# “Sentiment Analysis of Tweets and News using Big Data”

## A PROJECT

***Submitted in partial fulfillment of the requirements for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Under the supervision of

**Dr. Hemraj Saini**

**(Associate Professor)**

***By***

***Prateek Goel (141236)***

**to**

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY**

**WAKNAGHAT, SOLAN – 173 234**

**HIMACHAL PRADESH, INDIA**

**May-2018**

# CERTIFICATE

I hereby declare that the work presented in this report entitled **"Sentiment Analysis of Tweets and News using Big Data"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2017 to May 2018 under the supervision of **Dr. Hemraj Saini,** Associate Professor, department of Computer Science & Engineering.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

**Name:** Prateek Goel

**Rollno.:** 141236

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Hemraj Saini

(Associate Professor)

Department of Computer Science & Engineering

Dated:

# ACKNOWLEDGMENT

I take upon this opportunity endowed upon me by grace of the almighty, to thank all those who have been part of this endeavor.

I want to thank our supervisor 'Dr. Hemraj Saini’ for giving me the correct heading and legitimate direction in regards to the subject. Without his dynamic association and the correct direction this would not have been conceivable.

Last however not the minimum, I generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Prateek Goel

141236

Computer Science and Engineering

# TABLE OF CONTENTS

# LIST OF ABBREVIATIONS

| *Abbreviations* | *Explanation* |
|---|---|
| ***FP*** | False Positive |
| ***TP*** | True Positive |
| ***FN*** | False Negative |
| ***TN*** | True Negative |

# LIST OF TABLES

# LIST OF FIGURES

# ABSTRACT

In this manuscript, a comprehensive survey of sentiment analysis of tweets, news and emoticons is presented. The survey includes the articles published in recent years that focused on different aspects related to sentiment analysis. Then, there is a review of models, techniques or methodologies used in sentiment analysis. In addition, a taxonomy and comparison of techniques for sentiment analysis in form of tables in four terms, namely, sentiment analysis category, goals, main processes, and computation complexity is given. Based on the current survey, identified open issues and suggested hints for future research. Finally, a detailed framework is also proposed to infer knowledge about reviews identification for advanced countermeasures for needy users. The framework includes Machine Learning technique for knowledge discovery as opinion along with an efficient pre-processing of collected Big Data consisting of related tweets, News and Emoticons to discover the possible opinion.

**Keywords**: Sentiment analysis, Big Data, Machine learning, Medical Data Sentiment Analysis (MDSA).

*Chapter 1*

# INTRODUCTION

## 1.1 Sentiment Analysis

Opinion Mining, that we technically call Sentiment analysis is a process that is implemented for identifying and categorizing the opinions being expressed inside a block of the text, primarily in order to figure out what author is saying as well as mind set towards the concerned topic or the product using some algorithms. The process of Sentiment Analysis is the detection of sentiment score of a block of text. Or else we can say that, the process shows whether some block of text is negative, positive or neutral.

## 1.2 Twitter Data

Twitter.com, one of the largest social blogging website that makes the exchange of tons of tweets each day in the size of about Zetabytes per year. Massive unstructured datasets, that could be used by large business organisations by making computations according to their own requirements and performing some computations is provided by the website.

## 1.3 Point-by-Point Mutual Information

The process is a mathematical tool, which helps in finding a relationship of one word with the other. A formula which calculates every word associated in the document to already defined bag of adjectives for determining exact sentiment related to the word.

## 1.4 Hadoop Framework

The Apache Hadoop framework based on Google File System was developed for solving the problems having tons of unprocessed data. Divide and Rule technique was used for the processing of datasets. It makes us deal with large as well as complex structured as well as unstructured data that has makes problems while creating normal tables. Since, the Twitter big data is relatively unstructured so it can easily be stored using Hadoop Distributed File System or HDFS. The Hadoop framework also works good for the various tasks like search engines, online retailing, future recommendations etc

## 1.5 Hadoop Distributed File System

Hadoop provides a file system that sits on the existing existing operating system and works on the commodity hardware. This file system is known as Hadoop Distributed File System or the HDFS. The system is known for its high fault tolerance and is implemented on low maintenance machinery so the main focus could be put on the technicalities rather putting on the machinery. It has a high throughput. So, for dealing with the massive datasets, this proves to be the best. It is much the same as an ace slave design which has a solitary name node which controls the file system get to. The data node handles the reading and writing requests from the clients of the file system. Apart from this, but it performs creation of the blocks, the deletion of blocks, and the replication of the blocks as well on being instructed by the Name node.

Information replication is improved the situation accomplishing adaptation to non-critical failure. The expansive information group is put away as a succession of pieces. Piece measure and the replication factor can be physically chosen. The factor of Replication is 3 by default, that means there will be 3 duplicates of the each data block will be there at an instant of time in the Hadoop cluster.

## 1.6 The approach in our project

Here, I have underscored fundamentally on the precision of performing examination than its speed by making it into account on enormous information which is accomplished by part the different modules of information in following advances and teaming up with Hadoop system for mapping it onto diverse machines grammatical feature (POS tagging) labelled utilizing open NLP. This grammatical form labelling procedure is utilized for various purposes.

### 1.6.1 Removal of Stop words

Words like a, an, the are called stop words, which do not play any role in depicting the sentiment are vanished in this step. All the stop words are not considered.

### 1.6.2 Unstructured to structured

Tweets are mostly unstructured like the word ‘awesome’ as ‘awsm’, ‘happyyyyyy’ is ‘happy’ in reality. The final transformation of the unstructured to structured is accomplished by the data records that are dynamic and are transformed from unstructured to structured and the vowels would be added.

### 1.6.3 Emojis

Emoticons are the most expressive strategy accessible for opinion. These emoticons are commonly known as emojis are symbolic representation formed by a combination of symbols are transformed to corresponding words at this step i.e. happy.

### 1.6.4 Live streaming data and the features

Twitter provides us streaming APIs, that allow us to get like real time data which is required for our computations. The purpose of development is accompanied by twitter through it’s APIs, which permits the developer to access approx. 1-2% of the tweets posted at that time based on a keyword required. So, the twitter API takes the required word as the user input and provides us the necessary dataset by mining the twitter database.

Each tweet comprises has a limit of 140 characters. Additionally, it permits the utilization of emojis which are immediate markers of the client's view on the subject. Tweets additionally comprise of date and time as well as the client name. The date and time are helpful for speculating the future pattern of our task. The location of the user, if accessible can likewise plot the patterns in various geological areas.

### 1.6.5 Base Form

All the words present in the tweet are changed over to their root form to keep away from the undesirable additional data storage of the inferred words. The root shape informational collection is utilized to do what is made nearby as it is vigorously utilized is program, which brings down the entrance time and expands the general proficiency of the calculation.

### 1.6.6 Sentiment Directory

Sentimwordnet directory is used to create the sentiment words directory. All conceivable use of a specific word are viewed as, similar to "great" can be utilized as a part of a wide range of ways every way having its own feeling an incentive for each situation. Along these lines, the net supposition estimation of good is acquired from every one of its structures utilized and put away in an index which is neighbourhood (i.e. in essential memory) with the goal that time ought not be squandered in looking through the word in the auxiliary memory stockpiling.

### 1.6.7 Accuracy

Modules like opennlp, wordnet and sentimwordnet need to be accessed to find the actual sentiment related to the tweet. So, the general exactness of undertaking is controlled by time required to access from such modules.

As all components are in series, the hypothetical general exactness of the program is the result of precision of the considerable number of modules.

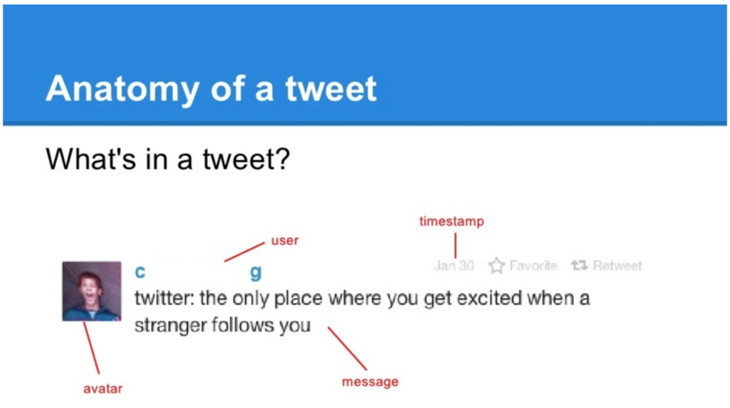
### 1.6.8 Time Efficiency

Time efficiency is the aspect of which we have taken care at every stage of our project. Data structures have played an important role in making us reduce the response time. The best decision of ours was to use the primary memory in a number of steps of our algorithm. So, there comes a reduction in the hard disk’s memory access time.

Usage of the Hadoop framework ensured that the processing would be distributed and which in turn lowered the access time as well as the time of execution. So, the overall time efficiency will increase with taking into account the above mentioned factors.

### 1.7 Twitter Data

What is a tweet?



Anatomy of a tweet

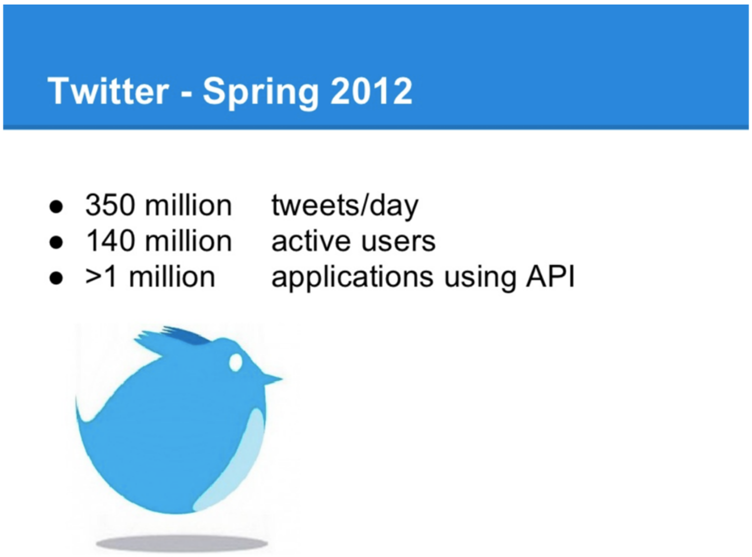
What's in a tweet?

c g

Jan 30 ☆ Favorite Retweet

twitter: the only place where you get excited when a stranger follows you

How large the data is?



Twitter - Spring 2012

- 350 million tweets/day
- 140 million active users
- >1 million applications using API

*Chapter 2*

# OBJECTIVE

## 2.1 Existing System

As already been discussed the manual way of collecting data and performing the sentiment analysis on the data. Here some coding techniques were being used for crawling the related tweets, where the data can be extracted from the Twitter social blogging site with the help of any of R, Python, Java etc. For the same, some libraries are downloaded that allow the crawling and accessing of data that we want particularly. After that, the raw data will be filtered by using some old techniques and carried out the positives, the negatives and the neutral words from a collected words list as a text file. The word corpus needs to be made separately for performing sentiment analysis each time. This corpus can be called as a dictionary set. After performing all these steps, this corpus is stored in the primary memory.

## 2.2 Proposed System

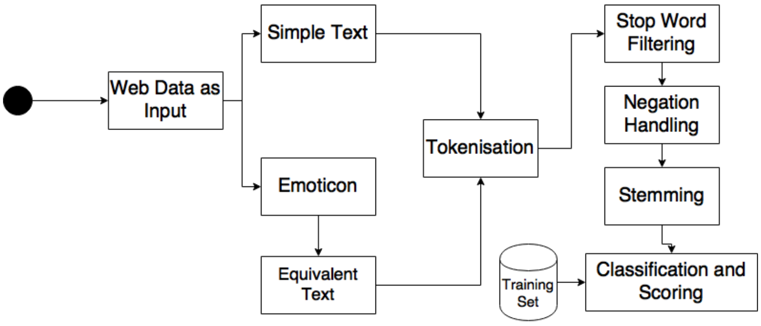
We have seen the drawbacks of existing system. So, here we are going to overcome some of those by using:

1. Machine Learning technique, which includes Naïve Bayes algorithm.

   In this, we used Naïve Bayes algorithm to find negative, positive and neutral words so as to calculate the overall score of the tweet to predict the actual sentiment behind the tweet.

   The twitter API allows the data to be downloaded, that further and is stored as a form of csv file. Not only this, but two more csv file will be created after performing NLP (Natural Language Processing) steps.

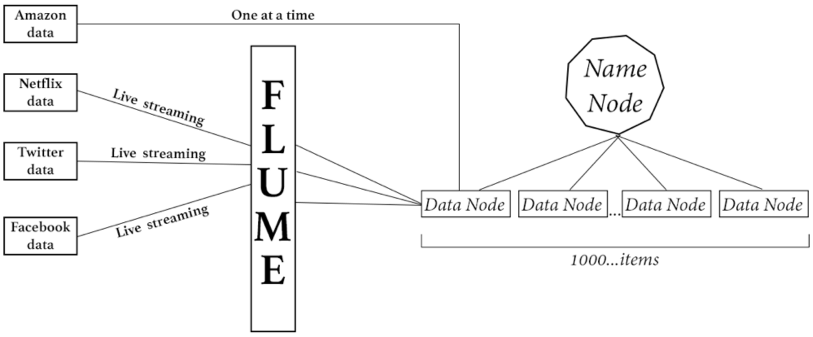
The following figure depicts the architecture view for the proposed system.



2. Apache Flume framework of Big Data.

Here we are going to use Hadoop and its Ecosystem Apache Flume, for fetching raw data from Twitter. This tool consists of configuration file that needs to be amended as per our requirements and everything that we want to get data from the Twitter. For this, we have to define what information we want to get form Twitter and the specifications of the API we are going to use. All the data will be saved into the HDFS (Hadoop Distributed File System)[12] in our prescribed format. From this data, we are going to create a table and implement NLP techniques to filter the contents that are actually required.

The following figure depicts the architecture view for the proposed system. How the data is going to be stored with the help of Flume [8].



## 2.3 Goals

- To implement an algorithm for automatic classification of tweets' text into positive, negative or neutral.
- Graphical representation of their respective sentiment.
- Create a WordCloud of the words used in the tweet based on the positive opinion, negative opinion or neutral opinion.

*Chapter 3*

# LITERATURE SURVEY

In the recent years, a lot work is being done in the domain of sentiment analysis. Sentiment analysis implementation is being performed for a variety of applications with the help of a number of algorithms and for varying data size. Here we have read and discussed about some of the work done by various people across the globe.

## 3.1 Research Papers Analysis

- **Alek Kolcz, Jimmy, and Lin, "Large-Scale Machine Learning at Twitter." In the Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 793-804. ACM, 2012. [3]**

This work tells us a technique for finding the drug users along with the latent adverse events from analysing the text of tweets with the implementation of NLP, i.e. Natural Language Processing and created SVM, i.e. Support Vector Machine classifiers. Since they were dealing with a billion tweets so it was large sized dataset, the experiments were being conducted on a system with very high performance capacity of computing with the help of Map & Reduce technique, that helps depict the trend of analytics done by big data. The outcome here recommended the regular day to day existence person to person communication information could help early recognition of patient's crisis wellbeing issues.

Dataset being used here is a mass of tons of Tweets that was piled over a period of May 2009 to October 2010, from which the author identified potential adverse consequences of some chosen drugs. Data collected using the twitter data stream was organized in a timeline. The unstructured raw data from Twitter was mined using the Twitter streaming API which permits us to get relevant data from the twitter database.

The work is catalogued considering the following fields for every single Tweet:

- ID of every tweet to make every single Tweet stand out with its own identity.
- ID of the user or the twitter handle linked with every single Tweet.
- The date and time with every Tweet.
- Text of the Tweet.

The author employed Elastic Computed Cloud 2 of Amazon (Amazon EC2) for running the Twitter catalogues on 16 different occurrences, that has 34 Gigabytes of memory, and 12 Computing Units with EC2 working in unison, which enable us to mine and index each of the billion tweets only in three days time. Storage of the indices is 890 Gigabytes.

- For mining the tweets, the process could be separated in two parts.
- Identify the dormant users of the drug.

To find the viable consequences displayed in the users' timeline that may be a consequence of some drug. The processes involve the building phase and the training phase of the models for classification based on the attributes brought out from the users' tweets.

The two main sets of the attributes, namely, semantic and textual are brought out from the concerned users' timeline for all these models for classification.

Some features of the texts like the models named the bag of words are carried out depending on our testing of all the Tweets. The significant features are carried out from the concept codes of UMLS, i.e. Unified Medical Language System Metathesaurus taken from all Tweets with the help of Metamap that was developed by NLM, i.e. National Library of Medicine. SVM, i.e. the Support Vector Machine with two classes played a vital role meeting the purpose classification.

Classifiers of SVM were gauged using factors like the area under curve and the curve of Receiver Operating Characteristic. Receiver Operating Characteristic curve using the average values of 1000 repetitions was outlined. The average predicted accuracy of over

1000 continuous iterations was found out to be 0.74 and the mean Area under the Curve value was 0.82.

- **Bingwei, Erik Blasch, Dan Shen, Yu Chen and Genshe Chen. "Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier", International IEEE Conference on Big Data, pp. 98-103, IEEE, 2013. [5]**

Technologies of machine learning and NLP are widely being used for the grouping of the sentiments as they have a potential to “learn” the trend shown by the dataset provided for training to foretell or just boost up the decision making process with comparatively better accuracy level. Nonetheless, when dealing with a massive dataset, a number of algorithms might fail to scale up well. Here, they have tried to evaluate the reliability of the Naïve Bayes classifier in large datasets. Authors have brought up an efficient and reliable system for scanning the sentiments of massive datasets with the help of ‘Naïve Bayes classifier’ accompanied by the Hadoop ecosystem. They implemented Naïve Bayes classifier in place of the Mahout Library, for having the grain level control of the procedure of analysis for the implementation of Hadoop frameworks. The authors demonstrated that Naïve Bayes classifier is able to cope up with millions of tweets for analyzing the sentiment related to reviews of some movie with increased accuracy level.

The unstructured raw data was collected from massive sets of reviews of movie mined by various research communities. Here, the authors have used two datasets:

- Movie review of The Cornell University
- Amazon movie review of Stanford SNAP

The dataset from Cornell had about 1000 positive and negative reviews each. Each review having information like the product ID, profile Name, user ID, summary, score etc. were organized in eight lines per review.

Only unigrams were casted off by them for the Naïve Bayes classifier.

This classification is further classified into following sequential jobs:

1) The training job: At this stage, the reviews are used to train the model for finding all the alike words with their respective negative or positive review in the related document.
2) The combining job: At this stage, the reviews mined for testing are combined to our classification model to create an intermediate table with all the required information for the final classification.
3) The classification job: The reviews are then classified at this stage accordingly and the results are further stored in the HDFS.

The framework they worked on consisted of a Virtualized cluster of the Hadoop framework having several nodes. It has a high speed and is a very reliable technique of testing a Hadoop based program on the cloud environment, though the performance nay degrade from a real Hadoop cluster. Their cloud environment is built on a server with 64GB memory within 15 Intel Xeon E5- 3633 2.5GHz cores.

The authors implemented the code on the dataset from Cornell and found out approx. 78.6% mean accuracy. In need of undergoing any change in the code of Hadoop system, program easily classified various other elements of the dataset of review of the movies of Amazon with almost equivalent similarity index. For testing the reliability of the Naïve Bayes classifier, the experiment was performed with a varying dataset from thousand to around a million reviews for each class.

- **María D. R-Moreno, ÁlvaroCuesta, and David F., "A Framework for Massive Twitter Data Extraction and Analysis", Malaysian Journal of Computer Science, pp 50-67 (2014):1. [6]**

Here author proposes the open source structure that automatically collects and analyses the data from Twitter public API. Here the framework is extensible and can be customized, so it can be used by the researchers for testing some new techniques. Being accompanied by a language based module, the framework provides some tools together to perform sentiment analysis of the related tweets.

What this platform is capable of doing is explained with two Spanish study cases,

1. Regarding a high impact Boston terror attack.
2. Regarding a regular political activity on Twitter.

Boston's case study calls for the twitter activity revolving around a whacky event, i.e. Boston Terrorist Attack. In this instance, author used a hashtag to find the relevant tweets. The other case study was focused on regular usage of Twitter, performing activity tracking of well-known Spanish political people, like political parties, politicians, activist organizations and the journalists too. Here, authors have considered some debatable accounts to have good basis for sentiment analysis.

There are various processing layers and the data needs to interchanged among these modules, using some open source data formats like JSON. Here most tools in this structure are implemented in R or Python, but the tester and the classifier web interfaces ran with NodeJS and were programmed on CoffeeScript (a language which is JavaScript pre-processed). MongoDB was the chosen backend database, which fits good for our purposes too since its atomic representation is JSON, just like the tweets. Overall implementation was done using the Natural Language ToolKit (NLTK).

The overall process of extraction of data and the sentiment analysis is categorized into three different steps:

1. Data acquisition
2. Training for sentiment analysis
3. Report generation.

The primary step is to gather data from Twitter with the help of twitter API. Then our classifier training is done after which, the final sentiment analysis is carried out. At the end, a set of reports are generated by the platform, counting sentiment analysis if it is enabled. Three-class-classification was done, i.e. 'negative', 'neutral' and 'positive'. A number of Naïve Bayes classifiers using a set of n-grams in order to choose the one that performs the best. In particular, they have tried {1}, {2}, {3}, {1, 2}, {1, 3} and {2, 3} kind of n-grams and a least score of 0, 1, 2, 3, 4, 5, 6 and 10.

Various techniques were tried using tenfold cross validation to avoid biasness made by the partition of our training set. The parameters like mean accuracy and accuracy variance, precision, fmeasure mean and variance and the recall were used for estimation. The cessation is that the best trainers had un-grams included and the minimum score was between 2 and 4.

- **Andrzej Romanowski, Michal, and Skuza, "Sentiment analysis of Twitter Data within Big Data Distributed Environment for Stock Prediction", 2015 Federated Conference on, pp. 1349-1354. IEEE, 2015. [7]**

The paper discussed a stock market prediction possibility based upon grouping of data coming from the required tweets from Twitter micro blogging platform.

Tweets were mined in real time with the help of Twitter API. A number of tweets were collected in a period of over 3 months starting from January 2, 2013 to March 31, 2013. Query made it already clear that the tweets have to involve names of the company or hashtag related to it. Prophecies were made for Apple Inc. to make sure that possibly massive datasets would be created.

Tweets only in language English were used in this project work. Retweeted posts were considered redundant for classification, so were removed. After data pre-processing, each tweet was saved as a model of bag-of-words, a standard technique for simplifying the represented information used in information retrieval.

The design of the system consisted of four major components:

- Retrieving, pre-processing and saving the twitter data to our database
- Stock market data retrieval
- Building the Model
- Future stock prices Prediction

Polarity analysis is that part of sentiment analysis, in which the input is grouped either as positive or negative. Programmed sentiment detection of tweets was pulled off by using

SentiWordNet. Future stock prices prediction is performed in this paper by joining the results of classified sentiment tweets and stock prices from some past interval.

Considering tons of data to be categorized and the fact they are a written text, the Naïve Bayes algorithm was selected for its fast process of training even with massive amount of training data and the fact that it can be increased. Considering tons of data also resulted in decision to implement the map-reduce version of Naïve Bayes algorithm.

- **Mohit, Tare, Indrajit Gohokar, Devendra Paratwar, Jayant Sable, Rakhi Wajgi. "Multi-Class Tweet Categorization Using Map Reduce Paradigm" International Journal of Computer Trends and Technology. pp 78 - 81 (2014) [8]**

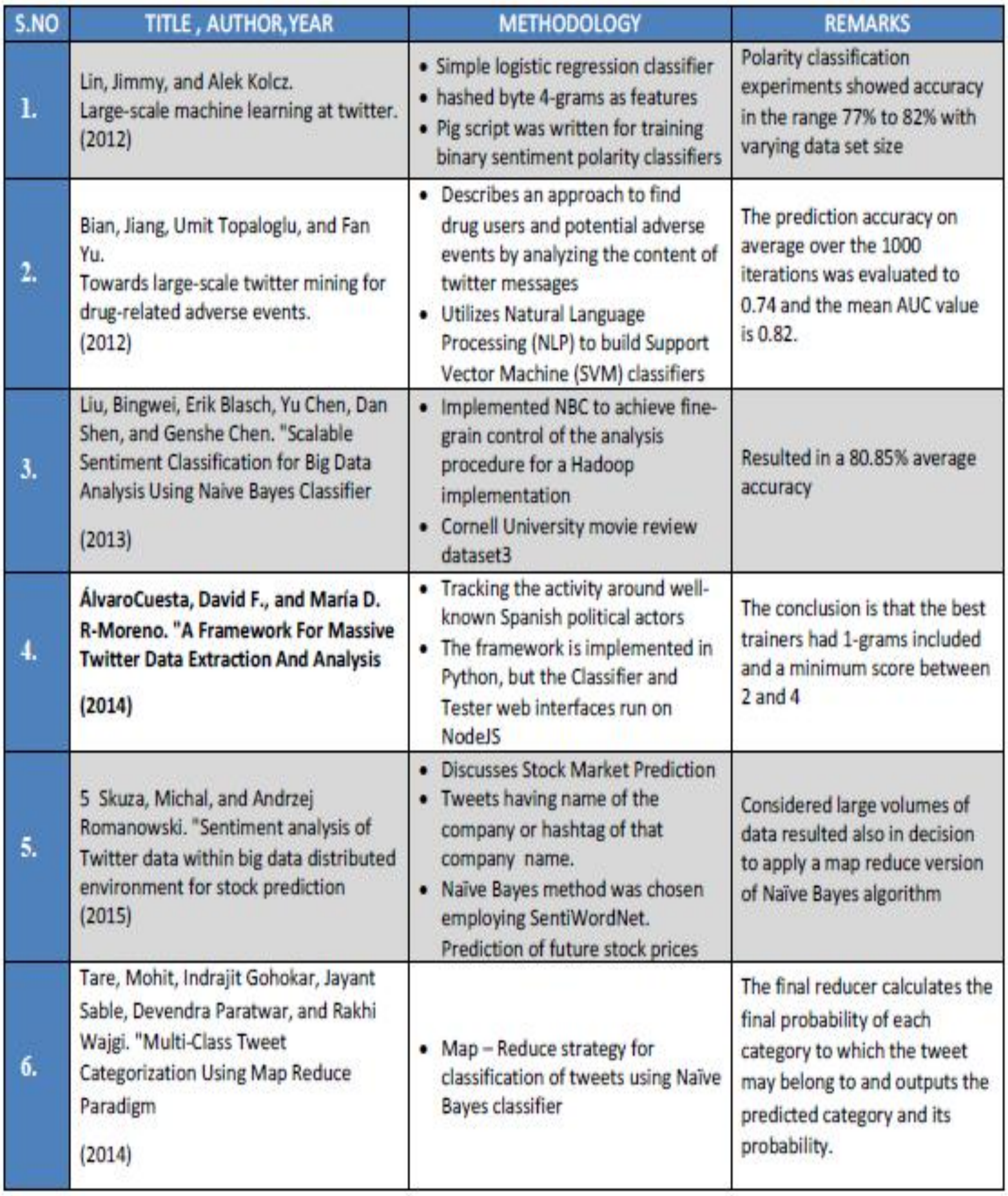
This paper proposes a strategy that used Apache framework of Hadoop, which is an open source framework that uses java, that clings on the Map-Reduce paradigm and the Hadoop Distributed File System (HDFS) for processing the dataset. The profiled strategy having Map-Reduce for the classification of tweets using the Naïve Bayes classifier sits upon two Map-Reduce passes.

They have used the library 'Twitter4j', for gathering the tweets which in turn uses the twitter REST API. This twitter library requires OAuth to access the API. Twitter uses the OAuth to provide authorization to its API.

Finally, after preprocessing, the tweets were labeled based on the categories namely technology, sports and politics.

In first Map & Reduce pass, the mapper function took various labeled tweets from the training data and puts the category and the word as a key-value pair. Then the Reducer summing up all instances of the words for each category, outputs the category and the word count as a key-value pair. The Map-Reduce in such a way deals with the formation of a model for classification.

Next Map-Reduce pass performs the classification by calculating the conditional probability of each word followed by putting the category and the conditional probability of each word as a key-value pair. Then, finally the reducer evaluates the final probability for every single category which the tweet may belong to and puts the anticipated category and the corresponding probability value as a key-value pair.



| S.NO | TITLE , AUTHOR,YEAR | METHODOLOGY | REMARKS |
|---|---|---|---|
| 1. | Lin, Jimmy, and Alek Kolcz.<br>Large-scale machine learning at twitter.<br>(2012) | • Simple logistic regression classifier<br>• hashed byte 4-grams as features<br>• Pig script was written for training binary sentiment polarity classifiers | Polarity classification experiments showed accuracy in the range 77% to 82% with varying data set size |
| 2. | Bian, Jiang, Umit Topaloglu, and Fan Yu.<br>Towards large-scale twitter mining for drug-related adverse events.<br>(2012) | • Describes an approach to find drug users and potential adverse events by analyzing the content of twitter messages<br>• Utilizes Natural Language Processing (NLP) to build Support Vector Machine (SVM) classifiers | The prediction accuracy on average over the 1000 iterations was evaluated to 0.74 and the mean AUC value is 0.82. |
| 3. | Liu, Bingwei, Erik Blasch, Yu Chen, Dan Shen, and Genshe Chen. "Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier<br>(2013) | • Implemented NBC to achieve fine-grain control of the analysis procedure for a Hadoop implementation<br>• Cornell University movie review dataset3 | Resulted in a 80.85% average accuracy |
| 4. | **ÁlvaroCuesta, David F., and María D. R-Moreno. "A Framework For Massive Twitter Data Extraction And Analysis**<br>**(2014)** | • Tracking the activity around well-known Spanish political actors<br>• The framework is implemented in Python, but the Classifier and Tester web interfaces run on NodeJS | The conclusion is that the best trainers had 1-grams included and a minimum score between 2 and 4 |
| 5. | 5 Skuza, Michal, and Andrzej Romanowski. "Sentiment analysis of Twitter data within big data distributed environment for stock prediction<br>(2015) | • Discusses Stock Market Prediction<br>• Tweets having name of the company or hashtag of that company name.<br>• Naïve Bayes method was chosen employing SentiWordNet. Prediction of future stock prices | Considered large volumes of data resulted also in decision to apply a map reduce version of Naïve Bayes algorithm |
| 6. | Tare, Mohit, Indrajit Gohokar, Jayant Sable, Devendra Paratwar, and Rakhi Wajgi. "Multi-Class Tweet Categorization Using Map Reduce Paradigm<br>(2014) | • Map – Reduce strategy for classification of tweets using Naïve Bayes classifier | The final reducer calculates the final probability of each category to which the tweet may belong to and outputs the predicted category and its probability. |

**Table 3.1** Some past work by researches in the same field

*Chapter 4*

# ALGORITHM DESIGN AND METHODOLOGY

A functional classifier can be made with the help of following five basic categories:

1. Acqiosition the Data
2. ID of the User
3. Extraction of the Features
4. Grouping
5. Web based Application for Analysis of Tweets

## 4.1 Acquisition of Data

Raw tweets in the form of unstructured data were collected by using the twitter API which provides a package for simple access of twitter database. This API works in two modes:

1. Sample Stream
2. Filter Stream.

The first one delivers a small and random sample of all the tweets streaming at a real time. The other one delivers the tweet which matches a certain criterion. It may filter the selected tweets on the basis of following three criteria:

1. Some specified keyword(s) to be searched in the tweets
2. Some specified twitter user(s) on the basis of their user-id's
3. The tweets that are originated from some specific location(s) using geo-tagging.

The programmer can choose any one or a combination of these filtering criteria, but for meeting our requirements, i.e. we have no restrictions, so we'll new to get stuck with the sample stream mode only.

Though we wanted our data to be more general, we obtained it in different portions at various points of time, instead of obtaining all of it at one time. If we used the second approach, then the generality of our tweets possibly have been compromised as a large portion of the tweets would be pointing to some certain ongoing topic and would thus have similar general mood or common opinion. This was experienced while going through the sample of obtained tweets. For example, the data sample obtained around Christmas and New Year's eve had a major portion of data referring to the joyous feeling and were in return had a general positive sentiment. Illustrating the data in portions at different points of time would thus try to cut down this issue. Henceforth, we obtained data at four unlike points which were December 17'2011, December 29'2011, January 19'2012 and February 8'2012.

The tweet obtained in such a way have a lot of unstructured information which may or may not be useful for our desired throughput. It usually comes in the form of various key-value pairs. Following is a list of some key-value pairs:

- Favorite or non- favorite tweet
- ID of the user
- Username
- Tweet's text
- Hashtags
- Re-tweeted or not
- Account registration language
- Geo-tagging of the location with the tweet
- Timestamp of the tweet

As this is tons of information, so we filter out only the information that we need and omit the rest. For our specific application, we go through each and every tweet in our dataset and put the actual textual content of each tweet in another file ensuring that the language for user's account is specified to be English. The real textual content of the tweet is provided under the dictionary key '**text**' and the language of the user's account is given under '**lang**'.

Since labeling of the user turns out to be very pricy, so we need to sieve the tweets such that they can be labeled for achieving the greatest amount of disparity in tweets without losing generality. Following are the filtering criteria used:

- Removal of the Re-tweeted posts
- Removal of tiny little tweets, i.e. tweets with a total length not more than 20 characters
- Removal of the tweets that are in language other than English
- Removal of the same tweets to avoid redundancy.

After being filtered, approx. 33% of the tweets remain for labeling the users on average per data sample, which created a collection of 20,175 unlabeled tweets.

### 4.2 ID of the user

For labeling the human, we need to make three duplicates of each tweet so that we can label them by four different sources. The purpose has to be solved so that we can take a general opinion of the people on the related sentiment of the tweet and thus the error and noise in labeling can be reduced. The crux is that more the number of copies of the labels we can get, the better it is but we need to take care of the labeling cost, so we decided to take the reasonable factor of 3.

The tweets were labeled in three different classes as per the sentiments expressed or observed in the tweets: positive, negative and neutral. Following are the guidelines using which the tweets were marked:

1. **Positive Tweet**: The tweet with an entire happy/positive/joyful/excited attitude or there is a presence of some positive implication. Not only this, but if there is more than a single sentiment being expressed with the help of the tweet but the positive attitude is overpowering. For Example: "*And 2 more years of being in this shithole China, then I'll be moving to America! :D*".
2. **Negative Tweet**: The tweet having a displeased/sad/negative kind of attitude or may have something that is mentioned with a negative connotation is considered to be negative tweet. Not only this, but if the tweet has more than one sentiment and the negative

sentiment is overpowering. For Example: "*I need an iPhone coz now this Android gets stuck :S*".

3. **Neutral**: The tweet expressing no exact sentiment or the opinion of the user and just provides some information is considered to be a neutral tweet. Just like the advertisements of various products could fall in this category.

4. **<Blank>**: Tweets in some language other than English were left unlabeled so that they could be ignored in the dataset while making computations.

Apart from this, the labeling features were trained to keep labeling out of any personal biasness and thus making no such assumptions like judging a tweet based on some past personal information but only using the provided information with each tweet.

So, once the tweets were labeled, combining the opinions of all the three people so as to get an averaged opinion would be our next step. We took help of the majority vote to do this. Like, if there is some tweet that had more than one label, we could label the overall tweet with the same attitude, and if all the labels were not same, we'll label that tweet "majority vote not received".

So, we formed the following statistical result using the majority voting technique:

1. Positive - 2550 tweets
2. Negative - 1907 tweets
3. Neutral - 4550 tweets
4. Unable to reach majority vote - 400 tweets
5. Non-English tweets - 370 tweets

Hence, considering such tweets for which we would be able to achieve a majority vote of negative, positive or neutral, we'll be having 9000 tweets to be used in the training set. Among these, approx. 4550 were objective and rest all were subjective tweets that could give up the sum of all the negative and positive tweets.

| | User 1 : User 2 | User 2 : User 3 | User 1 : User 3 |
|---|---|---|---|
| **Strict** | 48.7% | 69.6% | 61.6% |
| **Lenient** | 51.3% | 30.4% | 38.4% |

**Table 4.1:** The calculated user-to-user agreement for the task of labeling the tweets

The measure of "strict" agreement is shown by the above table. It shows us where the labels that were allotted by all the users should be identical in all three instances, whereas "lenient" is the one, in which if a tweet is marked "ambiguous" by a person but the other user marks it different, then it doesn't imply that it will be considered as a disagreement. Hence, in such a case, the classes could have been mapped to other classes. Since the user-to-user agreement stands in between 30-60%, proves that the act of opinion classification is one of the most tedious tasks for the human beings too.

| | Adjectives | Verbs |
|---|---|---|
| | **User 1 : User 2** | **User 1 : User 3** |
| **Strict** | 67.20% | 74.40% |
| **Lenient** | 78.80% | 78.70% |

**Table 4.2:** User-to-User Agreement for Labeling of Verbs and Adjectives

So, the strict estimation is done based on the grouping between three main classifiers, i.e. positive, negative and neutral, whereas the lenient one combines the positive and the negative classes and measure it as a one single class. So now the human beings are required only to classify the neutral and the subjective classes. All the results rehearse the primary claim that sentiment analysis is actually a tedious task. The final results we got were more than our agreement results because in this case humans are being asked to label individual words which is an easier task than labeling entire tweets.

### 4.3 Extraction of the Features

The hypothetical general exactness of the program is the result of precision of the considerable number of modules, these are:

- Tokenization : The process of separation a flood of content into words, images and other significant components called "tokens". Tokens can be isolated by whitespace characters as well as accentuation characters. It is done as such that we can take a gander at tokens as individual segments that make up a tweet.
- The content having tokens like ‘http’ or ‘@’ that are considered as URLs or act as a reference to the user would be removed as we are just dealing with the analysis of the tweet’s main text.
- Marks of punctuation as well as the numerical values might be expelled in the instances where we aim at contrasting the tweet with a rundown of words of English.
- Conversion of all the letters to Lowercase: For making the comparison easy with the English words corpus, all the textual content of the tweet is normalised, i.e. converted from uppercase to lowercase.
- The Stemming phase: The process of normalisation is the conversion of each word into its base or the root form. Just like happier would become happy. Since the comparison between grammatical conversions is very tangled, so this technique makes it easier for us to overcome such situations. Our model is based on the porter stemming technique implemented on the tweets as well as the words corpus.in short, it was done wherever we required comparisons.
- The elimination of Stop words: The words used by us in our daily conversations are known as stop words. Such words don’t provide any kind of information. Some of these are ‘the’, ‘his’, ‘an’, ‘she’ etc. Itiis some of the time helpful to evacuate such words since they provide no extra data as they are utilized similarly in all classifications of the content, like for instance while figuring earlier notion extremity of words in each tweet as indicated by their recurrence of event in various classifications and utilizing the same extremity to ascertain the general notion of the tweet over the bag of words utilized as a part of that tweet.

- POS Tagging: Grammatical form labeling is the way toward appointing a tag to every word in the sentence as to which linguistic form of grammar that the word has a place with, for example a thing, verb, descriptor, qualifier, organizing conjunction and so forth.

Since we have talked about a portion of the content organizing procedures utilized by us, we will shift towards the rundown of attributes that we have investigated. here, we can see that beneath an attribute is some variable that makes the classifier separate distinctive classes. We have two sorts of grouping in our framework (we'll discuss them in detail in the following segments), the arrangement based on objectivity / subjectivity and the negativity / positivity characterization. As recommended by the name itself, the previous is for separating amongst objective and subjective classification whereas the last one is for separating amongst positive and negative classes .

The rundown of the attributes investigated for the objective/ subjective grouping is as underneath:

- How many marks of exclamation are there in a tweet.
- How many interrogation marks are there in a tweet.
- Are there any marks of exclamation present in a tweet.
- Are there any marks of interrogation present in a tweet.
- Is there any url present in the tweet.
- Are there any emojis present in the tweet.
- The models of unigrams calculated by the Naïve Bayes classifier.
- Earlier extremity of words through online MPQA vocabulary.
- Quantity of the numerals present in the tweet.
- Total number of uppercase words in the tweet.
- The quantity of uppercase characters present in the tweet.
- The sum total of symbols as well as the punctuations present in the tweet.
- How many dictionary matched words are there to the number of dictionary unmatched words.
- Character count of the tweet.
- How many adjectives are there in a tweet

- What is the total number of comparative adjectives are there.
- The sum total of superlative adjectives in a tweet.
- The total number of normalized verbs present in the tweet.
- How many past form of the verb is there in the tweet.
- Number of present participle verbs in the tweet.
- The past participle verbs in a tweet.
- Presence of the 3rd form singular present verbs in a tweet.
- How many forms of singular present verbs other than 3rd form are there in a tweet.
- How many modifying words, i.e. adverbs are there in the tweet.
- How many first person pronouns in the tweet.
- The total of pronouns of possession in the tweet.
- All the singular proper nouns in the tweet.
- All the plural proper noun in the tweet.
- All the numbers of cardinality present in the tweet.
- All the possessive endings in the tweet.
- All the pronouns in the tweet.
- All the adjectives of all forms in a tweet.
- All the forms of the verbs in the tweet.
- All the forms of nouns in the tweet.
- Number of pronouns of all forms in a tweet.

The rundown of highlighted attributes investigated for positive/ negative arrangement are given beneath:

- The net score from the emojis (where 1 is added to the score if there should be an occurrence of positive emoji and 1 is subtracted if there should arise an occurrence of negative emoji).
- General score from online extremity vocabulary MPQA (the nearness of solid positive word in the tweet builds the score by 1 and the nearness of powerless negative word would diminish the score by half).
- Naïve Bayes algorithm plays a vital role in the calculation of the models.
- All the emojis in a tweet.

- All the positive emoticons in the tweet
- All the negative emoticons in the tweet
- All the positive words taken from MPQA lexicon in the tweet
- How many MPQA corpus negative words does the tweet have.
- How many normal forms of the verbs are there in the tweet.
- How many past forms of the verbs are there in the tweet.
- How many present forms of the verbs are there in the tweet.
- How many past participle forms of the verbs are there in the tweet.
- How many 3rd person singular present forms of the verbs are there in the tweet.
- How many non 3rd person singular present forms of the verbs are there in the tweet.
- How many plural forms of the nouns are there in the tweet.
- How many singular forms of the nouns are there in the tweet.
- How many cardinal numbers are there in the tweet.
- How many prepositions or coordinating conjunctions in a tweet.
- How many adverbs are there in the tweet.
- How many forms of all the verbs are there in the tweet.

After this, what we are going to do is, we will give numerical thinking of how we ascertain the models of unigram words utilizing the Naive Bayes. The essential idea is to figure the likelihood of a word having a place with any of the conceivable classes from our training set. Utilizing scientific formulae we will show a case of figuring likelihood of word having a place with objective and subjective class. Comparative advances should be taken for positive and negative classes too.

So let's begin by figuring the likelihood of a word in our preparation information for having a place with a specific class :

$$P(word_1|obj) = \frac{count(word_1\ in\ obj\ class)}{count(total\ words\ in\ obj)}$$

Here we express the Bayes' algorithm. As per the description, on the off chance that we have to discover the likelihood of whether some tweet is objective or not, we have to ascertain the likelihood of tweet given the target class and the earlier likelihood of target class.

*P(tweet)* could be calculated and replaced by *P(tweet | obj) + P(tweet | subj).*

$$P(obj|tweet) = \frac{P(tweet|obj).P(obj)}{P(tweet)}$$

Presently in the event that we accept autonomy of each unigram inside every tweet (which is the event of a word in every tweet won't influence the likelihood of event of some other word in the tweet ) we can estimate the likelihood of tweet considering the target class to a negligible result of the likelihood of the considerable number of words in the tweet having a place with target class. Besides, on the off chance that we expect a break even with class sizes for both target and subjective class we can disregard the earlier likelihood of the goal class. From now on we are left with the accompanying recipe, having two particular terms and all two are effortlessly ascertained through the equation specified before.

$$P(obj|tweet) = \frac{\prod_{i=1}^{N} [P(word_i|obj)}{\prod_{i=1}^{N} [P(word_i|obj) + \prod_{i=1}^{N} [P(word_i|subj)}$$

Since we have the likelihood of objectivity having a specific tweet, we can undoubtedly compute the likelihood of subjectivity taking that same tweet by just subtracting the prior term from 1. This is on the grounds that probabilities should dependably add to 1. So, we find the value of *P(subj | tweet)* with prior information of *P(obj | tweet)*.

$$P(subj|tweet) = 1 - P(obj|tweet)$$

At the final step, P(obj | tweet) will be calculated for each tweet. This term is used as the individual feature in the classification of objectivity / subjectivity.

We found two fundamental potential issues with this approach. To begin with being that on the off chance that we incorporate each one of a kind word utilized as a part of the informational collection then the rundown of words will be too substantial making the calculation excessively costly and tedious. To understand this we just incorporate words which have been utilized no less than 5itimesiiniour information. This decreases the span of our word reference for objective/ subjective grouping from 21,127 to 3,142. Whereas for positive/ negative order unigram lexicon measure is decreased from 7,500 to 2,130 words.

We calculate the likelihood of a word associated with some class using the formula given below:

$$\boldsymbol{P(word_1|obj) = \frac{count(word_1\ in\ obj\ class) +\ x}{count(total\ words\ in\ obj) +\ x(total\ unique\ words\ in\ obj}}$$

In the above discussed formula, the term 'x' is a consistent factor known as the smoothing factor, which is chosen to bei1 subjectively. The functioning is that regardless of whether the inclusion of a word of a specific class is zero, the numerator still has a little esteem so the likelihood of a word having a place with some classiwillineveribe equivalent to zero. Rather if the likelihood would have been zero as indicated by the before equation, it would be supplanted by a little non zero likelihood.

the investigation of an aggregate of 29 attributes for objectivity/ subjectivity arrangement and utilized WEKA for the computation of the data picked up from every one of these attributes. The following graph depicts the results:

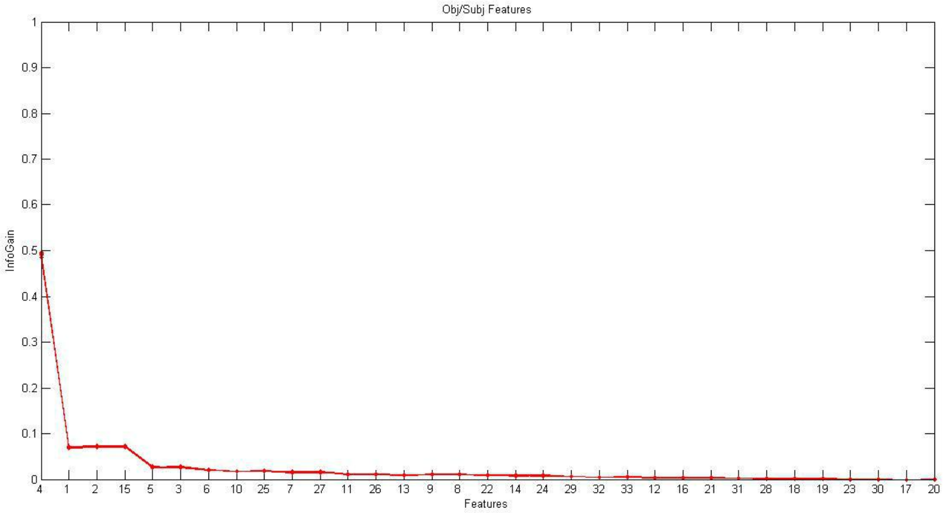


**Figure 4.1:** Knowledge attained by  Objectivity/ Subjectivity attributes

This plot is essentially the super-burden of 20 unique charts, everyone touched base through one overlap out of the 20-crease cross approval we performed. Since we see that every one of the plots are pleasantly covering so the outcomes each overlay is nearly similar which demonstrates to us that the highlights we select will perform best in every one of the situations. We chose the top attributes from this plot, which are as per the following:

1. Models of the words which are unigrams.
2. The tweet having some URL
3. The tweet having some Emojis
4. The frequency of all the personal pronouns.
5. The frequency of the marks of exclamation.

Correspondingly we investigated 20 attributes for  positive/ negative grouping and utilized WEKA for figuring the data picked up from every one of these attrubutes. The subsequent plot is demonstrated as follows:

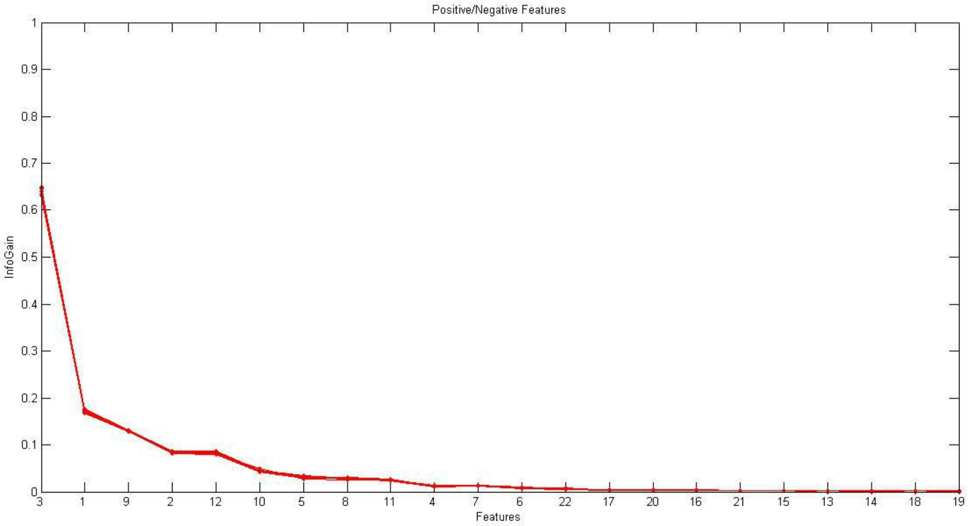


**Figure 4.2:** Knowledge attained from the attributes about Positivity/ Negativity

The above plot is essentially the super-inconvenience of 25 unique plots, everyone landed through one creases out of the 25-overlay cross approval we performed. Since we see that every one of the plots is pleasantly covering soithe outcomes each overlay are nearly a similar which demonstrates to us that the highlights we select will perform best in every one of the situations. We chose the best 10 includes out of which 5 were excess attributes and we were left with just 5 attributes for our positive/ negative characterization which are as per the following:

- Models of the words which are unigrams
- The sum total of the positive emojis
- The sum total of the negative emojis

There were tons of unessential features, that we need to ignore as they didn't provide any necessary information to us.

The following plot gives clear understanding of how this works. Here, we have plotted the actual text score in a 2-Dimentional environment:

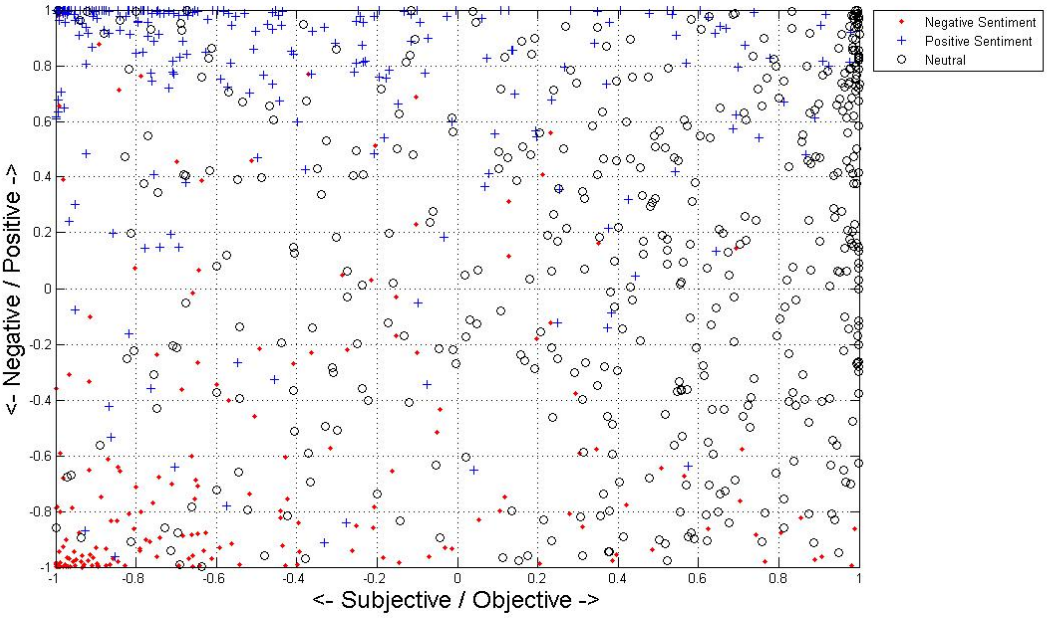


**Figure 4.3:** 2-D Scater Plot after Step 1

Here, the marks are the real ground truth and the conveyance indicates how the ordered information focuses are really scattered all through the contour. As we go right the tweet begins ending up progressively objective as well as we go up the tweet begins ending up more positive. The outcomes for our characterization approach are specified in the following segment of this report.

### 4.4 Application based on web for performing Tweet Analysis

We composed a web-based application, which carries out continuous opinion investigation on the tweets that coordinated specific catchphrases given by the client. For instance, ifia client is occupied with performing feeling examination on tweets that consist of the twitter handler '@PMModi' he/ she will enter that catchphrase and the web based application will play out the suitable assumption investigation and show the outcomes for the client.

Web based application has been created with the help of Shiny library of R in light of the fact that it can be utilized as a free web facilitating administration andiit gives ailayeriof deliberation to the designer from the low level web activities so it is less demanding to learn. We actualized our program in R programming language. We utilized the Google Visualization Chart API for exhibiting our outcomes in a graphical, straightforward way.

### 4.5 Methodology

As per our approach, we have implemented the task in two different ways:

1. Using the primary memory of the system.
2. With the help of Big Data ecosystem.

### 4.5.1 Using the primary memory of the system

This is the simplest way of performing the sentiment analysis with the help of some basic knowledge of programming and some algorithms.

Here, what we did was:

1. Created a web based application using shiny library in R.
2. Created the Twitter streaming API from apps.twitter.com
3. With the help of a few lines of the code, we downloaded a set of tweets related to a particular keyword.
4. Corpus of positives, negatives and neutral words were downloaded from the UCL repository.
5. Now the data downloaded in step 3 were saved into a csv file.
6. This csv file was then treated with basic steps of NLP.
7. This lead to the creation of final csv file.
8. The rows in this csv file were then compared to the corpus and the score was calculated.
9. The score based graphical representation was done with the help of ggplot2 library.

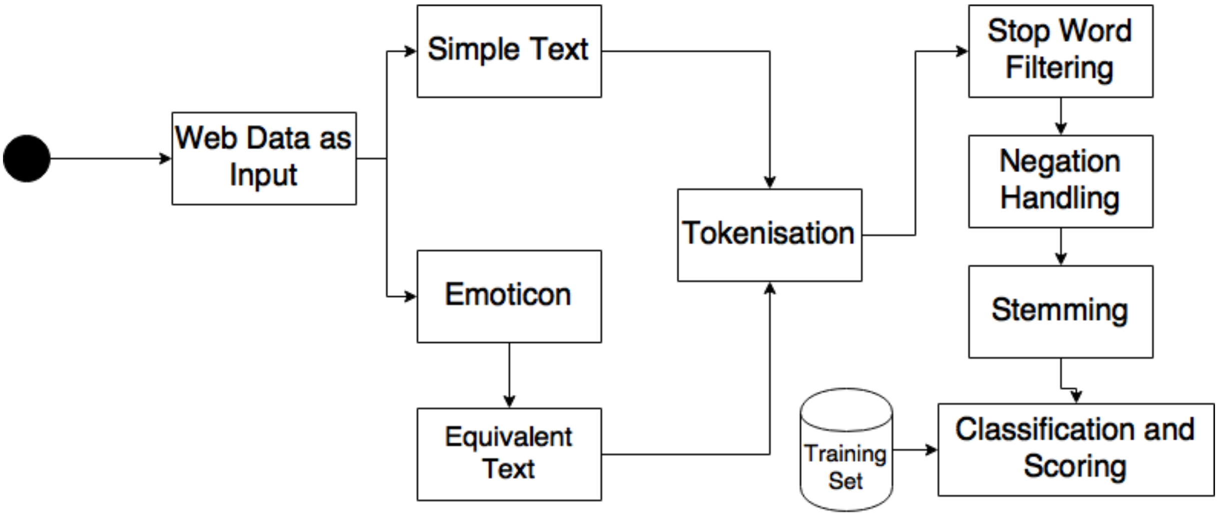


**Figure 4.4:** Methodology without Big Data

### 4.5.2 With the help of Big Data ecosystem

The method to defeat the issue that we are looking in the current issue that is demonstrated obviously in the Hadoop framework. Along these lines, to accomplish this we will take after the accompanying techniques:

1. To create the twitter API on apps.twitter.com.
2. Fetch data with the help of Apache flume.
3. Take outcomes using the noSQL or HQL (Hive Query Language).

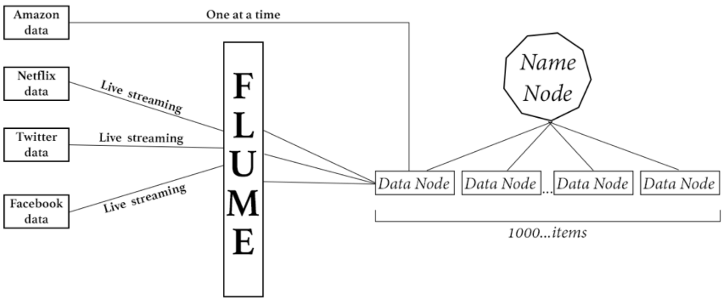
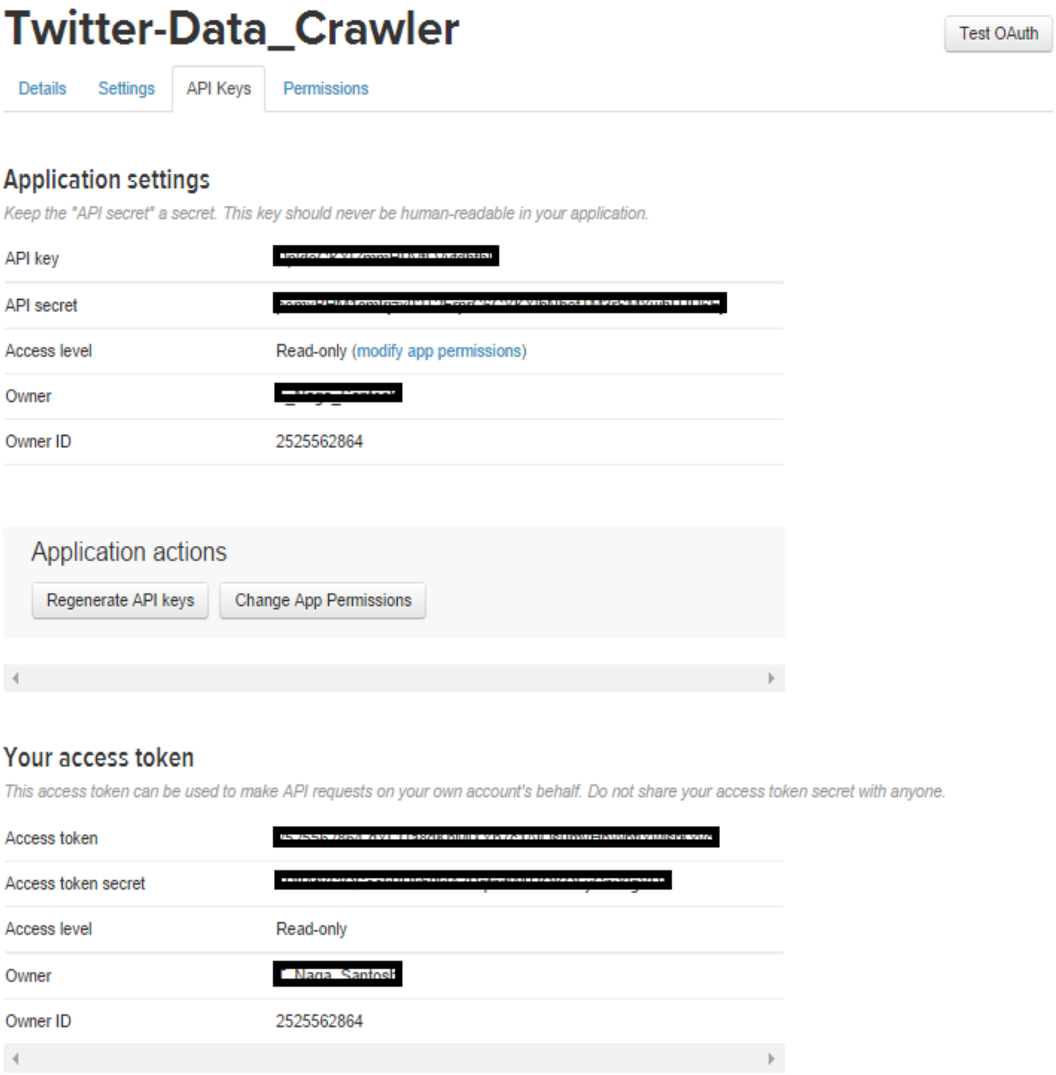


**Figure 4.5:** Methodology with Big Data

#### 4.5.2.1 To create the twitter API on apps.twitter.com



Application settings

Keep the "API secret" a secret. This key should never be human-readable in your application.

| | |
|---|---|
| API key | [redacted] |
| API secret | [redacted] |
| Access level | Read-only (modify app permissions) |
| Owner | [redacted] |
| Owner ID | 2525562864 |

Application actions

Your access token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

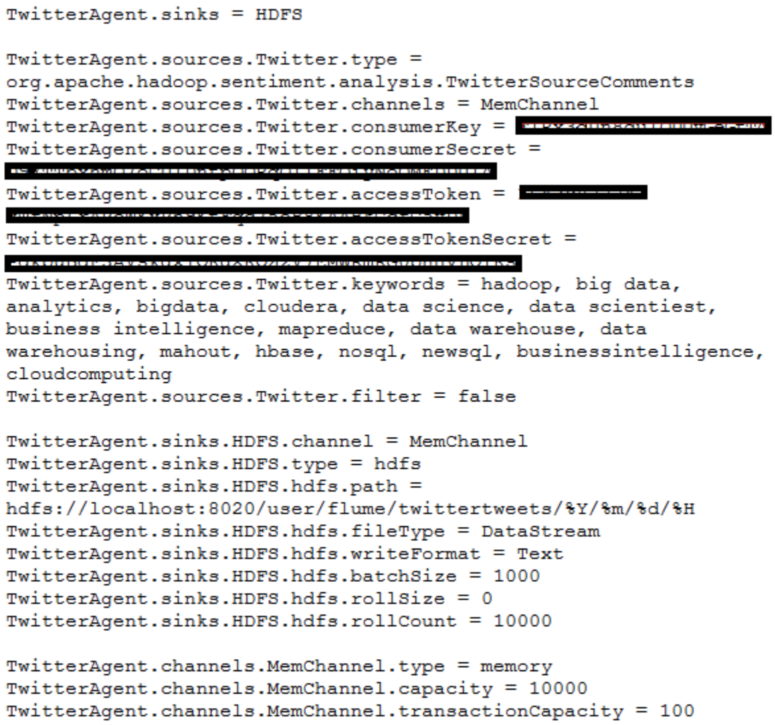
| | |
|---|---|
| Access token | [redacted] |
| Access token secret | [redacted] |
| Access level | Read-only |
| Owner | [redacted] |
| Owner ID | 2525562864 |

**Figure 4.6**: Creating Twitter application from Twitter Developer

Above image demonstrates obviously the application keys thatiare produced in the wake of making application and in this set of keys, we can see the upper two keys are the API key & API secret. Going for reaming two keys, it is only known as the access tokens that we need to produce it without anyone else' interference by tapping the create get to token. Subsequent to clicking that we can get the two keys that are our record get to token and going to that one is Access token and the other one is the Access token secret.

#### 4.5.2.2 Fetch data with the help of Apache flume

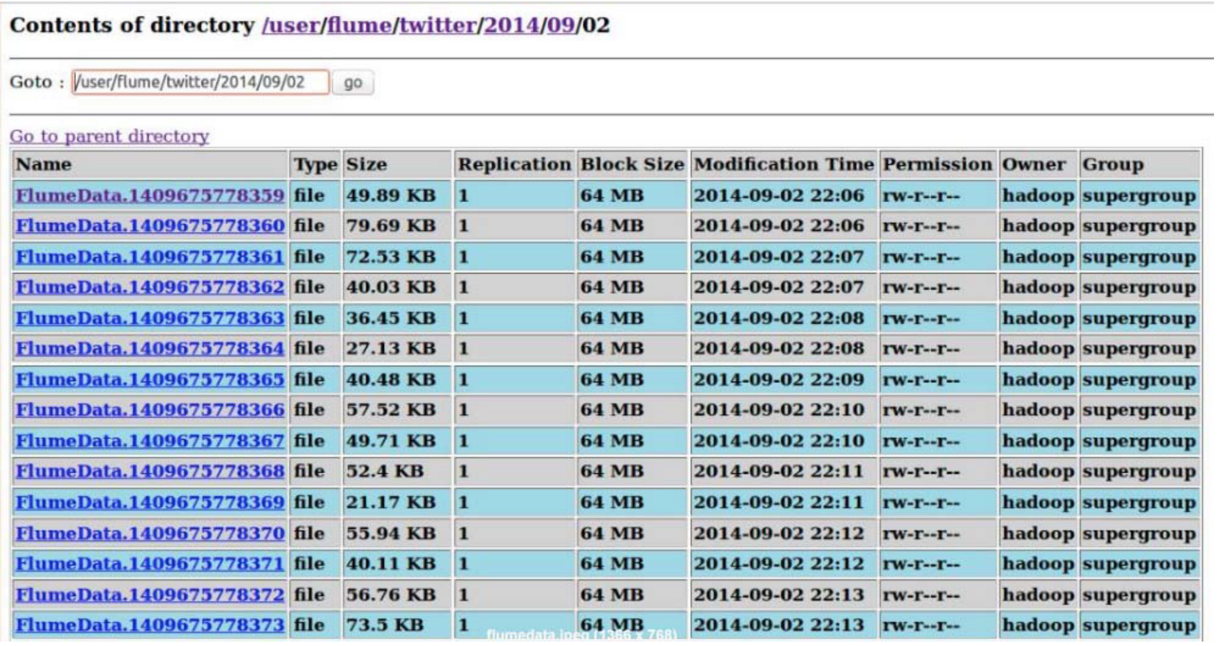
In the wake of making an application on apps.twitter.com, we need to utilize the consumer key and secret alongside the access token and secret esteems. The Twitter database will be accessed through these and the JSON configuration of the dataset can be downloaded from it's server and could be transferred to the HDFS. The accompanying is the design document that we need to use for getting the Twitter information from Twitter.



```
TwitterAgent.sinks = HDFS

TwitterAgent.sources.Twitter.type =
org.apache.hadoop.sentiment.analysis.TwitterSourceComments
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey = [redacted]
TwitterAgent.sources.Twitter.consumerSecret =
[redacted]
TwitterAgent.sources.Twitter.accessToken = [redacted]
TwitterAgent.sources.Twitter.accessTokenSecret =
[redacted]
TwitterAgent.sources.Twitter.keywords = hadoop, big data,
analytics, bigdata, cloudera, data science, data scientiest,
business intelligence, mapreduce, data warehouse, data
warehousing, mahout, hbase, nosql, newsql, businessintelligence,
cloudcomputing
TwitterAgent.sources.Twitter.filter = false

TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.hdfs.path =
hdfs://localhost:8020/user/flume/twittertweets/%Y/%m/%d/%H
TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream
TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text
TwitterAgent.sinks.HDFS.hdfs.batchSize = 1000
TwitterAgent.sinks.HDFS.hdfs.rollSize = 0
TwitterAgent.sinks.HDFS.hdfs.rollCount = 10000

TwitterAgent.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 10000
TwitterAgent.channels.MemChannel.transactionCapacity = 100
```

**Figure 4.7**: The changes to be made in the configuration file of flume

#### 4.5.2.3 Take outcomes using the noSQL or HQL (Hive Query Language)



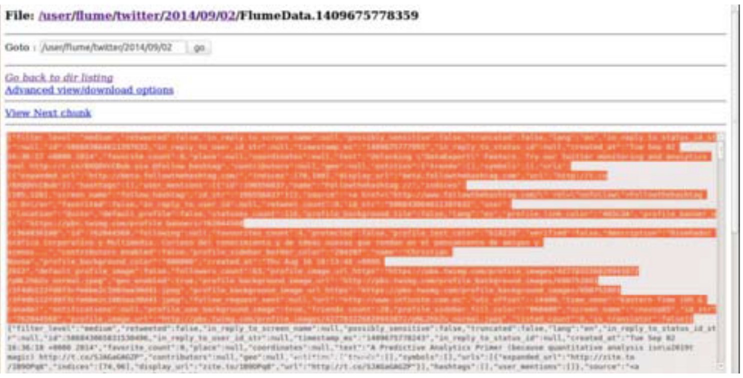
**Contents of directory /user/flume/twitter/2014/09/02**

Goto : /user/flume/twitter/2014/09/02 go

Go to parent directory

| Name | Type | Size | Replication | Block Size | Modification Time | Permission | Owner | Group |
|---|---|---|---|---|---|---|---|---|
| FlumeData.1409675778359 | file | 49.89 KB | 1 | 64 MB | 2014-09-02 22:06 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778360 | file | 79.69 KB | 1 | 64 MB | 2014-09-02 22:06 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778361 | file | 72.53 KB | 1 | 64 MB | 2014-09-02 22:07 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778362 | file | 40.03 KB | 1 | 64 MB | 2014-09-02 22:07 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778363 | file | 36.45 KB | 1 | 64 MB | 2014-09-02 22:08 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778364 | file | 27.13 KB | 1 | 64 MB | 2014-09-02 22:08 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778365 | file | 40.48 KB | 1 | 64 MB | 2014-09-02 22:09 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778366 | file | 57.52 KB | 1 | 64 MB | 2014-09-02 22:10 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778367 | file | 49.71 KB | 1 | 64 MB | 2014-09-02 22:10 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778368 | file | 52.4 KB | 1 | 64 MB | 2014-09-02 22:11 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778369 | file | 21.17 KB | 1 | 64 MB | 2014-09-02 22:11 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778370 | file | 55.94 KB | 1 | 64 MB | 2014-09-02 22:12 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778371 | file | 40.11 KB | 1 | 64 MB | 2014-09-02 22:12 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778372 | file | 56.76 KB | 1 | 64 MB | 2014-09-02 22:13 | rw-r--r-- | hadoop | supergroup |
| FlumeData.1409675778373 | file | 73.5 KB | 1 | 64 MB | 2014-09-02 22:13 | rw-r--r-- | hadoop | supergroup |

**Figure 4.8**: A glimpse of Hadoop having twitter data



**File: /user/flume/twitter/2014/09/02/FlumeData.1409675778359**

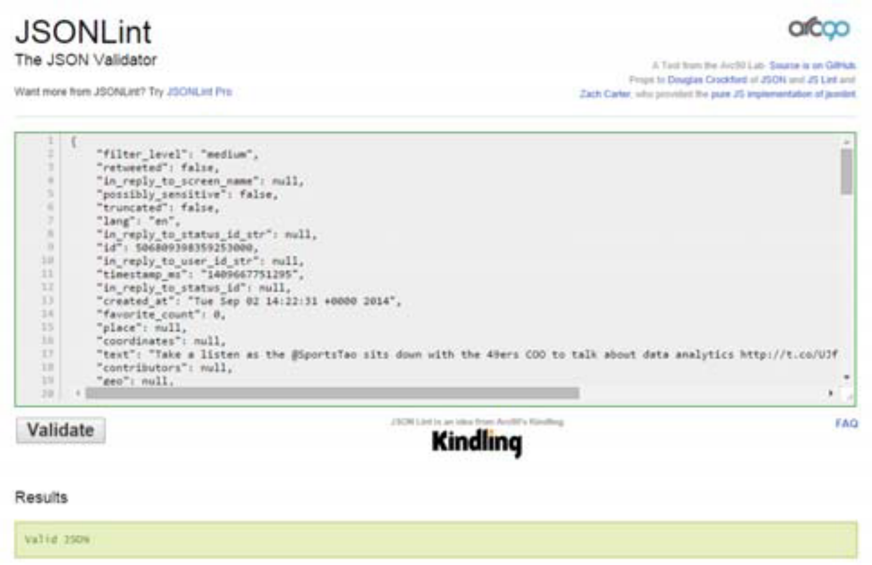
Goto : /user/flume/twitter/2014/09/02 go

Go back to dir listing
Advanced view/download options

View Next chunk

{"filter_level":"medium","retweeted":false,"in_reply_to_screen_name":null,"possibly_sensitive":false,"truncated":false,"lang":"en","in_reply_to_status_id_st [illegible]
{"filter_level":"medium","retweeted":false,"in_reply_to_screen_name":null,"possibly_sensitive":false,"truncated":false,"lang":"en","in_reply_to_status_id_st
r":null,"id":506843065831530496,"in_reply_to_user_id_str":null,"timestamp_ms":"1409675778243","in_reply_to_status_id":null,"created_at":"Tue Sep 02
16:36:18 +0000 2014","favorite_count":0,"place":null,"coordinates":null,"text":"A Predictive Analytics Primer (because quantitative analysis isn\u2019t
magic) http://t.co/SJ6GaGAG2P","contributors":null,"geo":null,"entities":{"trends":[],"symbols":[],"urls":[{"expanded_url":"http://zite.to
/1890PqX","indices":[74,96],"display_url":"zite.to/1890PqX","url":"http://t.co/SJ6GaGAG2P"}],"hashtags":[],"user_mentions":[]},"source":"<a

**Figure 4.9**: JSON format of twitter data


JSONLint
The JSON Validator
Want more from JSONLint? Try JSONLint Pro
A Tool from the Arc90 Lab · Source is on GitHub
Props to Douglas Crockford of JSON and JS Lint and
Zach Carter, who provided the pure JS implementation of jsonlint
Validate
JSON Lint is an idea from Arc90's Kindling
Kindling
FAQ
Results

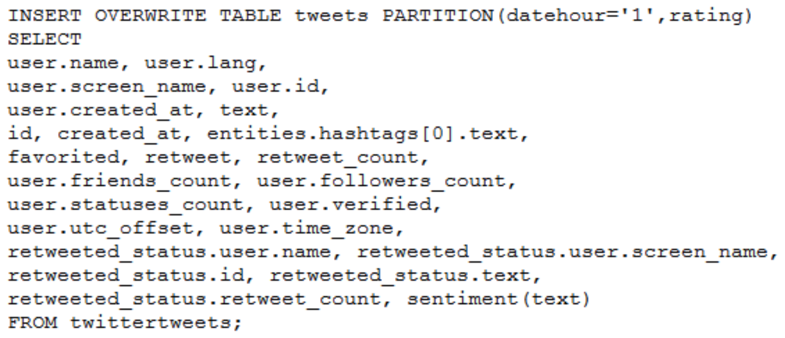
```
{
    "filter_level": "medium",
    "retweeted": false,
    "in_reply_to_screen_name": null,
    "possibly_sensitive": false,
    "truncated": false,
    "lang": "en",
    "in_reply_to_status_id_str": null,
    "id": 506809398359253000,
    "in_reply_to_user_id_str": null,
    "timestamp_ms": "1409667751295",
    "in_reply_to_status_id": null,
    "created_at": "Tue Sep 02 14:22:31 +0000 2014",
    "favorite_count": 0,
    "place": null,
    "coordinates": null,
    "text": "Take a listen as the @SportsTao sits down with the 49ers COO to talk about data analytics http://t.co/UJf
    "contributors": null,
    "geo": null,
```

```
Valid JSON
```

**Figure 4.10**: HQL validation of data

```
CREATE EXTERNAL TABLE tweets (
 username STRING,
 lang STRING,
 screen_name STRING,
 id BIGINT,
 created_at STRING,
 text STRING,
 post_id BIGINT,
 post_created_at STRING,
 hashtags STRING,
 retweet BOOLEAN,
 favorited BOOLEAN,
 retweet_count BIGINT,
 friends_count INT,
 followers_count INT,
 statuses_count INT,
 verified BOOLEAN,
 utc_offset INT,
 time_zone STRING,
 retweeted_username STRING,
 retweeted_screen_name STRING,
 retweeted_id BIGINT,
 retweeted_text STRING,
 retweeted_retweet_count BIGINT
)
PARTITIONED BY (datehour INT, rating STRING)
LOCATION '/user/flume/tweets';
```

**Figure 4.11**: Table creation using HQL queries



```
INSERT OVERWRITE TABLE tweets PARTITION(datehour='1',rating)
SELECT
user.name, user.lang,
user.screen_name, user.id,
user.created_at, text,
id, created_at, entities.hashtags[0].text,
favorited, retweet, retweet_count,
user.friends_count, user.followers_count,
user.statuses_count, user.verified,
user.utc_offset, user.time_zone,
retweeted_status.user.name, retweeted_status.user.screen_name,
retweeted_status.id, retweeted_status.text,
retweeted_status.retweet_count, sentiment(text)
FROM twittertweets;
```

**Figure 4.12**: Inserting data by performing sentiment analysis

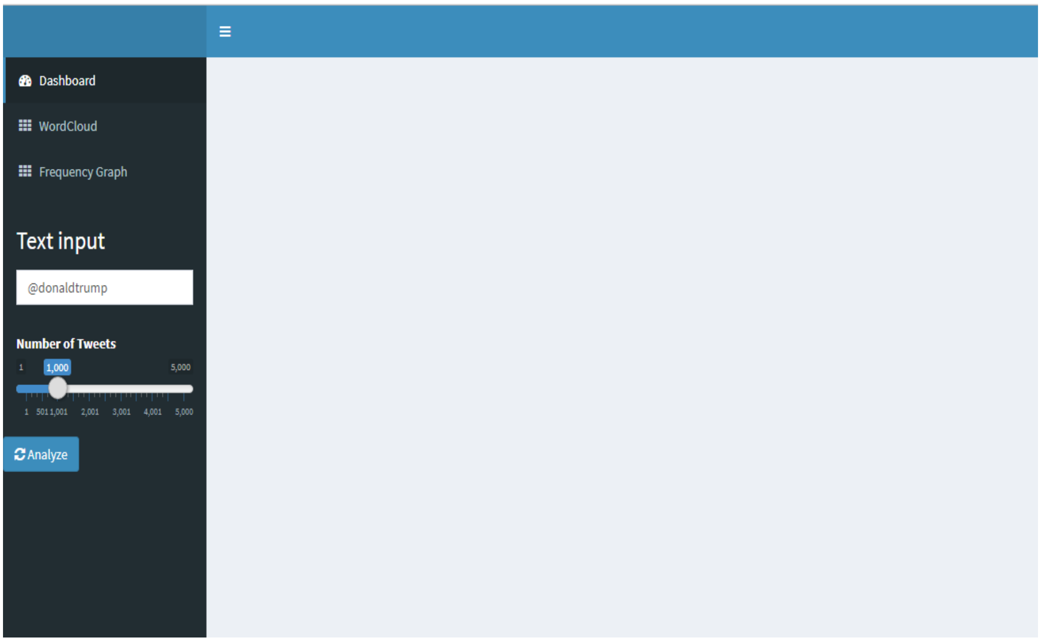
Additionally we are utilizing another UDF's (User Defined Functions) for playing out the opinion mining on the stories that are made by utilizing Hive.[8] From that we can play out the opinion mining. Furthermore, gain the outcomes where another table is made by segment idea to such an extent that every one of the remarks that are having positive will go into the positive parcel and every one of the remarks that are having moderate will go into direct segment lastly every one of the remarks that are having negative will go into negative segment.

*Chapter 5*

# RESULT

The reason for this venture is to manufacture a calculation that can precisely arrange Twitter messages as positive or negative, as for an inquiry term. Our theory is that we can acquire high exactness on arranging slant in Twitter messages utilizing machine learning procedures.

Below is the screenshot of the dashboard of our project that takes input of the twitter handler (like #apple) or user_id like (like @donaldtrump) from the user and analyze the related tweets.



Text input

@donaldtrump

Number of Tweets

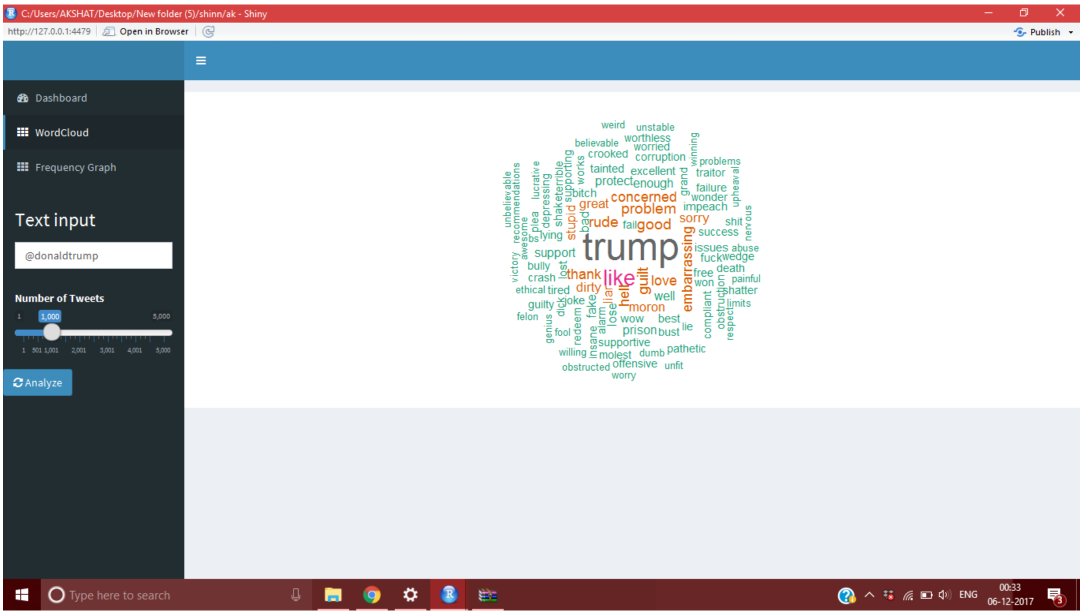
1 1,000 5,000

1 501 1,001 2,001 3,001 4,001 5,000

**Figure 5.1:** Dashboard showing home screen

Dashboard has two tabs:

- **WordCloud:** it shows words that are used more frequently by people to describe their sentiments (in this example it's for @donaldtrump)**.**



Text input

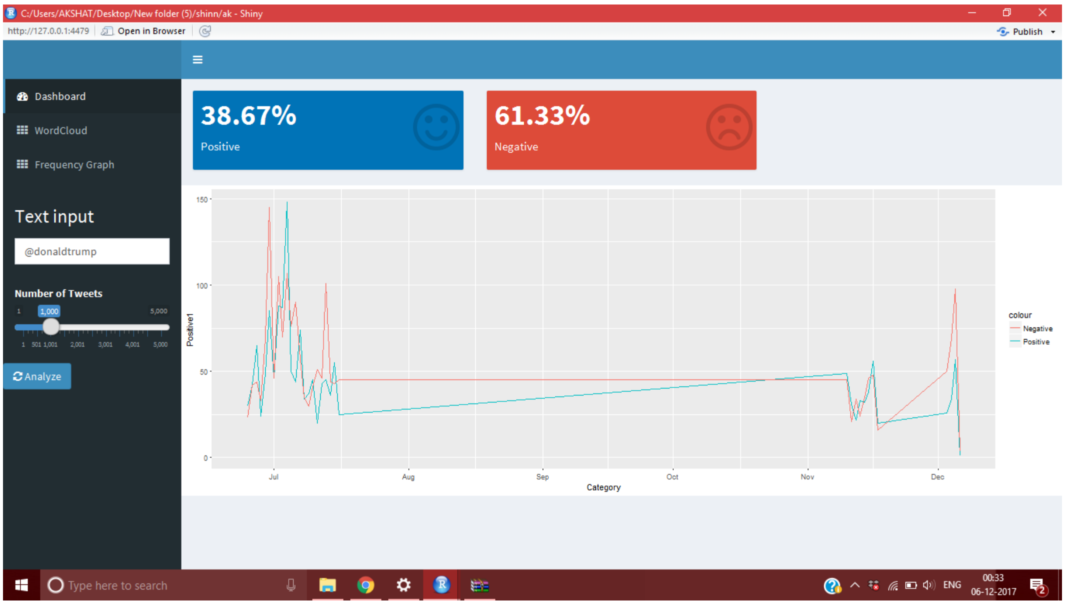
@donaldtrump

Number of Tweets

1,000

**Figure 5.2:** Word Cloud of the tweet's content

- **Frequency Graph:** it shows the frequency of word used and number of positive/negative tweets/words. Thus showing people's sentiments.



Text input

@donaldtrump

Number of Tweets

1,000

38.67%
Positive

61.33%
Negative

**Figure 5.3:** Frequency graph of the tweet's content

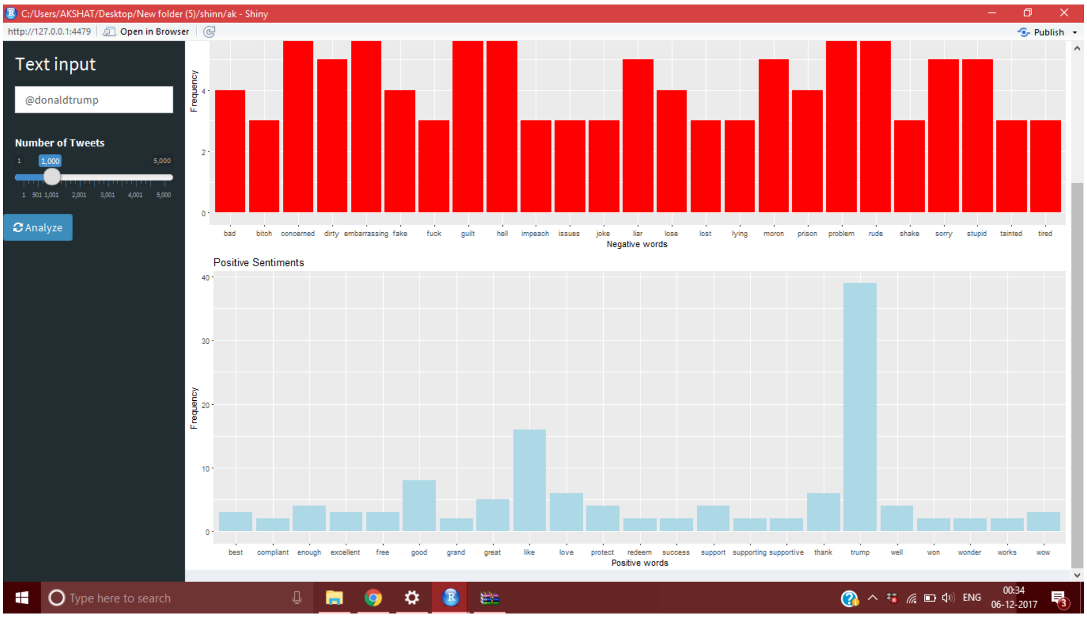
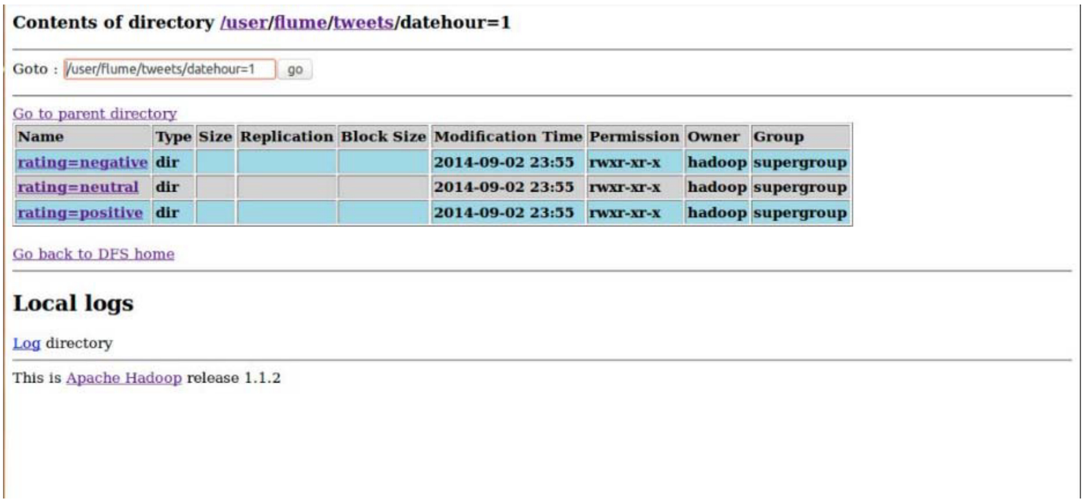


**Figure 5.4:** Bar plot of the tweet's content



## Contents of directory /user/flume/tweets/datehour=1

Goto : /user/flume/tweets/datehour=1 go

Go to parent directory

| Name | Type | Size | Replication | Block Size | Modification Time | Permission | Owner | Group |
|---|---|---|---|---|---|---|---|---|
| rating=negative | dir | | | | 2014-09-02 23:55 | rwxr-xr-x | hadoop | supergroup |
| rating=neutral | dir | | | | 2014-09-02 23:55 | rwxr-xr-x | hadoop | supergroup |
| rating=positive | dir | | | | 2014-09-02 23:55 | rwxr-xr-x | hadoop | supergroup |

Go back to DFS home

## Local logs

Log directory

This is Apache Hadoop release 1.1.2

**Figure 5.5:** The final result using flume

*Chapter 6*

# CONCLUSION AND FUTURE RECOMMENDATIONS

## 6.1 CONCLUSION

The opinion mining errand, is still in a developing stage, particularly in the field of miniaturized scale blogging it is exceptionally a long way from the finish. Along these lines, here I propose a few thoughts which I feel merit being investigated in future and may bring about additionally enhanced exactness.

At this moment I have worked just with extremely various unigram models. The enhancement of such models by including additional data like the closeness of the word with the invalidation term. We could indicate some sort of a window before the word (a window could be of like somewhere in the range of 3 or 5 words) under some thought and the impact of refutation might be joined into the model on the off chance that it exists in that window. The nearer the refutation word is to the unigram word whose earlier extremity is to be ascertained, the more it should influence the extremity. For instance, if the invalidation is ideal alongside the word, it might just switch the extremity of that word and more distant the nullification is from the word the more limited uncertainties impact ought to be.

## 6.2 EVALUATION METRICS

So, this is how we will be evaluating our experimental results with the help of following Information Retrieval matrices.

- Precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{TP+FN}$
- F-measure = $\frac{2*Precision*Recall}{Precision+Recall}$
- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

## 6.3 FUTURE RECOMMENDATIONS

### 6.3.1 Multiple classification

Till here, I have only worked on binary classification of twitter data, either as positive or negative. There are a number of tweets, for instance, those with URL's which do not have any sentiment, with pictures, GIFs or emojis or are neutral. These tweets are posted just to spread some useful information to the public, and not necessarily for giving an opinion. As a part of the future work, I would like to make classification into various levels of sentiment such as Non positive, positive, neutral, negative and Non negative.

### 6.3.2 Increased number of numeric features

The number of numeric features that I used in this project include number of negative and positive words, emojis, length of twitter posts and number of special characters such as twitter handles, hashtags and so on. These numeric features didn't yield good accuracy which was around 63 percent. So, as a part of the future work on this, I would like to bring about more as well as better numeric features.

### 6.3.3 More classifiers

In this project, I used Naïve Bayes Classifier, SVM (Support Vector Machine) Classifiers and MaxEnt Classifier extensively. So, now I would like to explore other algorithms like Artificial Neural networks, Convolution Neural networks, Deep Learning. Also generation of more number of numeric features (as discussed before) will make me to use more number of binary classifiers such as logistic regression and others.

### 6.3.4 Use Hadoop framework for Big Data

As I have observed, using more and more data improves the accuracy. But, instead of using Amazon EC2, the data still turned to be out of memory and CPU intensive. As a result of that, I was unable to run the SVM classifier and Naïve Bayes classifier along with POS Tagging. Not only this, but also I was unable to run MaxEnt algorithms on it. As a part of future work, I would like to make use of Hadoop frameworks like Apache flume for processing large data of this kind.

# REFERENCES

[1] T. Wilson, J. Wiebe and P. Hoffmann, “Recognizing contextual polarity in phrase-level sentiment analysis,” in Proceedings of HLT and EMNLP. ACL, **(2005)**, pp. 347–354

[2] C. C. Tao, S. K. Kim, Y. A. Lin, Y. Y. Yu, G. Bradski, A. Y. Ng and Kunle Olukotun, “Map-reduce for machine learning on multicore”, In NIPS, vol. 6, **(2006)**, pp. 281-288.

[3] L. Jimmy, and A. Kolcz, “Large-scale machine learning at twitter”, In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, ACM, **(2012)**, pp. 793-804.

[4] B. Jiang, U. Topaloglu and F. Yu, “Towards large-scale twitter mining for drug-related adverse events”, In Proceedings of the 2012 international workshop on Smart health and wellbeing, ACM, **(2012)**, pp. 25-32.

[5] L. Bingwei, E. Blasch, Y. Chen, D. Shen and G. Chen, “Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier”, In Big Data, 2013 IEEE International Conference on, IEEE, **(2013)**, pp. 99-104.

[6] Á. Cuesta, David F. and María D. R-Moreno, “A Framework for Massive Twitter Data Extraction and Analysis”, In Malaysian Journal of Computer Science, **(2014)**, pp. 50-67.

[7] S. Michal and A. Romanowski, “Sentiment analysis of Twitter data within big data distributed environment for stock prediction”, In Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on, IEEE, **(2015)**, pp. 1349-1354.

[8] T. Mohit, I. Gohokar, J. Sable, D. Paratwar and R. Wajgi, “Multi-Class Tweet Categorization Using Map Reduce Paradigm”, In International Journal of Computer Trends and Technology. **(2014)**, pp. 78-81.

[9] D. Jeffrey and S. Ghemawat, “MapReduce: simplified data processing on large clusters”, Communications of the ACM 51.1, **(2008)**, pp. 107-113.

[10] B. Yingyi, “HaLoop: Efficient iterative data processing on large clusters”, Proceedings of the VLDB Endowment 3.1-2, **(2010)**, pp. 285-296.

[11] T. Maite, “Lexicon-based methods for sentiment analysis”, Computational linguistics 37.2, **(2011)**, pp. 267-307.

[12] R. Tushar and S. Srivastava, “Analyzing stock market movements using twitter

sentiment analysis”, Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012). IEEE Computer Society, **(2012)**.

[13] D. Pessemier and Martens “MovieTweetings: A Movie Rating Dataset Collected From Twitter”, Ghent University, Ghent, Belgium, **(2013)**.

[14] Twitter. Twitter Search API, available at https://dev.twitter.com/rest/public/search.

[15] V. D. Katkar, S. V. Kulkarni, “A Novel Parallel implementation of Naive Bayesian classifier for Big Data”, International Conference on Green Computing, Communication and Conservation of Energy, 978-1-4673-6126-2/2013 IEEE, pp. 847-852.

[16] S. Kumar, F. Morstatter and H. Liu, “Twitter Data Analytics”, Springer Science & Business Media, **(2013)**.

[17] B. Vishal, “Data Mining in Dynamic Social Networks and Fuzzy Systems”, IGI Global, **(2013)**.

[18] G. Elmer, G. Langlois and J. Redden, “Compromised Data: From Social Media to Big Data”, Bloomsbury Publishing USA, **(2015)**.

[19] Nalini K. and L. J. Sheela, “Classification of Tweets Using Text Classifier to Detect Cyber Bullying”, In Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI, Springer International Publishing, vol. 2, **(2015)**, pp. 637-645.

[20] Jaba S. L. and Dr V. Shanthi, “An Approach for Discretization and Feature Selection Of Continuous-Valued Attributes in Medical Images for Classification Learning”, International Journal of Computer Theory and Engineering, vol. 1, no. 2, pp. 154.

[21] T. White, “Hadoop: The Definitive Guide”, Third Edition, O'Reilley, **(2012)**.

[22] L. George, “HBase: The Definitive Guide”, O'Reilley, **(2011)**.

[23] E. Hewitt, “Cassandra: The Definitive Guide”, O'Reilley, **(2010)**.

[24] A. Gates, “Programming Pig”, O'Reilley, **(2011)**.

# JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
## LEARNING RESOURCE CENTER

## PLAGIARISM VERIFICATION REPORT

Date: 11/05/2018

Type of Document (Tick): Thesis | M.Tech Dissertation/ Report | [✓ B.Tech Project Report] | Paper

Name: PRATEEK GOEL Department: CSE

Enrolment No. 141236 Registration No. ______

Phone No. 9736648042 Email ID. prateekgoyal29@gmail.com

Name of the Supervisor: DR. HEMRAJ SAINI

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):

SENTIMENT ANALYSIS OF TWEETS AND NEWS USING BIG DATA

Kindly allow me to avail Turnitin software report for the document mentioned above.

(Signature)

**FOR ACCOUNTS DEPARTMENT:**

Amount deposited: Rs. 300/- Dated: 11/5/18 Receipt No. BR1805 326

(Enclosed payment slip)

(Account Officer)

**FOR LRC USE:**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

| Copy Received on | Report delivered on | Similarity Index in % | Submission Details | |
|---|---|---|---|---|
| 12/05/2018 | 14/05/2018 | 8% | Word Counts | 10,595 |
| | | | Character Counts | 55,390 |
| | | | Page counts | 53 |
| | | | File Size | 2.91M |

Checked by
Name & Signature

Librarian

## "Sentiment Analysis of Tweets and News using Big Data"

ORIGINALITY REPORT

| | | | |
|---|---|---|---|
| 8% | 7% | 1% | 3% |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.sersc.org<br>Internet Source | 3% |
| 2 | arxiv.org<br>Internet Source | 2% |
| 3 | Submitted to University of Johannsburg<br>Student Paper | 1% |
| 4 | Submitted to Malaviya National Institute of Technology<br>Student Paper | <1% |
| 5 | grietinfo.in<br>Internet Source | <1% |
| 6 | issuu.com<br>Internet Source | <1% |
| 7 | aclweb.org<br>Internet Source | <1% |
| 8 | Submitted to Savitribai Phule Pune University<br>Student Paper | <1% |

Chauhan