*2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*
*Zeal Education Society, Pune, India, Feb 24-26, 2017*

# Twitter Sentiment Analysis for Product Review Using Lexicon Method

Paramita Ray[1] and Amlan Chakrabarti[2]

[1]Computer Sc Dept , Dinabandhu Andrews Institute of Technology and Management, Kolkata, India [2]A.K. Choudhury School of Information Technology , University of Calcutta, Kolkata, India

rayparamita@yahoo.com[1] achakra12@yahoo.com[2]

*Abstract--* **In recent times, people share their opinions, ideas through social networking site, electronic media etc. Different organizations always want to find public opinions about their products and services. Individual consumers also want to know the opinions from existing users before purchasing product. Sentiment analysis is the computational treatment of user's opinions, sentiments and subjectivity of text. In this paper we propose a framework for sentiment analysis using R software which can analyze sentiment of users on Twitter data using Twitter API. Our methodology involves collection of data from twitter, its pre-processing and followed by a lexicon based approach to analyze user's sentiment.**

***Keywords— Data analysis, data preprocessing, data mining, sentiment analysis***

## I) INTRODUCTION

Sentiment analysis (also called *opinion mining)* means to analyze people's real opinions, sentiments, evaluations, appraisals, attitudes, and emotions regarding specific product, organization, services, movies, individuals, political events, topics, and their attributes[1]. Customer feedback about particular products is very important for commercial organization. They can improve their product quality [2], services on the basis of customer opinion about their product. Twitter is a kind of micro-blogging social networking site and billions of users use it to give their opinion [3] related to a particular topic. On the basis of opinion, sentiments can be estimated through analysis.

In this paper we collected large number of tweets from twitter server by using Twitter API using different keywords [6] within a particular time period and date. We categorized the tweets into positive, negative, or neutral opinion [8] to estimate the overall sentiment of customer or user about particular products or services and that can be implemented in different domains to improve their product or service quality.

## II) SENTIMENT ANALYSIS METHODOLOGY: BACKGROUND

Text categorization was started long time ago (Salton and McGill, 1983), however [1] categorization based on sentiment was introduced more recently in (Das and Chen, 2001; Morinaga et al., 2002; Pang et al., 2002; Tong, 2001; Turney, 2002;Wiebe,2000).

In **2012, Federico Neri Carlo et al. [2]** had developed an idea of sentimental analysis using 1000 facebook posts about new casts, comparing the[4] sentiments for the Italian public broadcasting service - towards the emerging and more dynamic private companies.

In **2015, Xing fang et al presented an idea of sentiment analysis** using product review data which is collected from Amazon.com [7]. His main aim was to tackle the problem of sentiments polarity categorization of sentiments analysis [6].

### *A) Two approaches of Sentiment Analysis*

- **Supervised approaches or machine learning method:**

  Machine learning is one of the most prominent techniques gaining researchers interest [10] due to its adaptability and accuracy. This method comprises of three stages: (i) Data collection (ii) Pre-processing and (iii) Training data Classification [9].

- **Unsupervised (or lexicon-based)**
  Lexical analysis estimates the sentiment from the semantic orientation of words [8] or phrases that occur in a text. In this approach a dictionary containing positive and negative words that are matched with the words containing in tweet. However, these techniques totally depend on lexical resources [6] which are concerned with mapping words [7] to a categorical (positive, negative, neutral) or numerical sentiment score. In this method the unigrams, which are found in the lexicon [9] are assigned a polarity score.

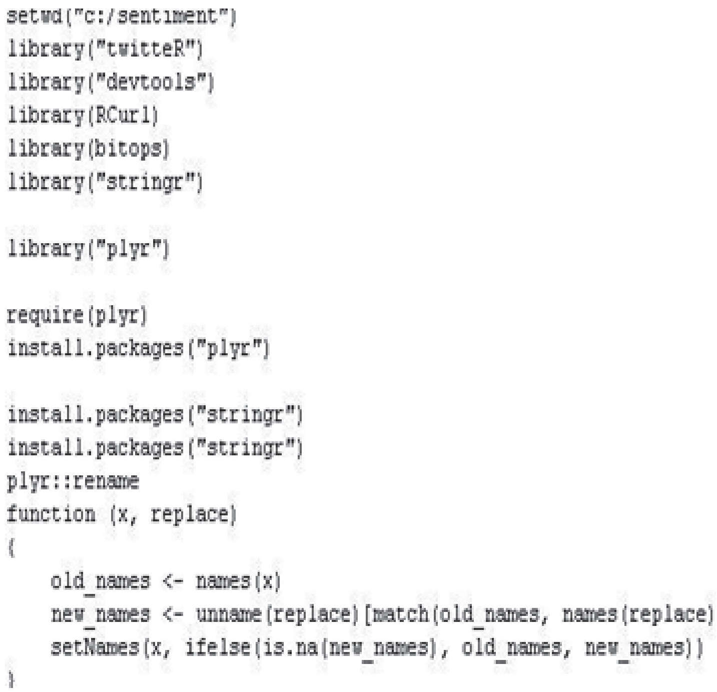
978-1-5090-4083-4/17/$31.00 ©2017 IEEE

## III) PROPOSED METHODOLOGY

Here we follow the following steps for sentiment analysis.

Step 1:

a) Creating a twitter application- For twitter sentimental analysis we have to create a twitter application. This application allows connecting the twitter server to crawl the data using the Twitter API [5].

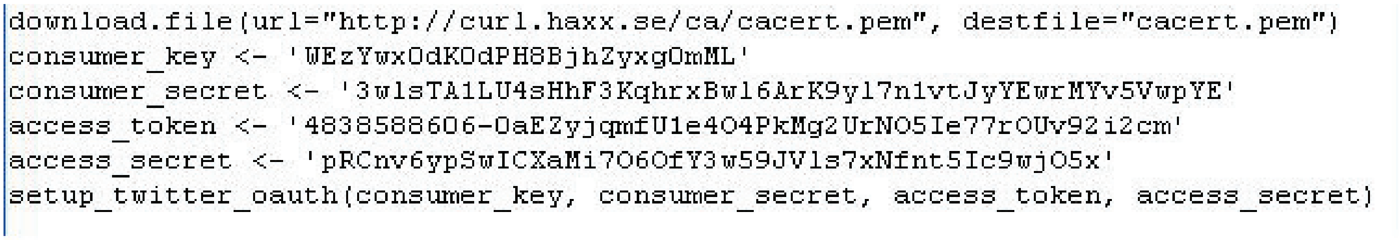


```
setwd("c:/sentiment")
library("twitteR")
library("devtools")
library(RCurl)
library(bitops)
library("stringr")

library("plyr")

require(plyr)
install.packages("plyr")

install.packages("stringr")
install.packages("stringr")
plyr::rename
function (x, replace)
{
    old_names <- names(x)
    new_names <- unname(replace)[match(old_names, names(replace)
    setNames(x, ifelse(is.na(new_names), old_names, new_names))
}
```

Fig1: Loading of R packages

b) Installing R packages- We installed twitteR, ROAuth, plyr, Stringr, ggplot2, tm etc. packages in the R environment, a sample code snippet is illustrated in Fig. 1.

c) Handshaking- This step is for accessing the Twitter API. This step includes the script code to perform handshaking using the Consumer Key and Consumer Secret of the application, the code snippet is illustrated in Fig. 2.



```
download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")
consumer_key <- 'WEzYwxOdKOdPH8BjhZyxgOmML'
consumer_secret <- '3wlsTA1LU4sHhF3KqhrxBwl6ArK9yl7n1vtJyYEwrMYv5VwpYE'
access_token <- '4838588606-OaEZyjqmfU1e4O4PkMg2UrNO5Ie77rOUv92i2cm'
access_secret <- 'pRCnv6ypSwICXaMi7O6OfY3w59JVls7xNfnt5Ic9wjO5x'
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

Fig2: Authentication and Handshaking with Twitter

**Step 2**: Collecting data from Twitter

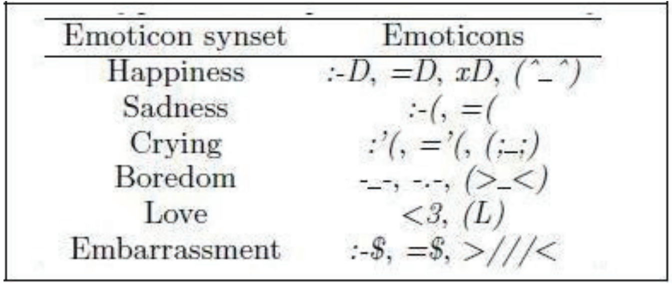
We collected tweets from twitter server by using Twitter API using different keywords within a particular time period and date.

**Step 3**: Data pre-processing:

After collecting tweets, the structure of a tweets data is analyzed properly for pre-processing of tweets data to eliminate all unwanted information. Pre-processing step is used for

(a) Removing punctuation, digit.
(b) Cleaning text: Remove non alphanumeric characters from the tweet.
(c) Removing URLs.
(d) Removing un-necessary white spaces and tabs, etc.
(e) Replacing emoticons: Emoticon is replaced by a word according to a lexicon of emoticons as shown in Table 1.

The meanings of the emoticons were taken from website[12].

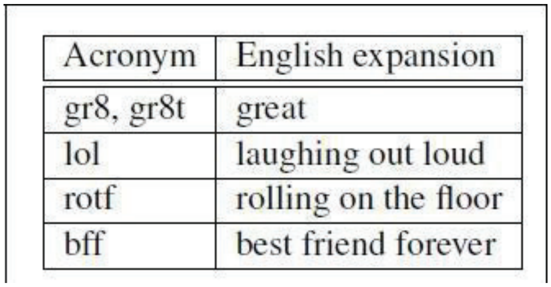


| Emoticon synset | Emoticons |
|---|---|
| Happiness | :-D, =D, xD, (^_^) |
| Sadness | :-(, =( |
| Crying | :'(, ='(, (;_;) |
| Boredom | -_-, -.-, (>_<) |
| Love | <3, (L) |
| Embarrassment | :-$, =$, >///< |

Table 1: Typical examples of emoticon synsets [3].

(f) **Stop-words removal**: Stop words are words which carry a connecting function in the sentence, such as prepositions, articles, etc.

(g) **Replacing acronyms:** Tweet containing acronym is replaced by proper word. The meanings of the acronyms were taken from website[11] and are illustrated in table 2.



| Acronym | English expansion |
|---|---|
| gr8, gr8t | great |
| lol | laughing out loud |
| rotf | rolling on the floor |
| bff | best friend forever |

Table 2: Typical examples of acronym and their expansion[3].

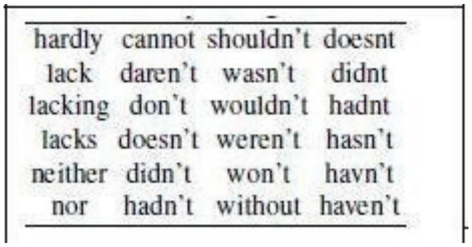
(h) **Finding target:** symbol "@" used by twitter to refer to user.

(i) **Finding Hash tags:"#"** used to mark topics and increase tweet visibility.

(j) **Negations Handling-** Negation word [Table 3] convert positive sentiment to negative or from negative to positive by using special words. If negation word was found with positive word, we decrement positive score by one and vice versa.

e.g. **I like iphone** □ **Positive**

**I don't like iphone** □ **Negative**



| | | | |
|---|---|---|---|
| hardly | cannot | shouldn't | doesnt |
| lack | daren't | wasn't | didnt |
| lacking | don't | wouldn't | hadnt |
| lacks | doesn't | weren't | hasn't |
| neither | didn't | won't | havn't |
| nor | hadn't | without | haven't |

Table 3: Typical examples of negation words[3].

(k) Changing tweets text data to lower case and splitting sentences to words

**Step 4: Classification**

Here we use lexical method for classification. We work on dictionary-based approach.

The dictionary-based approach depends on finding words from tweets, and then matches the word with the dictionary. If there is a positive match, the positive score is in cremented or the word is tagged as positive. If it is negative word then the negative score is incremented or the word is tagged as negative. Otherwise tag neutral word.

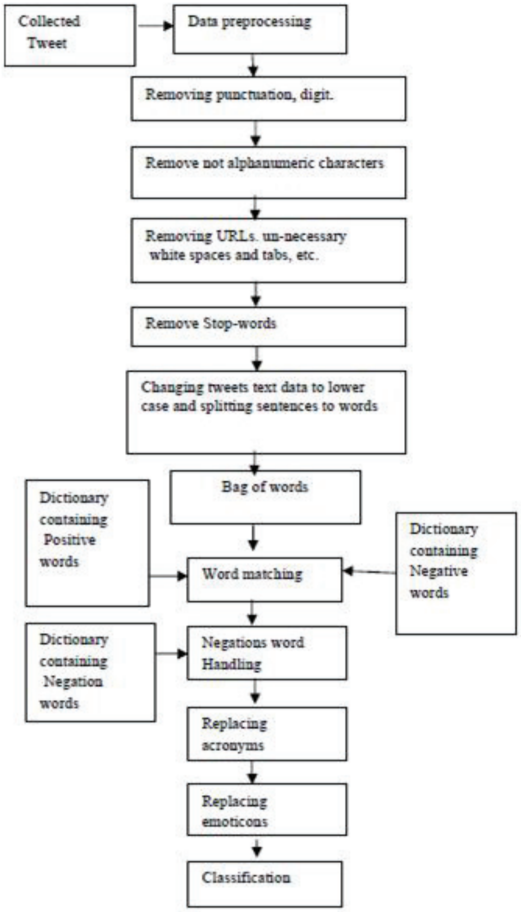
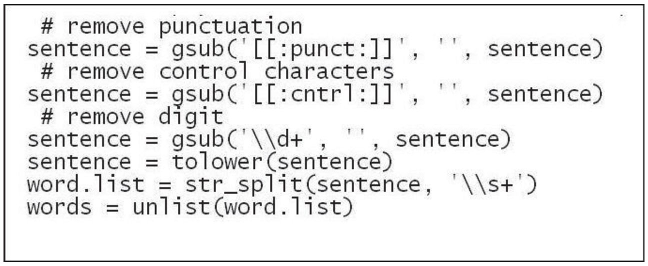

Collected Tweet
Data preprocessing
Removing punctuation, digit.
Remove not alphanumeric characters
Removing URLs. un-necessary white spaces and tabs, etc.
Remove Stop-words
Changing tweets text data to lower case and splitting sentences to words
Bag of words
Dictionary containing Positive words
Word matching
Dictionary containing Negative words
Dictionary containing Negation words
Negations word Handling
Replacing acronyms
Replacing emoticons
Classification

Fig. 3 Working of a lexical technique

## IV) PROPOSED ALGORITHM

- **Pre-processing of Input Tweets**:

    1) Delete punctuation marks (, @ [.? " // !)
    2) Delete hyperlink (http\\)
    3) Delete numbers, symbols
    4) Delete digits, tab,@
    5) Transformation to lower case



```
 # remove punctuation
sentence = gsub('[[:punct:]]', '', sentence)
 # remove control characters
sentence = gsub('[[:cntrl:]]', '', sentence)
 # remove digit
sentence = gsub('\\d+', '', sentence)
sentence = tolower(sentence)
word.list = str_split(sentence, '\\s+')
words = unlist(word.list)
```

- **Prerequisites:**

    1) File containing list of Positive Sentiment Words
    2) File containing list of Negative Sentiment Words
    3) File containing list of Negation Words
    4) File containing list of acronyms Words
    5) File containing list of emoticons with proper words

- **Algorithm Employed:**

```
Score = 0, Positive_Score=0, Negative_Score=0,
Negation_Score=0

/* Negations Handling */

 Match words with the dictionary containing negation words.
If Word= Negation word then

Match next words with the dictionary containing positive
sentiment words.
If Word== Positive Word then
Negative_Score = Negative Score +1

Else
Match next words with the dictionary containing negative
sentiment words
 If Word== Negative Word then
 Positive _Score = Positive_Score +1

 Else
 If Word== Positive Word then
  Positive _Score = Positive_Score +1

 If  Word==Negative Word then
 Negative_Score = Negative_Score +1

 /* Replacing acronyms */

If Word==Unknown Word  then

 Match words with the dictionary containing acronym words.
 If Word== acronym word, then

Replace acronym word with proper word.

Match words with the dictionary containing positive
sentiment words.
  If Word== Positive Word ,then

  Positive _Score = Positive_Score +1 Else
  Match words with the dictionary containing negative
sentiment words.
  If  Word==Negative Word, then

   Negative_Score = Negative_Score +1


   /* Detecting emoticon */

    If Word==Unknown Word,then
 Match words with the dictionary containing emoticon.

   If Word== emoticon,then
Replace emoticon word with proper word.

Match words with the dictionary containing positive
sentiment words

 If Word== Positive Word ,then Positive
_Score = Positive_Score +1 Else
Match words with the dictionary containing negative
sentiment words

 If  Word==Negative Word , then
 Negative_Score = Negative_Score +1

  /* Score calculation */

 Score= Positive_Score - Negative_Score

 If Score>0 Then print
```

```
"Positive"
 Else
 if Score < 0 then print
 "Negative"
 Else
 print "Neutral"
```

V) RESULT AND ANALYSIS

**Level of Sentiment analysis**

Sentiment analysis can take place mainly at three levels**. a) Document level b) Sentence level c) Aspect level**. Here we tried to analyze the tweets at document level and entity or aspect level. In document level, the task is to classify whether a whole opinion document expresses a positive or negative sentiment and in aspect level, task is aspect extraction and analyzes customer's feedback on the basis of that. We collected 3000 tweets using twitter

API from twitter with keyword "Iphone" from date 12.06.16 to 15.08.16 with the following aspects and performed sentiment analysis of twitter users.

**1) Voice Quality**
**2) Battery Quality**
**3) Service**
**4) Price**
**5) Picture Quality**
**6) Size**

On the basis of analysis we get the following result

| |
|---|
| Aspect: **GENERAL**<br>Positive: 2500 Negetive:300 |
| Aspect: **Battery life**<br>Positive: 190 Negative:90 Neutral :10 |
| Aspect: **Voice quality**<br>Positive: 200 Negative: 20 Neutral :20 |
| Aspect: **Service**<br>Positive: 250 Negative: 20 Neutral :5 |
| Aspect: **Size**<br>Positive: 100 Negative: 10 Neutral :20 |
| Aspect: **Picture Quality**<br>Positive: 250 Negative: 25 Neutral :6 |
| Aspect: **Price**<br>Positive: 30 Negative:140 Neutral :20 |

Table 4: Iphone:( An aspect-based opinion summary)

Now we summarize a set of reviews on Iphone. GENERAL represents the Iphone (the entity). 2500 reviews expressed positive opinions 300 expressed negative opinions. Voice quality and battery life, service, Size, Picture quality, Price are Iphone aspects. Reviews expressed positive opinions negative opinions and neutral and are illustrated in Table 4.

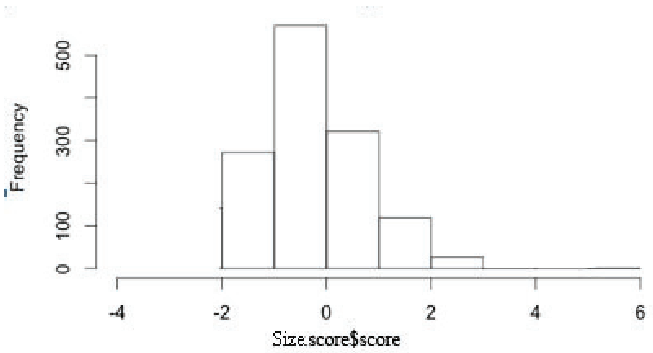


Fig 4: Histogram with entity Size

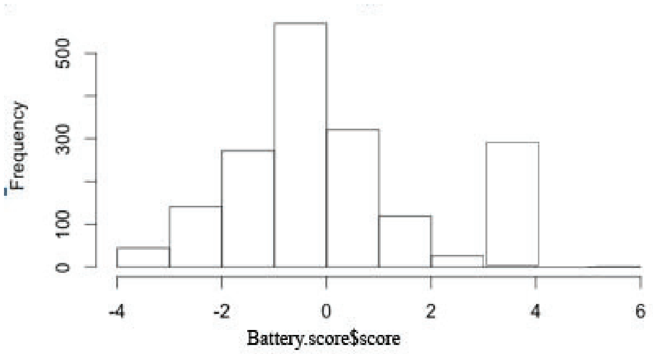


Fig 5: Histogram with entity Battery

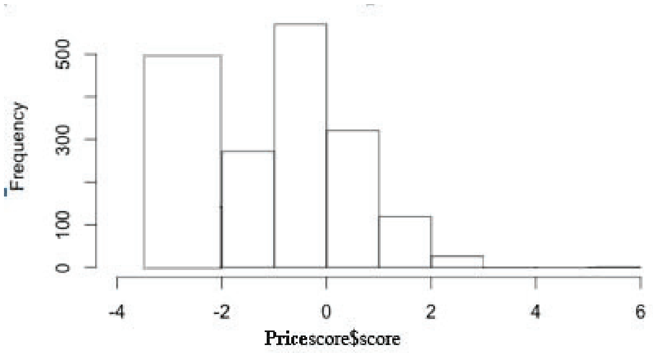


Fig 6: Histogram with entity Price

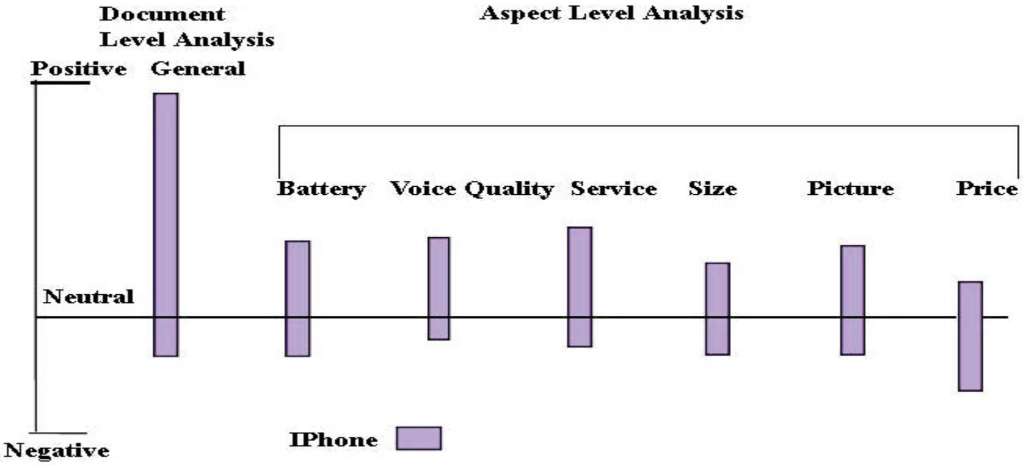


Fig 7.Visualization of aspect-based summary of opinions on Iphone[2]

Based on the above discussions, we can define a model of entity and opinion document and aggregate opinion. Let a sentence $s$ contains a set of entities $\{e_1, e_2, \ldots, e_r\}$ and a subset of their aspects $\{a_1, \ldots, a_m\}$ from a set of opinion holders[2] $\{h_1, h_2, \ldots, hp\}$ at some particular time point and a set of sentiment words or phrases $\{sw_1, \ldots, sw_n\}$ with their sentiment scores . The sentiment orientation for each aspect $a_i$ in $s$ is determined by the following aggregation function:

$$score(a_i, s) = \sum_{ow_j \in s} \frac{sw_j.so}{dist(sw_j, a_i)},$$

Where $dist(sw_j, a_i)$ is the distance between aspect $ai$ and sentiment word $sw_j$ in $s$. $sw_j.so$ is the sentiment score of $sw_i$. If the final score is positive, then the opinion on aspect $a_i$ in $s$ is positive. If the final score is negative, then the sentiment on the aspect is negative. It is neutral otherwise.

This method is applicable in other domain also. Eg. Customer review related to airlines services, Political review etc.

## VI) CONCLUSION AND FUTURE WORK

In this paper, we have tried to implement dictionary based methodology of sentiment analysis and developed an algorithm that is employed to large amount of data to estimate the sentiment of public.

We replaced acronym word by making acronym dictionary and also detected emoticons in the tweet. We have done both document level and aspect level analysis based on the proposed methodology, which helped in decision making. In future, we will work with comparative opinion and try to work with machine learning approaches and develop a hybrid working model for sentiment analysis.

## REFERENCES

[1] Olga Kolchyna1, Th´arsis T. P. Souza1, Philip C. Treleaven12 and Tomaso AsteTwitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination , Department of Computer Science, UCL, Gower Street, London, UK.

[2] Bing Liu, Sentiment Analysis and Opinion Mining, April 22, 2012

[3] Walaa Medhat , Ahmed Hassan , Hoda Korashy . Sentiment analysis algorithms and applications:A survey a School of Electronic Engineering, Canadian International College, Cairo Campus of CBU, Egypt Ain Shams University, Faculty of Engineering, Computers & Systems Department, Egypt,Received 8 September 2013; revised 8 April 2014; accepted 19 April 2014

[4] Harsh Thakkar and Dhiren Patel ,**Approaches for Sentiment Analysis on Twitter: A State-of-Art study ,** Department of Computer Engineering, National Institute of Technology, Surat-395007, India.

[5] Apoorv Agarwal Boyi Xie Ilia Vovsha Owen Rambow Rebecca Passonneau,, Sentiment Analysis of Twitter Data , Department of Computer Science ,Columbia University.New York, NY 10027 USA

[6] Cataldo Musto, Giovanni Semeraro, Marco Polignano, A comparison of Lexicon-based approachesfor Sentiment Analysis of microblog posts, Department of Computer Science,University of Bari Aldo Moro, Italy

[7] James Spencer and Gulden Uchyigit,Sentimentor: Sentiment Analysis of Twitter Data.School of Computing, Engineering and Mathematics.University of Brighton

[8] Anna Jurek,Maurice D. Mulvenna and Yaxin Bi,Improved lexicon-based sentiment analysis for social media analytics Science direct,Published: 9 December 2015

[9] Huifeng Tang, Songbo Tan *, Xueqi Cheng, A survey on sentiment detection of reviews, Information Security Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, PR China

[10] Shruti Wakade, Chandra Shekar, Kathy J. Liszka and Chien-Chung Chan, Text Mining for Sentiment Analysis of Twitter Data,
Department of Computer Science , The University of Akron

[11] http://www.acronymfinder.com/

[12] http://en.wikipedia.org/wiki/List_of_emoticons.