CLASSIFICATION OF

DATA

Nitesh Singh (101203067) and Sarah Afrin(101213048)

DR.PARTEEK BHATIA

THAPAR UNIVERSITY

Contents

Automatic Tweets Classification Using Mahout and Hadoop

Nitesh Singh[1]
3rd year BE Computer Science
Thapar University
Patiala, Punjab-147004

Sarah Afrin[2]
3rd year BE Computer Science
Thapar University
Patiala, Punjab-147004

Course: UMA061 Data Warehouse and Mining
Mentor: Dr. Parteek Bhatia

1 ABSTRACT

With the ever increasing amount of data in today's world it has become important to segregate and analyze useful information from rest of the data, But due to the extremely high rate of generation of data the size of data becomes larger enough, thus for handling this big data we need the power of Cloud Computing. Cloud environment provides the necessary computing and storage resources. Sentiment analysis or opinion mining has gained importance in the past few years as an application of big data and Cloud Computing. Microblogging sites have become the most popular method of communication and expression of views for the internet users for example 50 million tweets are generated each day and thus generate a larger amount of data, which makes them rich sources for sentiment analysis. The purpose of this paper is to find out feedback by the people or trends in the social networking sites like Twitter. We have introduced a novel approach for automatically classifying the sentiments of public tweets posted on Twitter. The tweets have been grouped with respect to a query term. Further they have been classified into different classes based on the tweet text. Then data has been analyzed to find trending hash tagged words separately for positive, negative and neutral tweets. Finally the experimental results have been validated on real Cloud environment. The model is first trained to classify tweets into different categories. Then the new tweets are tested on this model.

2 INTRODUCTION

Today, a large amount of people express themselves on the Web in numerous ways through platforms like blogs and social networks. The number of such people has escalated in the recent years. Twitter is a perfect example of such websites. There are 500 million tweets per day and around 200 billion tweets per year[1] .Twitter allows users to publish small updates (140 character limit) to their profiles. Encouraged by the growth of Twitter, people are seeking ways to mine Twitter for information about what people think and feel about their products and services. This opinion mining has become very important particularly for companies who want to know their customer's feedback. There are several companies who deliver Twitter sentiment analysis as a service. A few examples could be Twitratr, tweet feel and Sentiment140 give sentiments for products/brands.

Sentiment analysis is examination of data in order to determine response of a group of people towards a subject. The data to be analyzed could be blogs, discussions, comments or similar content. By sentiment analysis, sometimes also called opinion mining, we can deduce the overall response of the users. Generally this data is huge enough to be called 'big data'. Sentiment analysis can be applied not only to twitter but it could also be used to find trends in personal blogs, product /movie reviews or to forecast market movements by analyzing news and blogs.

The content of a tweet includes several slangs, punctuation mistakes, abbreviations and poor spelling. All of these have to be taken care of while processing data. Also, only 60% of users of twitter actually tweet on a regular basis.[1] These are a few challenges faced while analyzing data. Others include

- Name entity recognition: what does the noun actually refer to. Eg: galaxy could refer to a chocolate brand or a clothing brand or the word galaxy itself
- Sarcasm: a computer program can never understand sarcasm and thus may give a wrong analysis
- Anaphora resolution: for which noun is the pronoun used.
- Parsing: identifying the subject and object. Also identifying the subject for which a verb is used

This research presents the result of applying two different approaches to the problem of automatic sentiment classification of tweets. The first approach is using opinion lexicons .The second approach is we investigate the utility of linguistic features for detecting the sentiment of Twitter messages. We evaluate the usefulness of existing lexical resources as well as features that capture information about the informal and creative language used in microblogging. We take a supervised approach to the problem, but leverage existing hashtags in the Twitter data for building training data.

3 Related Work

Sentiment analysis is a widely studied and researched topic. There have been many research papers written on sentiment analysis for various fields like blogs, movie review, Product Review. (Pang and Lee 2008)[5] survey gives various methodology to directly enable opinion-oriented information seeking systems. Researchers have also analyzed the brand impact of microblogging (Jansen).
Sentiment Classification For Product Reviews (Mahmoud Mohamed Hassan Mostafa Mohamed Ameen Mohamed Abdelkader Hamed Mai Mohamed Mahmoud ,2013) also analyses reviews for a specified product.
There were not many papers which used Cloud based technology (Hive and Hadoop) to do sentiment analysis of twitter data. (Pang and Lee 2004) present a system in which first tweets are identified to have a sentiment then tey are classified as positive or negative. In (Yang et al., 2007) shows the use web-blogs to demonstrate a model for sentiment analysis and use emotion icons assigned to blog posts as indicators of users' mood. This is very relevant to Twitter because many users have emoticons in their tweets. Large-Scale Sentiment Analysis for News and Blogs(Namrata Godbole, Manjunath Srinivasaia, Steven Skiena ,) assigns scores indicating positive or negative opinion to each distinct entity in the text corpus.
The reasons for our project to be different from the ones mentioned above are:
1. **Dataset**: The Data used is not downloaded or extracted from any existing data corpus but instead live data of twitter was collected using twitter4j (Twitter Api implementation for java)
2. **Available data**: Another difference is the sheer magnitude of data. In (Pang and Lee 2002), the corpus size 2053. With the Twitter API, it is much easier to collect millions of tweets for training.

3. **Language model:** Twitter users post messages from many different mediums, including their cell phones. The frequency of misspellings and slang in tweets is much higher than other domains. The tweets are also differentiated based on language.
4. **Location:** Apart from classifying tweets as positive , negative and neutral we have also classified tweets based on location and show which country shows what type of sentiment for a particular tweets, which country has how many tweets for a particular topic.
5. **Device Used:** We have also analyzed which device was used to make a tweet

4 PROPOSED FRAMEWORK

4.1 DATA GATHERING AND PRE PROCESSING

One of the most important task involved in doing Twitter sentiment analysis is gathering data because it the quality of data gathered that eventually converts into accuracy of results. Instead of using the existing old data for analysis we used twitter API to collect data about recent topics. By using the API we could easily get millions of tweets for our analysis.

We used twitter4j, open source package with implementation of twitter API for java. A java program was written for collecting the tweets by creating a twitter stream giving the Stream the query term as input and the response given by the stream ,after it has verified the user starting the stream , was parsed and a comma separated file(csv) was made with data as described above(Section 4).

The following Diagram explains the process in detail:

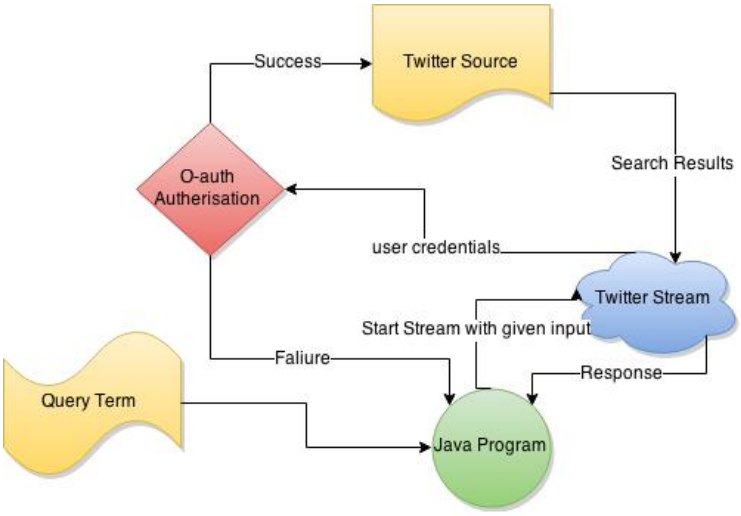


Figure 2Gathering data

Most of the data processing has already been done in the java program we made like removing any extra symbols , unnecessary new line also it was important to remove any extra delimiter which in our case was comma ,as we were using csv to separate different attributes so if there is a comma in attribute's value it can lead to error so it was also removed, Also a random sample data set from each input was manually checked for any error if found any the whole data set was re-processed to maintain accuracy.

Figure 1: Data gathering

Figure 3:Sample tweets

Data Description

The format of collected tweets has following fields:

- **ID**: The Unique Twitter id of the tweet
- **Created at** :The time tweet was created
- **Device**: The Derive used to tweet
- **Favorite count**: Indicates approximately how many times this Tweet has been "favorited" by Twitter users.
- **Retweet** :Text Written while retweeting
- **Retweet Name**: Name of user who retweeted the tweet.(twitter handle)
- **Retweet Screen Name**: Full name of user retweeting.
- **Screen Name**: Full name of user who has tweeted the current tweet.
- **Tweets** :Number of tweets the user has posted
- **Name** :Unique Twitter handle of the user
- **Friends Cou**nt: Number of People following the current user.
- **Follow Count**: Number of People Followed by the current user.
- **Verified**: This is either True or False depending on whether the user's Account is verified or not
- **UTC**: This tells the utc time of tweet
- **Time Zone**: This tells the time zone from which the tweet was made
- **Location**: The location that this tweet refers to if available.
- **Language**: This tells the language the tweet is made in.

4.2 MODEL FOR SENTIMENT ANALYSIS(USING LEXICON)

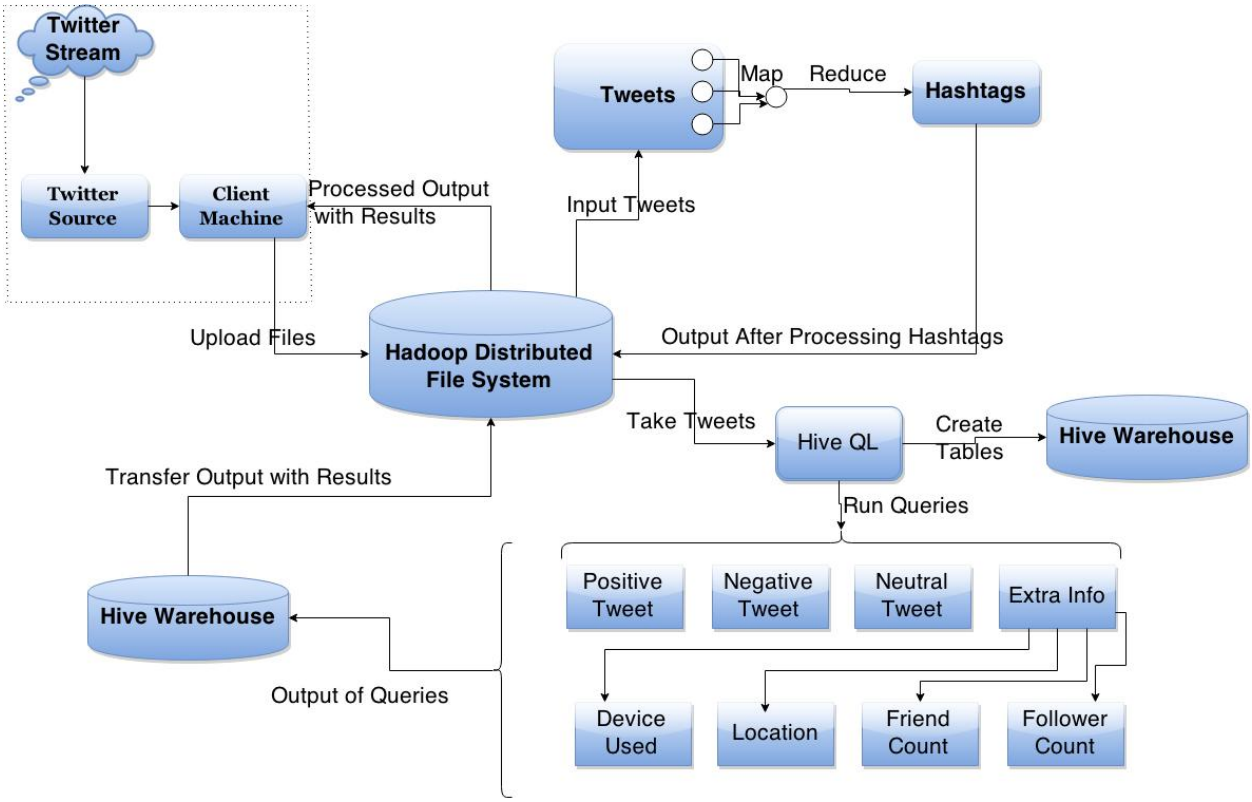


Figure 4:Framework

1. **Twitter Stream**: We use Twitter Api to get Tweets so a Stream is started.The stream is implemented in java using Twitter4j [6].We need to create a twitter Application to use twitter Stream.

2. **Twitter Source:** After the authorization the stream, connection is established between Twitter Source and our machine and we start getting public tweets into our file system.

3. **Client Machine**: This is our machine where the tweets that the twitter source sends are saved.

4. **Hadoop Distributed File System (HDFS)**: After we have the data (tweets), we need to upload them to HDFS to use it with Hadoop.

5. **MapReduce Program**: Now the tweets in HDFS are given as input to a mapreduce program which finds the pattern about Hashtags like the most trending hashtag.

6. **Hive QL:** Hive Query Language is used to take tweets from HDFS and map them to tables and store the schema of these tables in Hive Warehouse.

7. **Hive Warehouse:** It is the place where Hive tables and there scheme are stored.

4.3 MODEL FOR SENTIMENT ANALYSIS USING NAÏVE BAYES

Classification algorithms can be used to automatically classify documents, images, implement spam filters and in many other domains. In this research we are going to use Mahout to classify tweets using the Naive Bayes Classifier. The algorithm works by using a training set which is a set of documents already associated to a category. Using this set, the classifier determines for each word, the probability that it makes a document belong to each of the considered categories. To compute the probability that a document belongs to a category, it multiplies together the individual probability of each of its word in this category. The category with the highest probability is the one the document is most likely to belong to.

4.3.1 Naive Bayes

Naive Bayes is a simple model for classification. It is simple and works well on text categoration. We adopt multinomial Naive Bayes in our project. It assumes each feature is conditional independent to other features given the class. That is,

$$P(C|T) = \frac{P(C)P(T|C)}{P(T)}$$

where C is a specific class and T is text we want to classify. P(C) and P(T) is the prior probabilities of this class and this text. And P (T |C) is the probability the text appears given this class. In our case, the value of class c might be POSITIVE or NEGATIVE, and t is just a sentence.

The goal is choosing value of c to maximize $P(c \mid t)$: Where $P(wi \mid c)$ is the probability of the ith feature in text t appears given class c. We need to train parameters of P(c) and $P(wi \mid c)$. It is simple for getting these parameters in Naive Bayes model. They are just maximum likelihood estimation (MLE) of each one. When making prediction to a new sentence t, we calculate the log likelihood $log\ P(c) + \sum ilogP(wi|c)$ of different classes, and take the class with highest log likelihood as prediction. In practice, it needs smoothing to avoid zero probabilities. Otherwise, the likelihood will be 0 if there is an unseen word when it making prediction. We simply use add-1 smoothing in our project and it works well.

5 INTERPRETATION AND SCORING OF SENTIMENT DATA

5.1 SCORING

5.1.1 Lexicon Approach

The sentiment of the tweet is calculated using twitter text. We use two ways to calculate sentiment using two different lexicon. The sentiment was calculated using lexicon manually created for Twitter. The method used in each tweet is explained as follows

First the tweet text is broken into words, now each lexicon is used one after the other and sentiment is calculated in both the cases.

First lexicon gives a just a nominal category to each tweet, positive, negative and neutral. Here each word is given a polarity as +1 ,-1 or 0, after scoring each word the total score of each tweet is calculated a sentiment is assigned to is as shown:

$$sentiment = Sum(polarity\ of\ each\ word\ in\ the\ tweet)$$

$$if\ sentiment > 0\ \ tweet\ is\ positive$$

$$else\ if\ sentiment < 0\ \ tweet\ is\ negative$$

$$else\ sentiment = 0\ \ tweet\ is\ netural$$

Second lexicon gives an ordinal Score to each tweet. Here each word is given a sentiment value which is in the range +4 to -4, positive score denotes the word is affirmative else the word is negative. Now we calculate the overall score of each tweet as

$$sentiment = Sum(score\ of\ each\ word\ in\ the\ tweet)$$

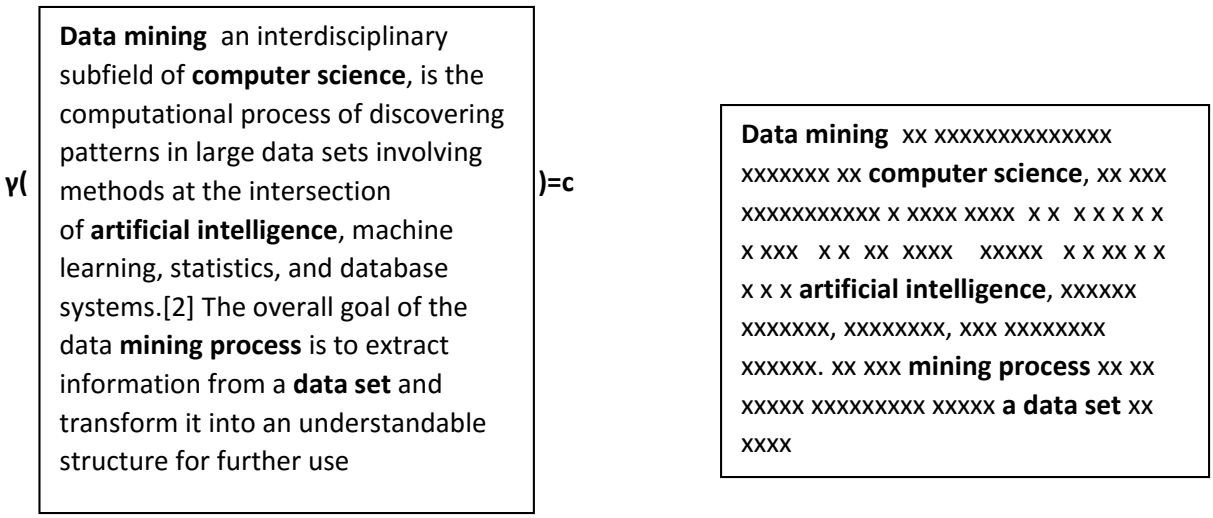
$$more\ postive\ the\ the\ sentiment\ more\ postive\ is\ the\ tweet$$

5.1.2 Naïve bayes Approach

In this approach the initial steps are similar but the scoring is different.

Simple (“naive”) classifica1on method based on Bayes rule
- Relies on very simple representa1on of document
- Bag of words

The bag of words representation



$$P(C|T) = \frac{P(C)P(T|C)}{P(T)}$$

Word	Count
Data Mining	1
Computer Science	1
Artificial Intelegence	1
Mining process	1
Data set	1

Assumptions

- Bag of Words assumption: Assume position doesn't matter
- Conditional Independence: Assume the feature probabilities $P(xi|cj)$ are independent given the class c.

5.2 Data Processing And Calculation of sentiment In Hive

Now all the required data is uploaded into hdfs so now we create Tables in hive.
The following explains the process in detail:

- First we create a table raw_tweets and load all the raw tweets gathered from Twitter In the following format:

  CREATE TABLE raw_tweets (id BIGINT,created_at STRING, source STRING,favorited int,retweet_count INT, Retweeted_text STRING, retweeted_username STRING, Retweeted_userscreenname STRING, screen_name STRING,text STRING, name STRING,friends_count INT, followers_count INT,statuses_count INT, verified BOOLEAN,utc_offset INT, time_zone STRING) ROW FORMAT delimited fields terminated by ',';

- So for this a view is created which selects only the tweets' text id and time zone from raw tweets named as plain tweets.

```
  Id                  Text          Time Zone
424653423245     this's bad good Tweet   New Delhi
```

- Now a view tweets array is created which shows the words of the tweets of an id as a single array of words. It basically breaks down the sentence into words and creates an array. The words are all converted to lower case then passed to the explode function. Then we create a lateral view of all word of an id.

Id	Text
424653423245	[“this’s”,“bad”,“good”,“tweet”]

- The next view created tweet_row further breaks down the array. Each word of a user is written in a single row along with the id acting as key

Id	Text
424653423245	this’s
424653423245	bad
424653423245	good
424653423245	tweet

- Now the tables of dictionaries are used to compute the sentiment of each word by using the left side join operation of hive and sentiment of each word is added after it.
  - Dictionary 1

Id	Text	Score
424653423245	this’s	+1
424653423245	bad	-4
424653423245	good	+3
424653423245	tweet	+1

- By using left outer join on tweet_row and dictionary1 every word in the table a view computed_1 is created. It has a score associated to it by matching the same words on dictionary1. If the word is not present in dictionary1 it is marked as 0. For dictionary2 already a score is present for each word. By using left outer join, on tweet_row and dictionary2, a view computed_2 is created.

- Now we combine the rows in the computed table and group them by id and sum the tweets to find the sentiments. Table’s sentiment1 and sentiment2 are created respectively from computed_1 and computed_2. The score for each id is added using sum function and group by id function for computed_2 and saved in the first table. Whereas for computed_1 the polarity for each id is added and sentiment is added as positive for score greater than zero and vice versa (for zero sentiment is zero).

- The result table also has a column for country which is mapped using left outer join on time zone map. Also for sentiment _result1 (for dictionary 1) the sentiment ‘positive’ is replaced by score +1 and so on.In the end the column for ids is removed to give the final table grouped by country by adding all scores of a particular country.

  The above steps are repeated for each of the topic being analyzed.

5.3 SENTIMENT CALCULATION USING NAÏVE BAYES

5.3.1 Training of model

The only columns required for naïve bayes analysis are tweet id and the tweet itself. Rest of the columns is removed. The dataset collected for training of model is first converted to Hadoop sequence file

format. Sequence file is a flat file consisting of binary key/value pairs. The sequence file generated must have [category] as key and [tweet] as value.

This sequence file is then uploaded to hdfs using the put command.

To use data for analysis in mahout this sequence file has to be further converted to vector files. The command used to convert to vector files is:

$ mahout seq2sparse -i tweets-seq -o tweets-vectors

It will generate the following files in hdfs in the directory name mentioned above:

- df-count: number of document containing this word
- dictionary.file-0: word id
- frequency.file-0: word count
- tf-vectors: sequence file with the term frequency for each document
- tfidf-vectors: tfidf weight for each word in the document
- tokenized-documents: list of words
- wordcount

Now the tfidf vector file is used for training. The significance of tfidf file is:

tf-idf means term frequency-inverse document frequency. It is used as a measure to determine how important a word is to a document in a collection or corpus.

Term frequency: it is the number of times a term occurs in a document. Sometimes this number can be misleading in calculating the weight of a word. For example, the word 'the' will have a high term frequency but if we want to determine the importance of the phrase 'the black sheep' then the term frequency of 'the' overshadows that of 'black sheep', although these two words are more relevant. To avoid this situation an inverse document frequency factor is incorporated. This factor decreases the weight of the word which occurs too frequently in a document and increases the weight of the words which occur very rarely.

Then tf–idf is calculated as

tfidf(t,d,D) = tf(t,d) X idf(t,D) where,

tf(t,d) is the *raw frequency* of a term in a document, i.e. the number of times that term *t* occurs in document *d*.

tfidf(t,D) is the logarithmically scaled fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

where

- N: $total\ number\ of\ documents\ in\ the\ corpus$

- $\{d\ E\ D : t\ E\ d\} : number\ of\ documents\ where\ the\ term\ t\ appears\ (i.e.,\ tf(t,d) = !0).\ If\ the\ term\ is\ not\ in\ the\ corpus, this\ will\ lead\ to\ a\ division - by - zero.\ It\ is\ therefore\ common\ to\ adjust\ the\ denominator\ to\ 1 + \{d\ E\ D : t\ E\ d\}$.

The tfidf vector file is used for training the model. 40% of data is used for training and rest for testing using naïve bayes. The command written is:

$ mahout split -i tweets-vectors/tfidf-vectors --trainingOutput train-vectors --testOutput test-vectors --randomSelectionPct 40 --overwrite --sequenceFiles -xm sequential

We use the training set to train the classifier:

$ mahout trainnb -i train-vectors -el -li labelindex -o model -ow –c

To test that the classifier is working properly on the training set:

$ mahout testnb -i train-vectors -m model -l labelindex -ow -o tweets-testing -c

6 TESTING

The testing of data classified using hive was tested using a list of pre-classified tweets, train them through our unsupervised and compare the results.

The testing of the model build using Naïve Bayes can be done as:

$$\$\ mahout\ testnb\ -i\ test-vectors\ -m\ model\ -l\ labelindex\ -ow\ -o\ tweets-testing\ -c$$

```
        File Input Format Counters
                Bytes Read=2628921
        File Output Format Counters
                Bytes Written=902335
15/04/18 10:02:48 INFO test.TestNaiveBayesDriver: Complementary Results:
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :      17302        99.534%
Incorrectly Classified Instances        :         81         0.466%
Total Classified Instances              :      17383

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       c       d       <--Classified as
367     0       0       0        |  367         a     = InterMilan
0       316     13      0        |  329         b     = netneturality
5       16      13601   45       |  13667       c     = topgear
0       0       2       3018     |  3020        d     = whataretheodds

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.9854
Accuracy                                   99.534%
Reliability                                79.0999%
Reliability (standard deviation)            0.4425

15/04/18 10:02:48 INFO driver.MahoutDriver: Program took 265553 ms (Minutes: 4.4
25883333333333)
[cloudera@quickstart movie]$ java -cp target/twitter-naive-bayes-example-1.0-jar
```

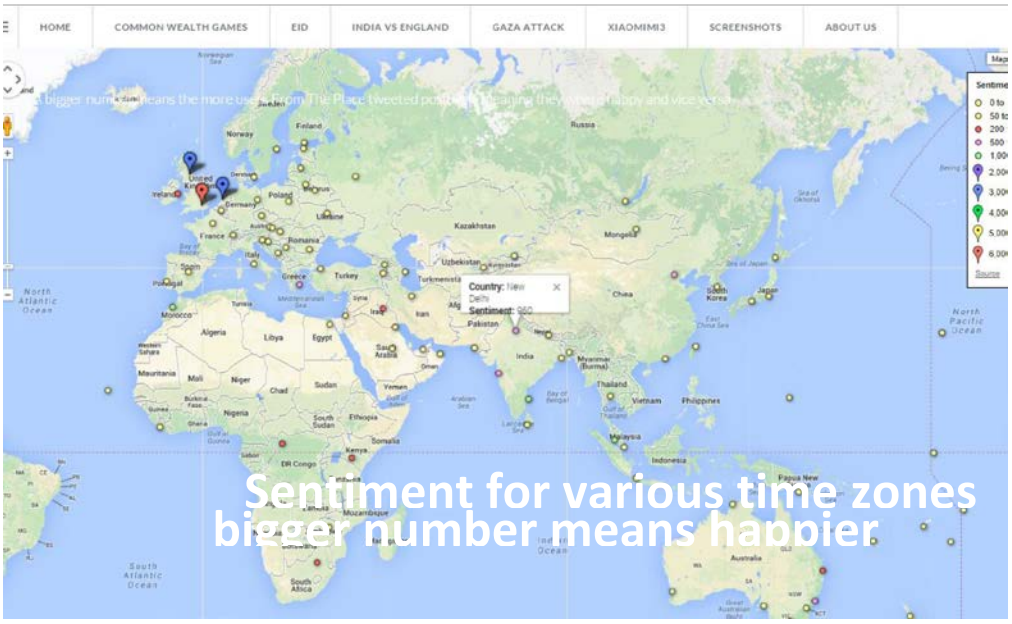
As we can see an accuracy is generated and also a confusion matrix. The confusion matrix shows the incorrect classifications in each category. To improve this accuracy number of tweets in each category can be increased. The categories which rarely appear can be removed.

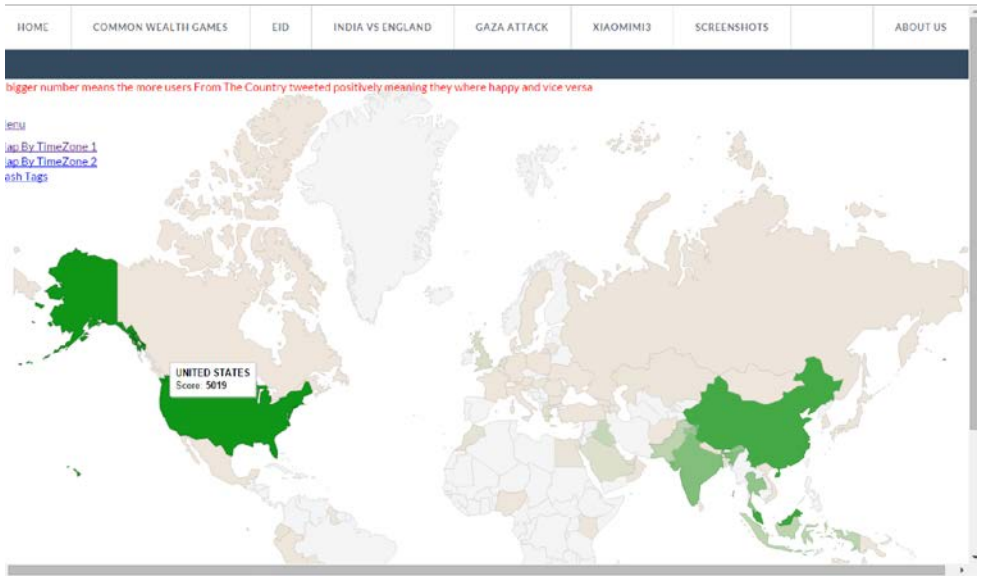
7 EXPERIMENTAL RESULTS

7.1 LEXICON APPROACH

For this project we used 1, 00,000+ tweets collected using the twitter API. The sentiment score for each tweet was calculated using the 2 lexicons and the results obtained grouped using country and time zone and were visualized as shown below.

7.1.1 Screenshots



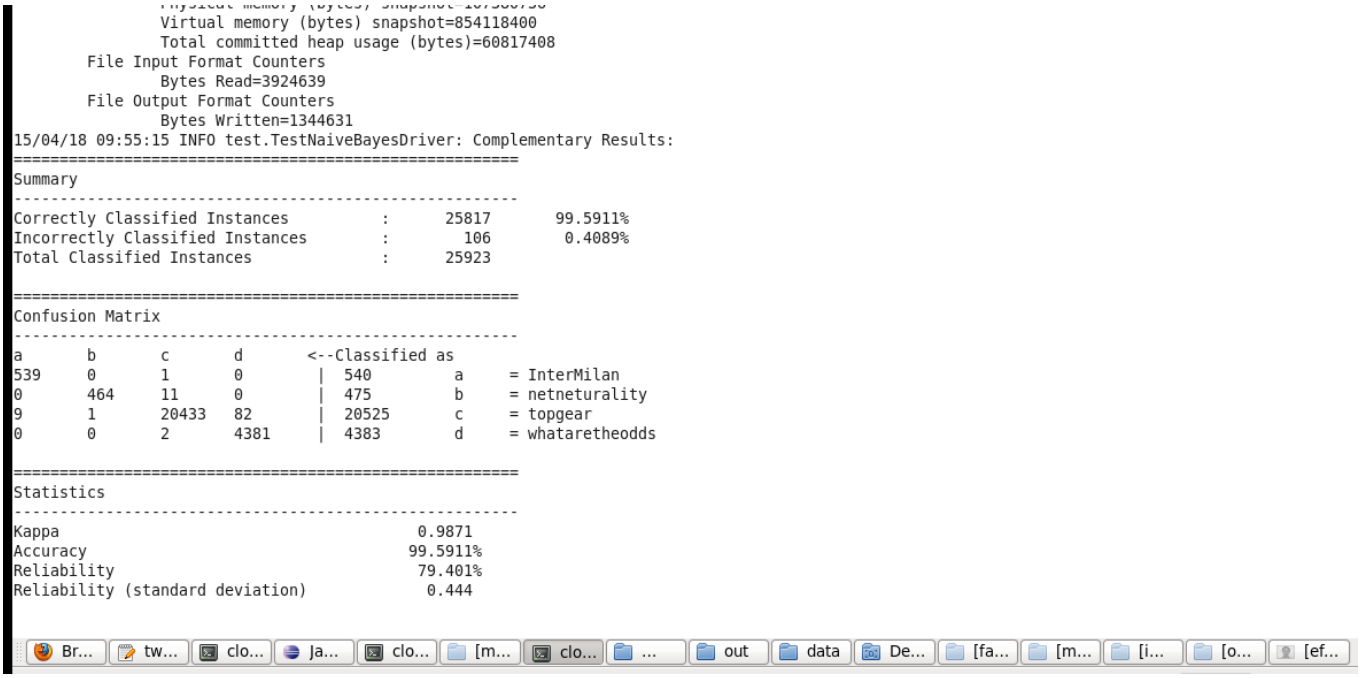


UserName	Followers
bc_321	
Shathasa3eed	
rootshell99999	
e_saudia1	
broadcast_kuw	
al_s3ud_	
mjnonk_q8	
axxa00	
7elaa	
ksatofy	

Country	Tweet
AUSTRALIA	@Fakii: It\'s Kenya 2 4 as ""King"" David Rudisha finishes 2nd in the 800m men\'s final #CommonwealthGames"
CHINA	@Oggyoggs: Yeah boi! ?@Nadalchukwu: She won? ""@micodon2k10: Done and Dusted @Oggyoggs: C\'mon Blessing! Let\'s do this #CommonwealthGam?"
CHINA	@PuddiSRC: ""@BBCSpo : Blessing Okagbare wins gold for Nigeria in the 200m completing the #Glasgow2014 sprint double http://t.co/qX2iRd?"
CHINA	@abuongi: #Atletica Altro ko per @rudishadavid negli 800: ai Commonwealth Games di Glasgow il botswano Nijel Amos
CHINA	@olchux: That girl is something else! Where are the Jamaicans? ""@ColinUdoh: Okagbare. Gold #CommonwealthGames"""
CHINA	@sademazi: Beautiful Motswako...""@OrlandoGhost: Bianca Williams is beautiful #CommonwealthGames #final200m women"""
CHINA	@2014games: Disability Activists spread \'ATOS Kills \' message at #CommonwealthGames http://t.co/WjRCE63ctx by @RoryMacKinnon via @M_Star?
CHINA	@500voicesforyes: See if Scotland doesnt #voteyes after outrageous @bbc #Glasgow2014 coverage you\'ve got to ask yourself is this country?
CHINA	@ASFevents: SILVER for Eilidh Child #goscotland http://t.co/3B2Ed87rSp @Team_Scotland @Glasgow2014

7.2 NAÏVE BAYES APPROACH

The classification of new tweets is done on the model built. For every classification we obtain accuracy. The screenshot after testing of data is presented in the following figure.



Only 106 instances or tweets are wrongly classified out of 25,817 tweets.

8 CONCLUSION

In this project, we presented the experiments we have done on sentiment classification using large-scale data set. This project gives us an overview and a simplistic approach to sentimental analysis. As we can see both the approaches used are only in its beginning stages. Improvements can be done in both data preprocessing as well as the classifying algorithms. For instance, tweets of languages other than English can be incorporated. We have observed that supervised classification of tweets is more efficient and accurate as compared to that of unsupervised approach. As the number of categories increase it becomes more and more difficult to classify tweets initially for obtaining a labeled data set required for training. Our experiments on twitter sentiment analysis show that part-of-speech features may be useful for sentiment analysis in the microblogging domain. More research is needed to determine whether the POS features are just of poor quality due to the results of the tagger or whether POS features are just less useful for sentiment analysis in this domain. Using hashtags to collect training data did prove useful, as did using data collected based on positive and negative emotions. We could further get better results if you also confider groups of words there meaning when used together,POS unlike the present approach of using words to be mutually exclusive.

9 FUTURE SCOPES

- The limitation of this paper can be that the data which is processed is not live. The data being streamed and stored in the text file is live. But this gathered/collected data is further processed to compute results hence it is not live.
- Regarding sentiment analysis it isn't complete yet. It is still in its infancy, and there are limitations. Despite significant advances in machine learning, it's extremely difficult (or not practically efficient) for computers to understand and process natural language, automate sentiment analysis, or determine ambiguous context.

- Twitter has users from all over the world but all the tweets are not being processed. This is because foreign scripts are not detected. The dictionaries used have only English words with their sentiment scores. Tweets with other scripts are filtered out.

- The problem is that the more you break down the data, the less likely it is that automated analysis will get it right. Automated sentiment works best with large amounts of data and can't be relied upon for smaller samples. A lot of humans struggle with sarcasm and irony, so how can we expect computers to cope? This is particularly problematic when looking at Twitter. There are also examples of posts which contain both positive and negative sentiment. For instance: "The food was fantastic but the service was terrible". In this case a computer won't know which way to turn.

10 REFERENCES

[1] http://www.internetlivestats.com/twitter-statistics/

[2]http://docs.aws.amazon.com/gettingstarted/latest/emr/getting-started-emr-sentiment-tutorial.html

[3] http://www-01.ibm.com/software/data/infosphere/hadoop/what-is-big-data-analytics.html

[4] https://hadoop.apache.org/

[5] http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf

[6] http://twitter4j.org/