Sentiment Analysis of Streaming Data (Twitter)

Robin Singh Nanda
Department of Electronic System Engineering
Indian Institute of Science, Bangalore 560012
India
robin@dese.iisc.ernet.in

Nithin Raju D
Department of Management Studies
Indian Institute of Science, Bangalore 560012
India
nithinr@mgmt.iisc.ernet.in

***Abstract* – What's happening to a business, product or other key events in the world right now and what people think about it is an important parameter to understand. Performing scalable analytics in real-time to make sense of the ubiquitous data available is invaluable to many domains. Twitter Streaming API provides access to live tweet stream data[3] that enables to understand the top trending events and provides a sense of what is happening in the world now that can aid in key decision making for people, small and large companies, and even governments in times of natural/man-made disasters or emergency situations. We tried several approaches and finalized on the NLP Sentiment Analysis deep learning model from Stanford to analyze the sentiment in each tweet (Positive, Negative and Neutral). Apache Storm was used to process the low latency data and the output of which was stored in the Cassandra database (a NoSQL database). The visualizations demonstrate the power of real-time data and the scalability was measured by increasing the rate at which the streaming data was streamed on the cluster. Our results demonstrate the power of harnessing real-time data and highlight the potential scope for further work in this space to take more informed decisions.**

I. INTRODUCTION

6000 per second translating to 500 million per day[1]?
What are these numbers – on an average, these are the number of tweets people around the world tweet in a day.
As of December 31 2015, around 1B unique visits monthly and around 320 million monthly active users[2] were on Twitter around the globe. The mission of Twitter is to give everyone the power to create and share ideas and information instantly, without barriers[2] and this means generating huge volume of data at a high rate. This nature of Twitter data translates analytics around it as a Big Data problem.

Mining this volume and velocity of data real-time can potentially uncover valuable insights about the trending topics, people's sentiment over a particular topic/event or a product and aids in the process of decision making to companies. Twitter provides a Streaming API that gives access to public to real-time tweets but in general it is limited to 1% of the tweets (statistically sampled) [3]

Apache Storm is open source distributed real-time computation system that enables us to process unbounded streams of data at real-time. This is designed for real-time processing and we have used the same to process the Twitter streaming data.

In the report, we will cover the sentiment analysis of tweets using the NLP library from Stanford [1]. Java provides a library twitter4j to process twitter streaming data and this is used in our topology. Post processing of the twitter streaming data, the tweets are visualized on a map indicating the degree of sentiment (positive, negative and neutral)

II. APPROACH

A. *First Approach*

Sentiment Analysis by the use of emoticon:-
In this approach, we tried to decide the polarity of a word based on the emoticon present in a tweet. The streaming data was processed to extract tweets and other related information such as time zone, user details, media, geo location etc. Each tweet was split into words and then based on the emoticon, the polarity was decided.
We used the Naïve Bayes Classifier to train the model i.e. labelling the data by classifying the tweets as positive,

[1] Internet live stats | Twitter Usage Statistics
[2] Twitter Usage | Company Facts
[3] The Streaming APIs | Overview

negative or neutral and then using the trained model on the test data to predict the sentiment.

Few constraints we faced here were:

1) Filtering tweets with emoticons reduced the number of tweets further.
2) Context of the sentence was not considered.
3) Tweets with sarcasm and a positive emoticon led to training the model as a positive sentiment, but this was a misclassification.

Given the above constraints, we decided to try the next approach.

B. *Second Approach*

Sentiment Analysis – deep learning NLP model by Stanford:-

In this approach, the primary difference between the deep learning model and others is that most sentiment analysis systems look at words in isolation i.e. use the bag of words approach [2] and decide the sentiment of a sentence, whereas the deep learning model builds the sentiment based on the sentence structure.

For example: This movie was actually neither funny, nor super witty.

Because of the positive words (funny, super, witty), most sentiment analysis systems may classify this sentence as positive whereas as humans we would treat this sentence as negative understanding the sentence structure and meaning. This is exactly what the NLP model does and classifies the sentence as negative.

To capture the meaning of longer phrases and based on the semantics, determining the sentiment was possible by the Stanford Sentiment Treebank and the Recursive Neural Tensor Network. For example [4] –

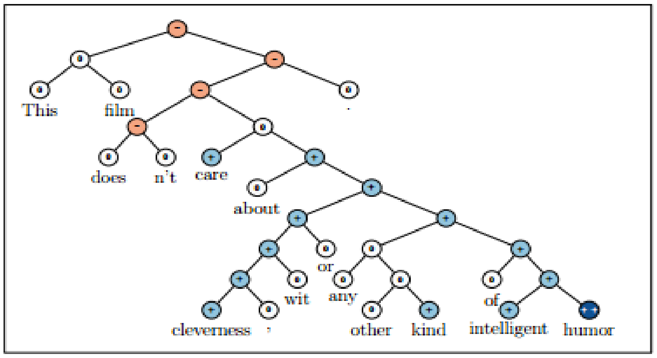
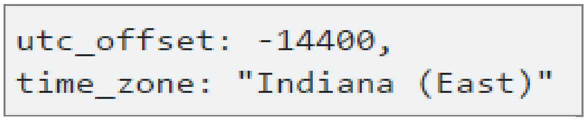


Figure 1. Example of the Recursive Neural Tensor Network predicting the sentiment at each node of the parse tree

In both the approaches, the twitter stream data was cleaned before the algorithms were applied. Following are the things we removed from the stream data: Hyperlinks, RT (retweet) tag, special symbols such as #, @, :, ; etc. Multiple simultaneous occurrences of "!?." were replaced with a single occurrence.

The Status object of twitter4j library provides the tweet and other related information such as time zone, geolocation etc. The time zone is extracted from this object using the methods getName(), tzinfoName() and utcOffset().

Please see Figure 2 below (Twitter time zone format)



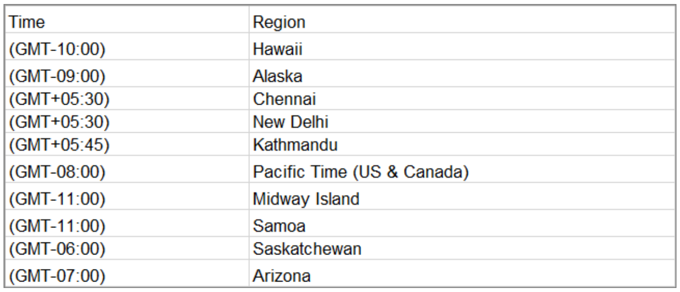
```
utc_offset: -14400,
time_zone: "Indiana (East)"
```

Figure 2. Twitter Time zone data (twitter4j)

Since the geolocation data is limited, we have used the time zone to approximately map to the country and then later used this mapping to do the visualization.

This is how the mapping was done –

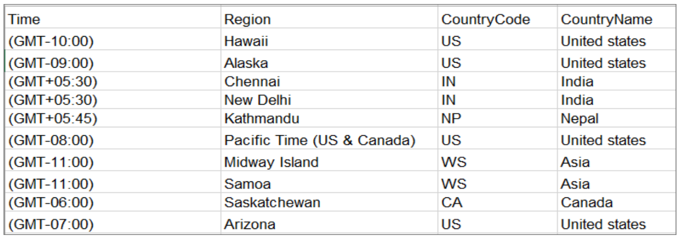
Step 1: Twitter provides the following time zone information i.e. by Region and GMT offset. Please see Figure 3 below.



Time	Region
(GMT-10:00)	Hawaii
(GMT-09:00)	Alaska
(GMT+05:30)	Chennai
(GMT+05:30)	New Delhi
(GMT+05:45)	Kathmandu
(GMT-08:00)	Pacific Time (US & Canada)
(GMT-11:00)	Midway Island
(GMT-11:00)	Samoa
(GMT-06:00)	Saskatchewan
(GMT-07:00)	Arizona

Figure 3. Twitter provided time zone information (twitter4j)

Step 2: Manually the Country code for a particular region is looked up and the corresponding country is identified. Please see Figure 4 below.



Time	Region	CountryCode	CountryName
(GMT-10:00)	Hawaii	US	United states
(GMT-09:00)	Alaska	US	United states
(GMT+05:30)	Chennai	IN	India
(GMT+05:30)	New Delhi	IN	India
(GMT+05:45)	Kathmandu	NP	Nepal
(GMT-08:00)	Pacific Time (US & Canada)	US	United states
(GMT-11:00)	Midway Island	WS	Asia
(GMT-11:00)	Samoa	WS	Asia
(GMT-06:00)	Saskatchewan	CA	Canada
(GMT-07:00)	Arizona	US	United states

Figure 4. Time zone mapped to Country

[4] Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

III. Topology

The topology is a network of spouts and bolts and we have used 1 spout and 4 bolts to achieve the sentiment analysis. Please see Figure 5 below.

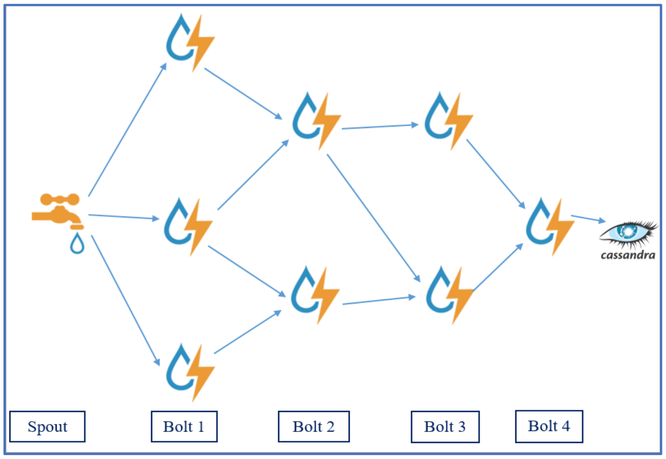


Figure 5. Our Topology

The topology starts with the spout and ends with the fourth bolt. Please see Table 1 below for the input and output at each stage of the topology.

	INPUT	OUTPUT
SPOUT	HDFS File/Twitter Streaming API	Tweets + other related data such as time zone, user details, media, geolocation etc.
BOLT 1	Tweets + other related data such as time zone, user details, media, geolocation etc.	Sentiment (Positive, Negative, Neutral) & Time zone of tweet
BOLT 2	Sentiment (Positive, Negative, Neutral) & Time zone of tweet	Sentiment & Country mapped to time zone
BOLT 3	Sentiment & Country mapped to time zone	Count of positive, negative, neutral sentiment (By country) & country
BOLT 4	Count of positive, negative, neutral sentiment (By country) & country	Store to Cassandra DB

Table 1. Our Topology

What is Cassandra DB? It is an open source distributed database management system designed to handle large amounts of data across many commodity servers[5].

Why Cassandra DB? The Apache Cassandra database is the right choice when scalability and high availability is needed without compromising on performance. The proven fault tolerance on commodity hardware and ability to provide low latency for users makes it the perfect platform for mission-critical data[6].

IV. VISUALIZATION

A. *Where in the world are people tweeting now*

1. Setting up an endpoint to connect to Twitter Streaming API using the twitter4j library.
2. Parsing the data using twitter4j library that internally uses a JSON parser and the **count of tweets** is stored in the Cassandra DB at the final bolt.
3. There are 2 Java Servlets that are used, first one creates a map instance that is sent it to the web browser.
4. Web browser periodically requests the data servlet to provide data, which in turn fetches it from the Cassandra DB.

Please see Figure 6 for a snapshot.

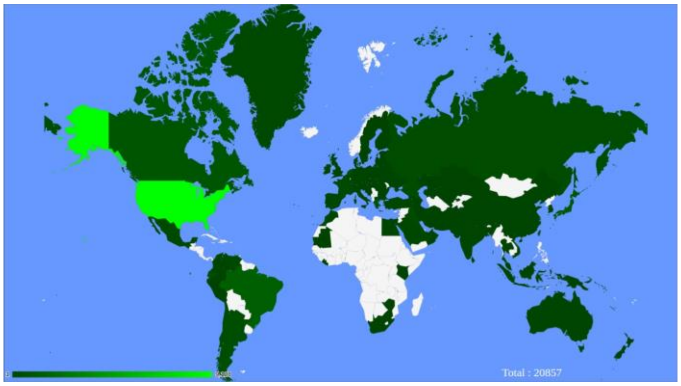


Figure 6. Count of tweets in the world (Twitter Streaming API)

[5] Apache Cassandra | Wikipedia

[6] Welcome to Apache Cassandra | cassandra.org

Analysis: The code was run for 10 minutes and the number of tweets processed was 20,857.

1. The number of tweets is more in the US and western countries (light green signifies more tweets) and lesser in the developing countries.
2. In Africa, as you may notice, there is almost a negligible twitter presence indicating the shortage of power and internet reach in the continent and this analysis correlates with the general thought of how internet reach and power shortage can be potential barriers for development of a nation.

B. *Sentiment Analysis of the tweets over different time zones mapped to country*

1. Setting up an endpoint to connect to Twitter Streaming API using the twitter4j library.
2. Parsing the data using twitter4j library that internally uses a JSON parser.
3. The **sentiment and count of tweets** is stored in the Cassandra DB at the final bolt.
4. There are 2 Java Servlets that are used, first one creates a map instance that is sent it to the web browser.
5. Web browser periodically requests the data servlet to provide data, which in turn fetches it from the Cassandra DB

Please see Figure 7 for a snapshot.

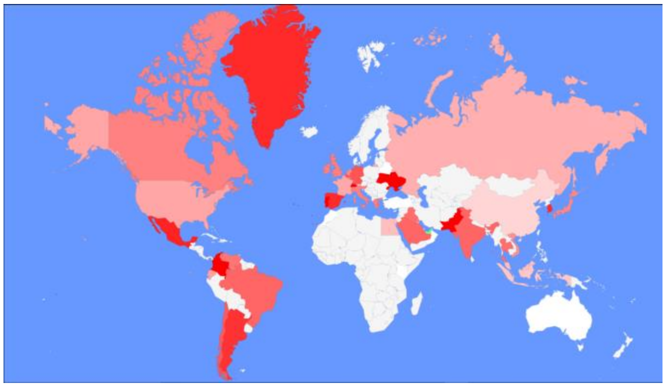


Figure 7. Sentiment of tweets in the world

Analysis: The code was run for 20 minutes and the number of tweets processed was 45,675.

The dark red colour signifies negative sentiment and the lighter pink signifies positive sentiment and white signifies wither neutral sentiment or lack of twitter presence.

We notice that few parts of South and North America and India show a higher negative sentiment relative to other countries.

This visualization is in synchronisation with the previous one for Africa where Twitter is not present because of power shortage or lack of internet reach.

V. SCALABILITY

For the purpose of scalability, we increase the rate at which the tweets were streamed using a simulator. Please see Figure 8 for the results.

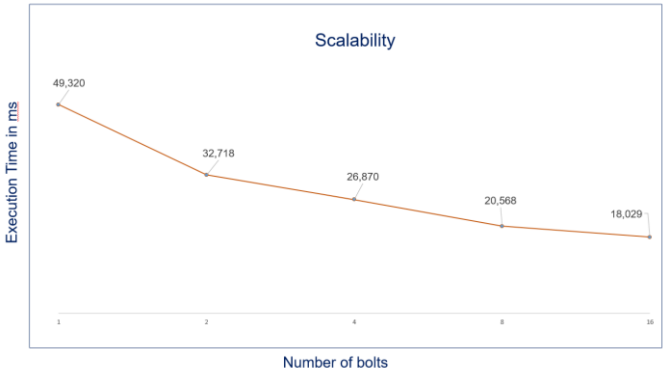


Figure 8. Scalability

Analysis: The rate at which the tweets were streamed were increased with the increase in the number of bolts on the cluster. We observe a decreasing trend in the execution time (in ms) till 16 bolts scenario, post which the execution is time is almost stable at ~18-17 s with increase in the number of bolts.

VI. CONCLUSION

In summary, we have evaluated the Emoticon approach and then took the NLP deep learning model for Twitter Streaming data (NLP java library) and Storm enabled us to process real-time, low latency data in a distributed fashion.

Cassandra DB played a key role in providing a distributed, fault tolerant database that helped to achieve scalability. The data is visualized using Java web servlets and the web browser by taking inputs from the Cassandra DB. While we have explored the sentiment analysis and number of tweets in real-time for the data publicly available, full data access to Twitter data such as GNIP (Twitter enterprise API platform) can enable companies to understand the sentiment of their products and the brand image in the eyes of the public.

REFERENCES

[1] Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., and Potts, C., Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, In: Proceedings of the Conference on Empirical methods in Natural Language Processing (EMNLP) (2013)

[2] Pang and Lee, L., 2008. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1–135.

FOOTNOTES

1. Internet live stats | Twitter Usage Statistics
2. Twitter Usage | Company Facts
3. The Streaming APIs | Overview
4. Recursive Deep Models for Semantic Compositionality Over Sentiment Treebank | EMNLP
5. Apache Cassandra | Wikipedia
6. Welcome to Apache Cassandra | cassandra.org