Sentimental Analysis of Twitter Data Using Classifier Algorithms

Sharvil Shah*, K Kumar, Ra. K. Saravanaguru****
* Software Developer, Triforce Solutions, Ahmedabad, India
** School of Computing Science and Engineering, VIT University, Vellore, India

Article Info

Article history:

Received Sep 7, 2015
Revised Oct 26, 2015
Accepted Nov 11, 2015

Keyword:

Microblogging
Naïve Bayes
Sentimental Analysis
Twitter

ABSTRACT

Microblogging has become a daily routine for most of the people in this world. With the help of Microblogging people get opinions about several things going on, not only around the nation but also worldwide. Twitter is one such online social networking website where people can post their views regarding something. It is a huge platform having over 316 Million users registered from all over the world. It enables users to send and read short messages with over 140 characters for compatibility with SMS messaging. A good sentimental analysis of data of this huge platform can lead to achieve many new applications like – Movie reviews, Product reviews, Spam detection, Knowing consumer needs, etc. In this paper, we have devised a new algorithm with which the above needs can be achieved. Our algorithm uses three specific techniques for sentimental analysis and can be called a hybrid algorithm – (1) Hash Tag Classification for topic modeling; (2) Naïve Bayes Classifier Algorithm for polarity classification; (3) Emoticon Analysis for Neutral polar data. These techniques individually have some limitations for sentimental analysis.

Corresponding Author:

Sharvil Shah,
7, Heritage Enclave,
Thaltej, Ahmedabad – 380059, India
Email: sharvil.shah1994@gmail.com

1. INTRODUCTION

With the increasing number of users and tweets, it would be best to analyze the twitter data to get to know about various relevant things going on around us. Monitoring and reviewing the perspective from social media provides great opportunities for public and private sector. For example, a company is able to know if the announcement of a product has negative or positive impact. A Political leader can know if he has got any chances to win in the upcoming elections. Area of Sentimental Analysis is appealing to a lot of researchers and scientists due to the challenges it offers and its potential applicability [1].

The sentimental analysis could lead to several challenges like data sparsity which is because of slang language used due to word limit. Also, this platform is an open domain where users can post about anything which leads us to build a sentiment classifier. To reach maximum efficiency and accuracy our algorithm should run in real time [3].

In this paper, we not only give a binary classification of positive and negative data but also give a hash tag classification for topic modeling, an emoticon analysis for determining polarity of the post, multilingual support by using tools like *Google Language Detector* and *Langid* [1]. We also give a graphical representation of the sentimental analysis by making use of Google Chart Tools. In this paper, we portray an algorithm which can try to detect the current attitude of the user towards a particular topic.

For sentimental analysis our approach mentioned in this paper is divided in the following parts:

- Data Retrieval: The first approach is retrieval of data from twitter by using twitter APIs

- Pre-processing: The retrieved data is pre-processed for further action
- Hash Tag Classification [12]: The pre-processed data is then classified into topic wise data by parsing
- Polarity Classifier: After hashtag classification the data is then classified to subjective and objective data with help of Naïve Bayes Classifier and Polarity Shifter [4, 5].
- Emoticon Analysis: If the algorithm is unable to classify polarity, then the parser looks for emoticon and classifies data accordingly

Please refer to Figure 1 for further details

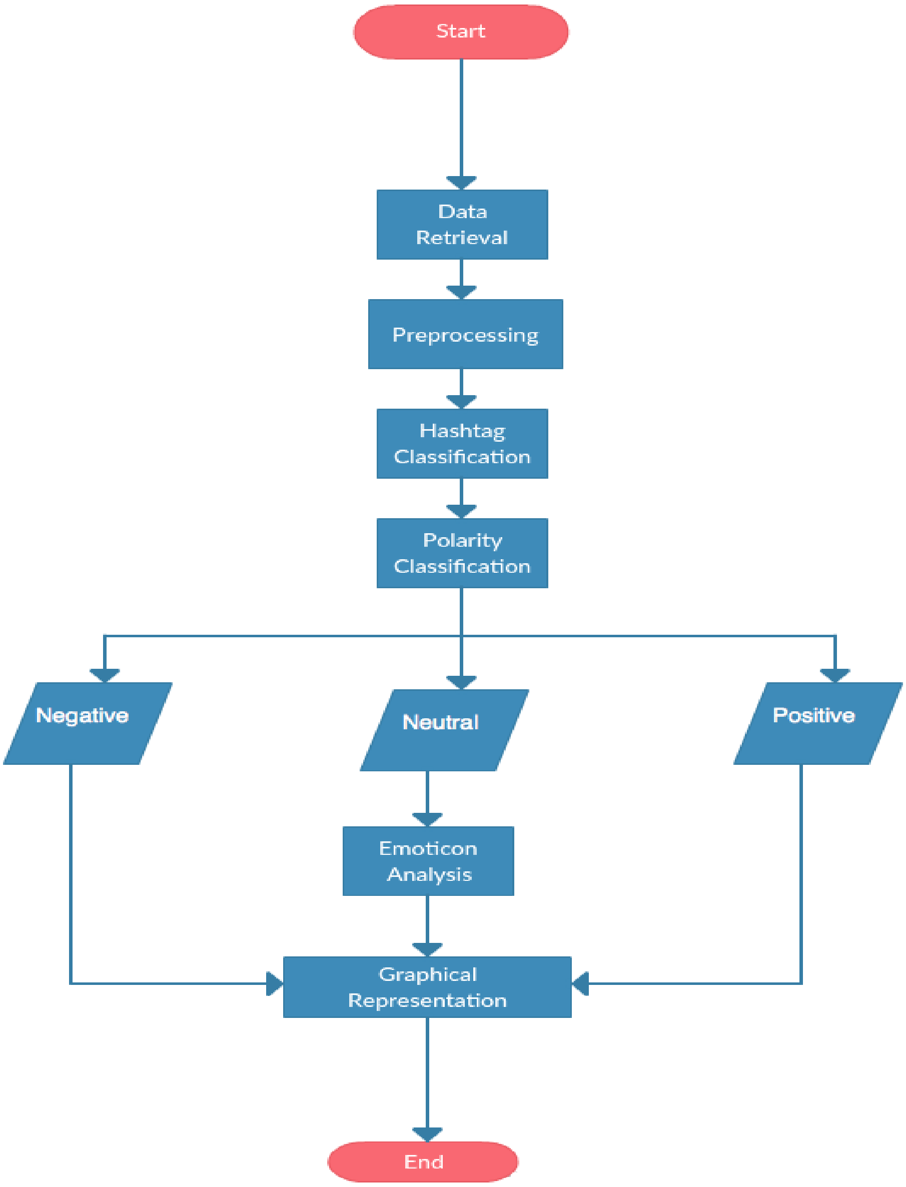

Figure 1. System Flowchart

2. RELATED WORK

Sentimental Analysis is a booming topic in field of research. It has been studied for years on various text corpus like newspaper articles, movie reviews and product reviews. In the very beginning, research on this topic was done with the help of Maximum Entropy and Support Vector Machines to detect the sentiments. The maximum optimal result of 83% was claimed by one of the researchers named MaxEnt. But during this traditional research they were not able to classify data in neutral sentiment. To overcome this limitation, Pak and Paroubek additionally retrieved neutral tweets and then used 3-class Naïve Bayes Classifier which was able to detect neutral messages along with the polar ones [9].

Researchers from IIT Bombay, India published paper on Twisent [23], a sentimental analysis system for Twitter. It collects tweets pertaining to it and categorizes them in different polarity classes – positive, negative and objective. However, analyzing micro-blog posts have many inherent challenges compared to other text genres [24]. Researchers named Barbosa and Feng used 2 classifiers - Subjective versus Objective classes and Positive versus negative classes. They present separate evaluation on both models but do not explore combining them or comparing it with a 3-way classification scheme [14]. Jiang et al., 2011 present results on building a 3-way classifier for Objective, Positive and Negative tweets. However, they do not explore the cascaded design and do not detect Neutral tweets [16].

Table 1. Related Work by different authors

Author	Advantages	Limitations
Spencer [4]	Naives Bayes Classification	No hash tag classification
Apoorva [11]	End to end pipeline for classifying tweets, tree kernel model, 100 senti features model, kernel plus senti features, unigram plus senti features	Topic modeling
Alexander [1]	Classifies data with proper accuracy	Multi lingual support
Theresa [12]	Hash Tag classification	Uses iSieve data set- very narrow data
Sunil [25]	Real time analysis using Hadoop	Does not understand sarcasm
Go et al.	Binary classification	Did not improve classification performance
Pak and Paroubek [1]	Entropy, salience and naïve bayes classification for classifying microblogs	They remove URLs, usernames, retweets, emoticons and article stopwords (a, an, the) from all tweets and tokenize on whitespace and punctuation
Barbosa and Feng [13]	Two step Classifier – Subjective and Objective for better classification	Their approach only possible with limited data
Bermingham and Smeaton [26]	Collect tweets of ten so-called trending topics for each of the five categories "entertainment, products and services, sport, current affairs and companies" (Bermingham and Smeaton, 2010) to build a manually annotated dataset	More accurate with short tweets
Sumit	Graphical Rep of Twitter Data	No emoticon analysis

3. DATA RETREIVAL

The data from Twitter can be retrieved in many ways like – Using Twitter Search APIs, NodeXL or Kimonofy Tool using which we can generate APIs and import all the required data. We need to do this in real time and so our system faces millions of tweets at once. This data is preprocessed and classified according to the polarity. The Emoticon dataset can retrieve from *twittersentiment.appspot.com*.

In this section, results of research are explained and at the same time is given the comprehensive discussion. Results can be presented in figures, graphs, tables and others that makes the reader understand easily [2], [4]. The discussion can be made in several sub-chapters.

3.1. Preprocessing

The preprocessing of tweets is a very important part of this paper. The data retrieved in JSON format is first converted to normal text message. It contains following –

- All caps identification
- Lower casing
- URL Removal
- Emoticon Analysis
- Removal of Punctuations and White spaces
- Letter Redundancy / Compression of Words

Since the tweet can be in Lower case or Upper case, for the convenience of the algorithm at first the text is converted to lower case [11]. It is possible that the tweet can have URLs, so all the URLs are eliminated from the messages with the help of regular expression or replacing with generic word URL.

The usernames mentioned in the data retrieved are eliminated with the help of regular expression or replaced by any other word which is having a neutral polarity. The words having hash tag remains unchanged so that they can be used for topic modeling [24]. If the word is having many redundancies like 'happpppyyyyyyy', then such words are converted to 'happyy' by removing maximum redundancies possible and keeping up to two repetitions. Punctuations and additional white spaces are removed keeping only one white space in the middle of words and eliminating punctuations with the help of parsing.

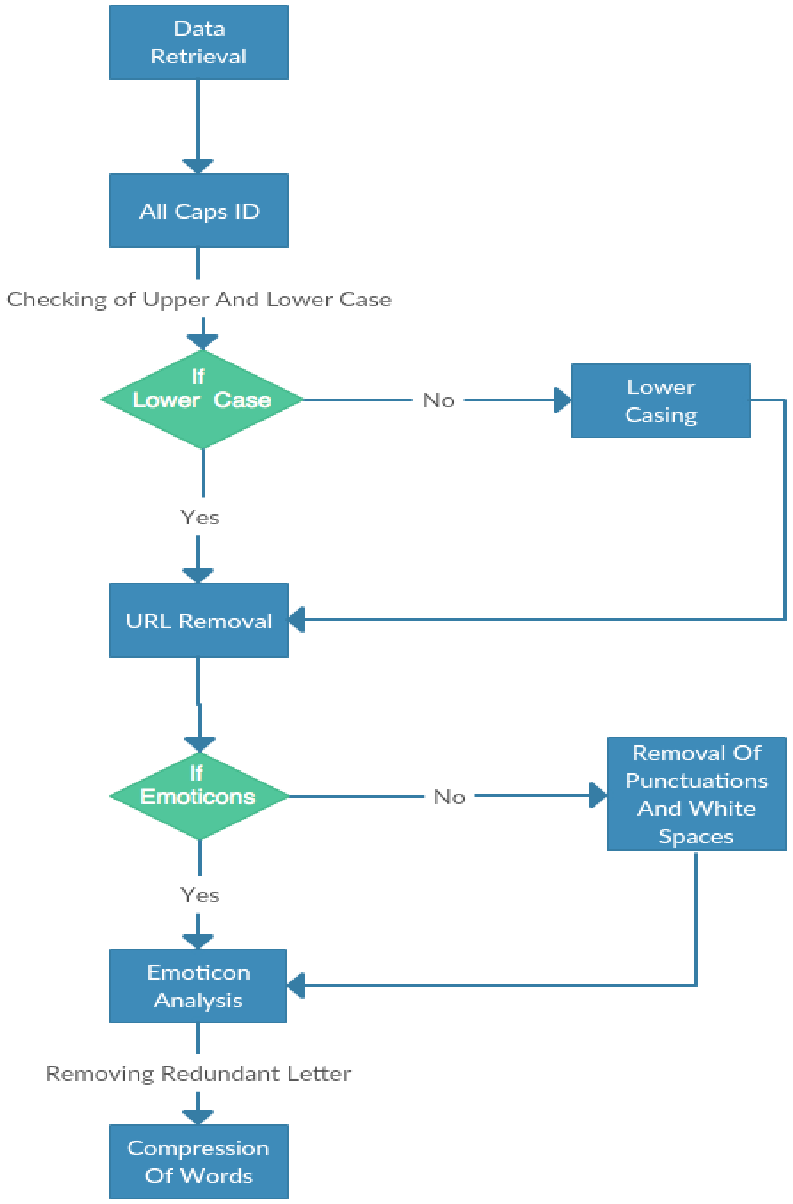

Figure 2. Preprocessing

3.2. Hashtag Classification

Hashtag classification is very important for topic modeling [6, 10]. While posting any message, the user uses a hash tag, for eg. #IndvsAus. So, from this we can know that the post is about the India versus Australia match. This can help in classifying the preprocessed data in various topics.

We do not change the hash tag words during preprocessing. With the help of data parsing, the algorithm can identify the hash tagged words and with the help of that particular text message is classified into that group so that the data does not get mixed up and because of that accuracy increases. Our algorithm does not remove the less used hashtags instead it concentrates on the most used hashtags [13]. Hashtag in the message can prove very much crucial for classifying the data.

One challenge that Hashtag classification might pose is that after the hash symbol, the text is concatenated because of which there might be a problem of topic modeling [8]. To overcome this, we have proposed a small algorithm as follows –

In general, people write hashtags in a concatenated format. There are no white spaces or special character in between which parser can identify to split the text. For example during World cup 2015, tweets related to all Indian matches had a hashtag '#WeWontGiveItBack' or '#wewontgiveitback' or

'#WEWONTGIVEITBACK'. First kind is used more than the second and the third kind. But, we cannot rely on people putting a capital letter at starting of every new word. So, we make a list of prepositions, conjunctions and 'wh' question words. Using that list, the parser searches for the word (irrespective of the case used) as given in the list, if it finds the word, it puts whitespace in front and rear of that particular word. If the word searched is having the first position i.e. immediately after the hashtag then the hashtag is removed and white space is inserted only in the rear. So, in our example the parser make the hashtag text as 'We WontGive It Back' and then the tweets are classified accordingly [20].

4. POLARITY CLASSIFIER

Polarity classifier is the heart of this paper. We use Naïve Bayes Classifier, Unigram and Bigram models for classification of polar data. We have distributed data into – Subjective and Objective data. In subjective data, we include data with positive and negative sentiments. In objective data we include data having neutral sentiments and emoticons [7].

4.1. Naïve Bayes Classifier

This classifier uses simple approach based on Bayes Theorem which describes - how the conditional probability of each of a set of possible causes for a given observed outcome can be computed from knowledge of the probability of each cause and the conditional probability of the outcome of each cause. It is a Bag of Words approach for subjective analysis of a content [9, 10].

According to the Bayes Theorem, for a document d and class c –

$$P\left(\frac{c}{d}\right) = \frac{P\left(\frac{d}{c}\right).P(c)}{P(d)}$$

Naïve Bayes Classifier would be –

$$c *= \arg maxcP(\frac{c}{d})$$

Using Naïve Bayes Classifier we can determine the accuracy of classification [5, 22]. Generally, for efficient algorithm the accuracy turns out 80%. According to Figure 3 given below, after the topic modeling is done the data is given sentiments and distributed according to the polarity – positive, negative and neutral. The crucial disadvantage of Naïve Bayes Classifier is that it supposes conditional independence among linguistic features.

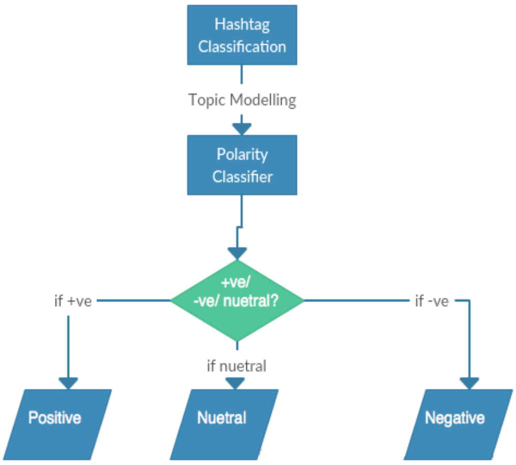

Figure 3. Polarity Classifier

4.2. Data Classification

In this paper, we use two strategies for classifying the polarity [14] –

4.2.1. Binary

In binary classification, the data is classified just in two types for the convenience of the algorithm – positive and negative. We consider this classification for the case when we only need positive or negative results and no neutral classification [17]. So, a basic binary classification helps in separating positive and negative sentiments of the data. If the data is classified with positive and negative polarity then the remaining data having no polarity (neutral) is sent for emoticon analysis. Later, according to the emoticons used the data is classified accordingly. Thereafter, the remaining neutral data is ignored for the case when we need only positive and negative results.

4.2.2. Baseline

In baseline classification, we distribute data into positive, negative and neutral. We consider the topics in which neutral sentiment is having an importance [18]. For example during elections, some people are neutral towards a particular party. We use a rule based classifier in which according to the polar lexicon list, the data are given their sentiments. The neutral results are sent again for emoticon analysis and if the sentiment is found negative or positive then the data is classified accordingly, rest of the data is hence termed as neutral data. From the baseline Naïve Bayes Classifier we can achieve an accuracy of about 80% [19].

4.3. Polarity Shifter

If a noun, verb or adjective is having a positive polarity and the word before that is a negation like *'not'* then the accuracy might decrease. To overcome this, we have proposed an algorithm in which it searches for the negation words. When the parser finds the negative word it looks for three words beyond negation. If the three word window is having a noun, verb or an adjective which has positive polarity then the polarity of that data is reversed. Using this polarity shifter, we can achieve results with maximum accuracy.
For example Let us take a data *'The movie was not good'*. Now, the text is having a positive sentiment word i.e. 'good', so there is a possibility that the machine classifies this data as positive sentiment. But, with polarity shifter, that would not be possible because we even look at negation. The polarity is reversed and that data is classified into negative sentiment [22].

4.4. Emoticon Analysis

After the polarity classification phase, the neutral data having emoticons are then analyzed. If the sentence is having positive and negative emoticons are then classified into positive and negative sentiments [15, 21]. For example let us look at a tweet posted by an athlete *'I just finished a 2.66 mi run with a pace of 11'14"/mi with Nike+ GPS :D :D. #nikeplus #makeitcount'*. According to this tweet, the text is not having any positive or negative sentiments hence it is classified to neutral tweet. But, the tweet is having an emoticon which is ':D' which shows positive sentiment about nikeplus [21]. Therefore, this text is classified as positive sentiment text. The algorithm of emoticon analysis works as follows –

The data from the neutral section is analyzed and emoticon is searched in the sentence [18]. The two letters after ':' symbol is important in this case. If white space is present after the ':' symbol then it is ignored but if a letter is present after the symbol then the emoticons are then classified accordingly.

Table 2. Emoticon List

Emoticon	Sentiment
:)	Positive
:(	Negative
:D	Positive
:|	Negative
:'(	Negative
;)	Positive
:/	Negative
:O	Negative

Table-2 shows list of example emoticons with their respective sentiments. In this way, they are classified according to their sentiments.

5. ALGORITHM AND CASES

So from the above modules, our overall algorithm works as follows –

- The data is extracted from twitter using twitter APIs or Hadoop. The data imported is stored in JSON or any other relevant format.
- This data is then sent for pre-processing where the data is simplified. Here, the following processes takes place –
  1. Caps Identification
  2. Lower Casing
  3. URL Removal
  4. Removing Username from post
  5. Removal of Punctuations and white spaces
  6. Taking care of letter redundancy
- After the data is pre-processed we obtain a refined data. Now, the algorithm does topic modeling of the given data and it is classified according to the various topics using the hashtag algorithm as stated above.
- Later on the data is classified according to the sentiments using Naïve Bayes Classification Algorithm. We use basic binary and baseline pattern to classify the data. To improve the accuracy we also follow a polarity shifter algorithm which can identify the negation used in the text and then act accordingly.
- The data which is classified into neutral sentiment is then sent for an emoticon analysis. Here, the neutral data is classified to subjective sentiments and remaining data which does not have any emoticons stay in the neutral section.

5.1. Pseudo Code

```
Algorithm: Sentimental Analysis of Twitter Data
Input: Set of all the data retrieved D
Output: Polarised data P
Initialize Data Retrieved set D
Initialize Selected token set S
//Converting to Lower case
foreach t € D do
i ←t.tweet;
if S(i) = NULL then
S(i) = t;
else S(i)=lowercase();
//Remove URL
foreach t € D do
i←t.tweet;
if S(i)=NO URL then
S(i)=t;
else S(i)=t.sub('((www\.[^\s]+)|(https?://[^\s]+))','URL',tweet);
//Removing username
foreach t € D do
i←t.tweet;
S(i) = t.sub('@[^\s]+','AT_USER',tweet);
//Remove additional white spaces
foreach t € D do
i←t.tweet;
S(i) = t.sub('[\s]+', ' ', tweet);
//Topic Modeling
S = t.sub(‘#word according to the list’,’’)
Load the topic wise separated tweetes in different data store
foreach t € D do
i←t.tweet;
S(i)=t.store();
//Polarity Claissifier
if(tweet containing positive word) then
t.positivesentiment();
elseif(tweet containing negative word) then
t.negative sentiment();
elseif(tweet containg negation) then
    if(next 3 words are polar noun, verb or adj)
```

```
t.reversepolarity();
elseif(emoticon=TRUE) then
            if(emoticon=positive) then
t.positivesentiment();
elseif(emoticon=negative) then
t.negativesentiment();
            else t.neutralsentiment();
```

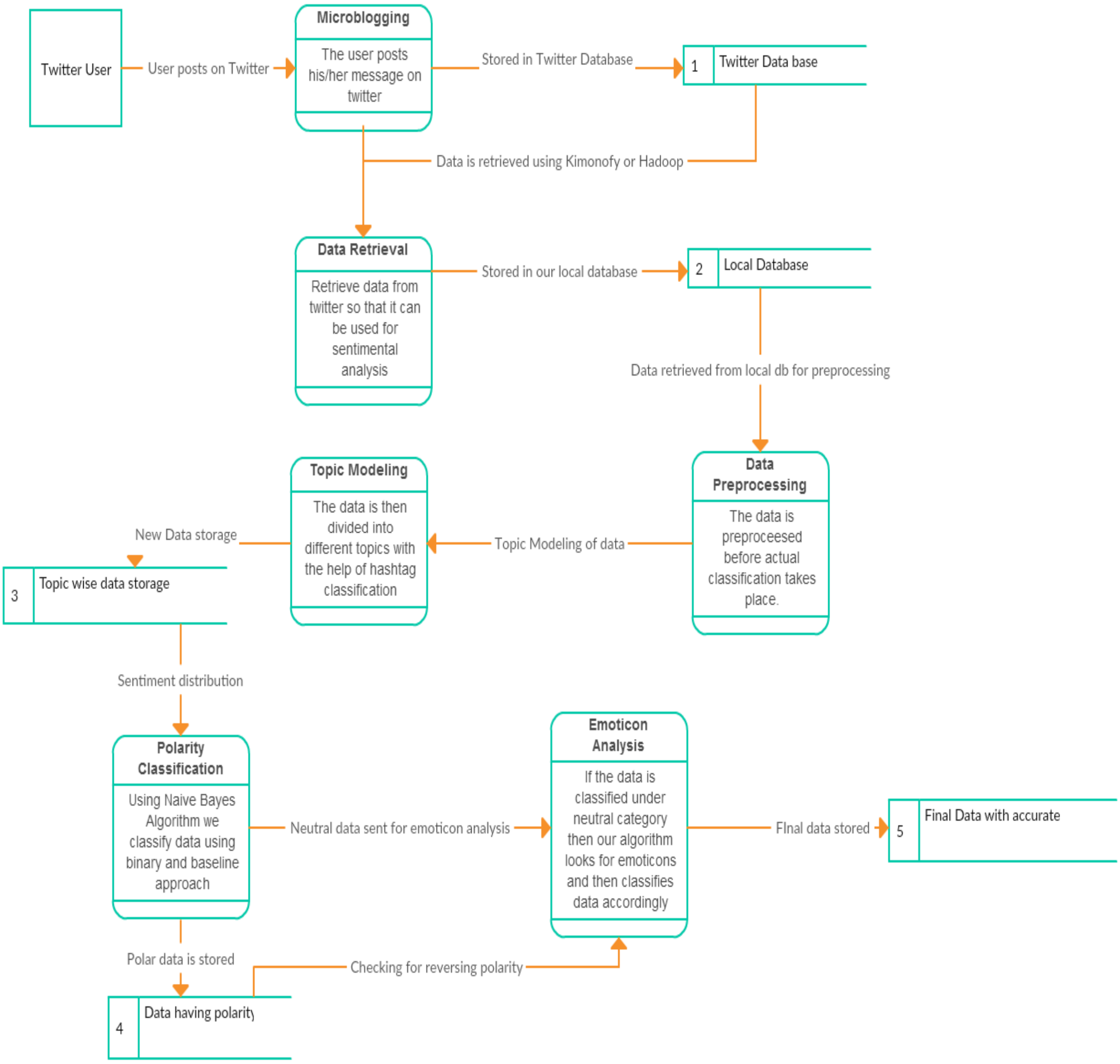
Twitter User
User posts on Twitter
Microblogging
The user posts his/her message on twitter
Stored in Twitter Database
1
Twitter Data base
Data is retrieved using Kimonofy or Hadoop
Data Retrieval
Retrieve data from twitter so that it can be used for sentimental analysis
Stored in our local database
2
Local Database
Data retrieved from local db for preprocessing
Data Preprocessing
The data is preproceesed before actual classification takes place.
Topic Modeling of data
Topic Modeling
The data is then divided into different topics with the help of hashtag classification
New Data storage
3
Topic wise data storage
Sentiment distribution
Polarity Classification
Using Naive Bayes Algorithm we classify data using binary and baseline approach
Neutral data sent for emoticon analysis
Emoticon Analysis
If the data is classified under neutral category then our algorithm looks for emoticons and then classifies data accordingly
FInal data stored
5
Final Data with accurate
Polar data is stored
4
Data having polarity
Checking for reversing polarity

Figure 4. Overall Flow of Algorithm

5.2. Cases

Table 3 explains different cases for our algorithm

Table 3. Cases

Case No	Case	Description	Example	Result
1	Positive Sentiment Data	Tweets with positive sentiment is tested with our algorithm	@XYZ: The movie was aamaaaazzzingg !! :D	Success
2	Negative Sentiment Data	Tweets with negative sentiment is tested with our algorithm	@ABC: It is such a bad day	Success
3	Neutral Sentiment Data	Tweets with neutral sentiment is tested with our algorithm	@USR: I gave an English Test today	Success
4	Neutral Sentiment Data with Emoticon	A tweet which is having neutral sentiment but with emoticon is tested with our algorithm	@USR: I ran 2.5 kmstoday ! :D	Success
5	Positive Sentiment with Negation	A tweet which has positive words but a negation is starting	@USR: Today is not good	Success
6	Topic modeling using hashtag	A tweet is classified according to topics by using hashtag	@USR: Hoping for narendramodi to win elections #NamoForIndia	Success

6. RESULTS AND DISCUSSION

Compared to other works done uptill now, our final algorithm has all the accuracy maintaining features like – Hash Tag Classification for Topic Modeling, Polarity Shifter, emoticon Analysis and Graph generation for getting accurate results. Because of these features, our algorithm is better than other works done in this field. An average accuracy of 81% is among the highest reported in research of this field. The polarity shifter and topic modeling are two crucial steps in our algorithm which leads our algorithm to a higher accuracy.

Let us take seven tweets regarding a specific match India vs Australia. In this paper, we will show working of one tweet in the algorithm and then the graph generation. Let us take an example tweet as –

"*@abc: Having a great feeling while watching the match #IndvsAus*"

Step-1

The above tweet first goes for preprocessing. The whole tweet is converted to lower case. Hence the tweet looks like – "*@abc: having a great feeling while watching the match #indvsaus*"

Step-2

The algorithm searches for URL, but since there are no URL in this tweet so moves ahead to the next step in which it removes the username. So after removing username and converting it to generic name, our tweet looks like – "*at_user having a great feeling while watching the match #indvsaus*"

Step-3

In this step, additional white spaces are removed from the tweet and topic modeling takes place. After applying the hashtag classification algorithm the data is stored under India vs Australia match topic and our tweet looks like – "*at_user having a great feeling while watching the match*"

Step-4

The polarity classification takes place using Naïve Bayes classification algorithm and hence the graph is generated as shown below. This example tweet is classified into positive sentiment. Please refer figure 5 for graphical view.

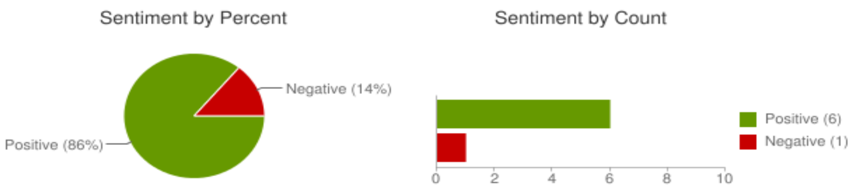

Figure 5. Graphical View

The result shown in figure 5 is not 100% accurate. On manual check it is found that one of the tweet was showing wrong polarity, so accuracy comes around 85% for seven tweets. Similarly, we tested this algorithm for various number of tweets and an average accuracy of 81% was found by using our algorithm.

7. CONCLUSION

The algorithm used by us is pretty much accurate for classification of data according to the sentiments as discussed earlier and gives us a great average accuracyas discussed in Results and Discussion 6. The speciality of this algorithm is that it uses all the three techniques and this mixture results into a good outcome as seen in this paper.All the steps used in the algorithm are well built and tested several times and this custom algorithm designed by us is efficient and better than other works in this field.There are no unnecessary steps in this algorithm which would lead to a time consuming process.

As we can see in Table 1, all the works have some or the other limitations. Our work over comes these limitations. Spencer's limitation of hash tag classification, Apporva's limitation of Topic modeling, Theresa's limitation of narrow data, Go et al.'s limitation of classification performance and Bermingham's limitation of accuracy with short tweets are all covered up by our algorithm. In our future work, we would like to implement an algorithm which can detect sarcasm in a better way and can give accurate results. Pattern extraction can be considered for getting recurring information.

REFERENCES

[1] Twitter as a Corpus for Sentiment Analysis and Opinion Mining By Alexander Pak, Patrick Paroubek.

[2] EthemAlpaydin. 2004. Introduction to Machine Learning (Adaptive Computation and Machine Learning). The MIT Press.

[3] Emoticon Smoothed Language Models for Twitter Sentiment Analysis by Kun-Lin Liu, Wu-Jun Li, MinyiGuo.

[4] Sentimentor: Sentiment Analysis of Twitter Data by James Spencer and Gulden Uchyigit.

[5] Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets* by Pablo Gamallo and Marcos Garcia.

[6] Antonio Fernandez Anta, Philippe Morere, and Agust´ın Santos. 2013. Sentiment Analysis and Topic Detection of Spanish Tweets: A Comparative Study of NLP Techniques. Procesamiento del Lenguaje Natural.

[7] Alec Go, RichaBhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision.

[8] Pak, A., Paroubek, P. Twitter as a corpus for sentiment analysis and opinion mining. In: Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10).

[9] David Ahn& Balder ten Cate. Simple language models and spam filtering with Naive Bayes, 2005.

[10] S. Baccianella, A. Esuli, and F. Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.

[11] End-to-End Sentiment Analysis of Twitter Data by Apoor v Agarwal and Jasneet Singh Sabharwal.

[12] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data.

[13] Barbosa, L. and Feng, J. (2010). Robust sentiment detection on twitter from biased and noisy data. Proceedings of the 23rd International Conference on Computational Linguistics.

[14] Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. Technical report, Stanford.

[15] Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011) on Target-dependent twitter sentiment.

[16] Kim, S. M. and Hovy, E. (2004). Determining the sentiment of opinions.

[17] Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity analysis using subjectivity summarization based on minimum cuts.

[18] Turney, P. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews.

[19] Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. Conference on Empirical methods in natural language processing.

[20] David Haussler. 1999. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz.

[21] CM Whissel. 1989. The dictionary of Affect in Language. Emotion: theory research and experience, Acad press London.

[22] T. Wilson, J. Wiebe, and P. Hoffman. 2005. Recognizing contextual polarity in phrase level sentiment analysis.

[23] TwiSent: A Multistage System for AnalyzingSentiment in Twitter by Subhabrata Mukherjee, AkshatMalu, A.R. Balamurali, Pushpak Bhattacharyya.

[24] Sentimenatal Analysis of Twitter Data by Apoorv Agarwal BoyiXie Ilia Vovsha Owen Rambow Rebecca Passonneau Department of Computer Science Columbia University New York, NY 10027 USA.

[25] Real Time Sentiment Analysis of Twitter Data Using Hadoop by Sunil B Mane, YashwantSawant, SaifKazi, VaibhavShinde in IJCSIT, ISSN:09745-9646.

[26] Classifying Sentiment in Microblogs: Is Brevity an Advantage? By Adam Bermingham and Alan Smeaton, Dublin City University.