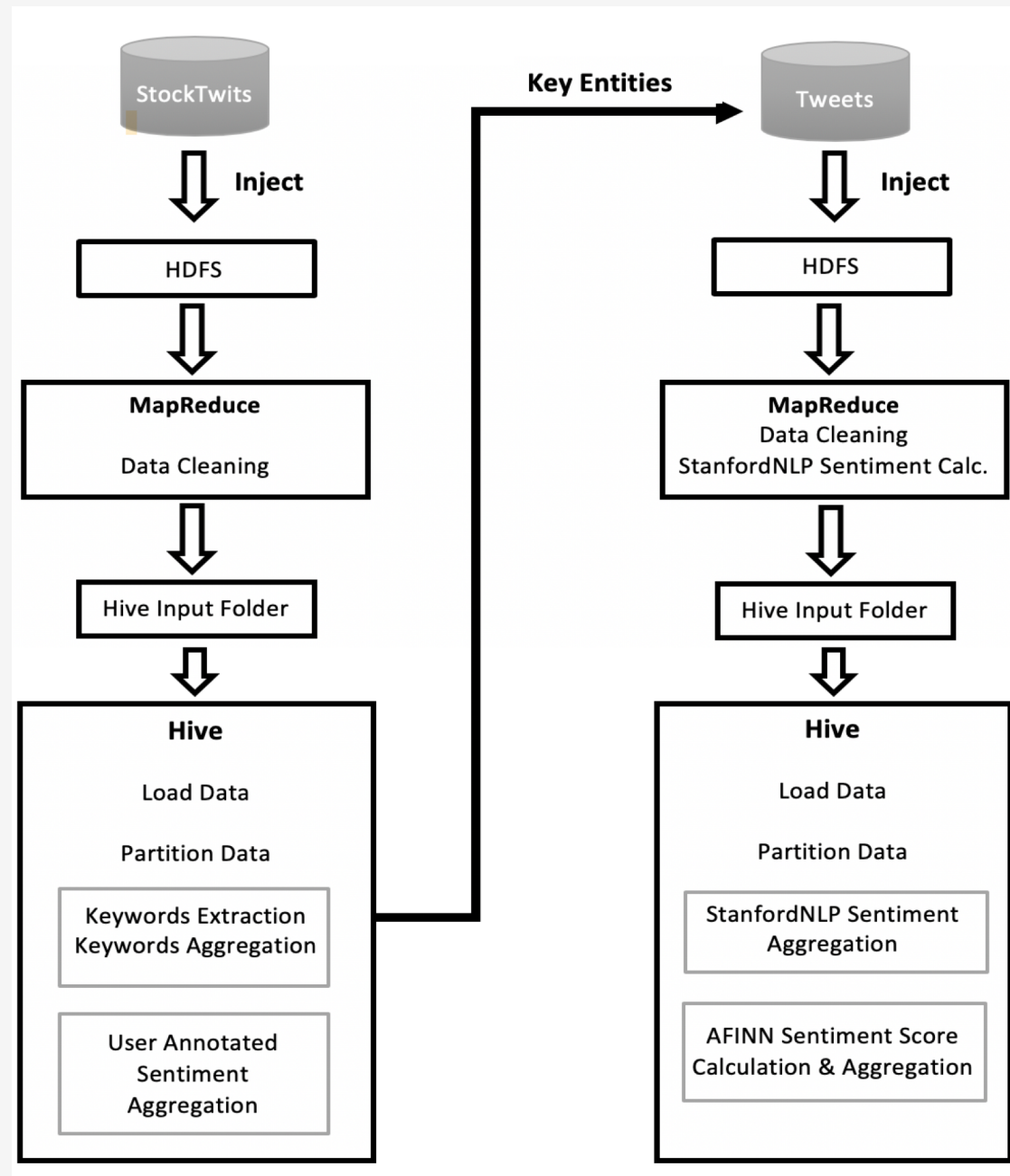
- StockTwits

We accumulate realtime data from stocktwits by making one batch calls every hour, one for each of the given companies (using their publicly listed symbol). We get around 30KB data per hour.

- Twitter

Once we derive the key name entities from the StockTwits data for each of the subject companies, we use Twitter Streaming API to fetch tweets about the mentioned entities. Fetches around 350 KB of data per hour.

Design Diagram





Results


1. According to our results, Stanford NLP underperforms when applied to twitter data. Generally it predicted either negative or neutral aggregate sentiment for most of the tweets and nearly all (date, hour) partitions.
2. If we use AFINN scores, a strong correlation is observed between Twitter's derived sentiment and StockTwits' user annotated bullish/bearish sentiment. This is in accordance to our hypothesis.

Twitter Sentiment	Stock-Twits Sentiment		
	Total = 480	Bullish	Bearish
	Positive	237	78
	Negative	41	124



Obstacles

- Using Stanford NLP jar with Hadoop.
 - Java requires third-party and user-defined classes to be on the command-line's "-classpath" option when the JVM is launched. Initially we tried to use the "-libjars" option to distribute the third-party Stanford NLP jar to all nodes, but it didn't work. Finally we ended up making a fatjar containing all the required libraries and our Mapper, Reducer and NLP code.
- Processing Highly Unstructured StockTwits & Twitter Data
 - We faced a lot of difficulties due to the random nature of social network data. Newline character in the tweets led to MapReduce treating it as a separate line input. We had to use a multicharacter delimiter "\\|" in our hive input file, rather than the usually tab character.
- Custom UDFs in Hive.



We feel, we have successfully implemented production quality big data processing pipeline, which performs the following tasks sequentially:

1. Fetching hourly data from our data sources.
2. Cleaning the data.
3. Uploading it on distributed storage.
4. Hourly partitioning.
5. Ingesting data in hive.
6. Creating backup for downstream services.

Summary



References

- <http://cs229.stanford.edu/proj2016/report/Tsui-PredictingStockPriceMovementUsingSocialMediaAnalysis-report.pdf>
- <https://cwiki.apache.org/confluence/display/Hive/MultiDelimitSerDe>
- <https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL>

Acknowledgements

- Professor Suzanne McIntosh for hand-holding us through the project.
- HPC for support on Dumbo. Wensheng rocks!