www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11315-11321

Twitter Sentiment Analysis with Emoticons

Waghode Poonam B,Prof. Mayura Kinikar
MIT Academy of Engineering, Alandi, Pune, Maharashtra, India.

Abstract: Sentiment analysis, which is also called opinion mining is the field of study which analyzes people's opinions, sentiments, evaluations, appraisals, attributes and emotions towards entities such as products services, organizations, individuals, issues, events, topics, and their attributes through twitter. People use micro-blogging (twitter) to talk about their daily activities and to seek or share information. It is an online social networking and micro-blogging service that enables users to send and read "tweets", which are text messages limited to 140 characters. In this paper we propose a model that can spot the public opinion with their emotions.

Keywords—Sentiment analysis, tweets, Social network, Lexicon-Based Classifier

I. INTRODUCTION

ON Web 2.0, content which is user-generated, that is material submitted by users who connected with social network sites, is a major subject. Twitter messages that are very short. It is convenient for research because there are a very large number of messages, many tweets are openly available for user, and obtaining them is technically simple compared to huge number of blogs from the web.

Sentiment analysis is a technique for extracting sentiment associated with polarities of positivity, negativity and neutrality. It is one of the types of natural language processing in which we can track the mood of the public about a particular entity. Sentiment analysis, which is also called opinion mining, is used for constructing a system to collect and examine opinions about the entities made in tweets. Due to the explosion of social media services present a great opportunity to understand the sentiment of the public via analyzing its large-scale and opinion-rich data. In social media, it is easy to amass vast quantities of unlabeled data, but very costly to obtain sentiment labels, which makes unsupervised sentiment analysis essential for various applications. It is challenging for traditional lexicon-based unsupervised methods due to the fact that expressions in social media are unstructured, informal, and fast-evolving. A traditional way to perform unsupervised sentiment analysis is the lexicon-based method [2]. These methods employ a sentiment lexicon to determine overall sentiment polarity of a document.

The proposed system will identify the polarity of tweets by sentiment analysis. The proposed model will find out the polarity of one entity as well as of two different entities and based on the polarities the entities will be compared.The remainder of the paper is organized as follows. Section 2 provides a brief review of Literature Survey. Section 3 illustrates the sentiment analysis processing framework and technique. The implementation details are reported in Section 4.

II. RELATED WORK

Many sentiment analysis approaches has been used far for text type of tweets for twitter,But Emoticons has expressed the sentiment with the strong and real feeling.In existing work[10],Pang and Lee provide a general overview of the field . Thelwall et al. [11] propose SentiStrength to detect the strength of sentiment in short, informal exchanges in social media, with a focus on MySpace comments.[13]Neviarouskaya et al. presented the Affect Analysis Model (AAM) in order to address the issue of recognizing emotion in text messaging. The approach is based on WordNet-Affect [14]. The overall aim is to predict the nature of emotional content in several different categories as defined by Izard [15], such as anger, disgust, fear, and guilt. Nonetheless, in comparison to the proposed work more accuracy is maintained as different database and varieties of emoticons are used which gives more accuracy in the emotional feelings context.

III. SENTIMENT ANALYSIS OVERVIEW

Given a set of tweets, T , which contains a set of sentences, s, T = {s1 , s2 , … , si}; and each sentence sk describes something on a subset of entities e = {ei , ..., ej|ei , ej Ɛ E}, where E is the set of all entities. An entity can be a person, an organization, a location, a product, etc. Each sentence also contains a set of opinion word, w, s = {w1, w2,… ,wj}. At first, a Sentence Sentiment Scoring Function(SSSF) is used to determine the orientation of sentiment expressed on each entity ei in s (i.e., the pair of (ei , s)). Then an Entity Sentiment Aggregation Function (ESAF) is used to obtain the total sentiment scores for an given entity ei.

1) *Sentence Sentiment Scoring Function:* in this stage, the classification algorithm detects all words that belong to Wilson lexicon list and extracts their polarity. Adjectives are good indicators of sentiment and have been used as features for sentiment classification by a number of researchers [8],[5].

However, it does not necessarily imply that other parts of speech do not contribute to expressions of opinion or sentiment. In fact, nouns (e.g., “gem”) and verbs (e.g., “love”) can be strong indicators for sentiment. Therefore, in this study, we use all the parts of speech. We summed up the semantic orientation score of the opinion words in the sentence to determine the orientation of the opinion sentence. The score function for a sentence is as follow:

$$score(s) = \sum_{w_j : w_j \in s \wedge w_j \in WL} \frac{w_j \cdot sentOri}{dis(w_j, e_i)}. \qquad (1)$$

where wj is an opinion word, WL is the set of all opinion words from Wilson lexicon list and s is the sentence that contains the entity ei , and dis (wj , ei) is the distance between entity ei and opinion word wj in the sentence s, and wj.sentOri is the semantic orientation of the word wj (i.e.,+1, or +0.5, or 0, or −1, or −0.5). If a sentence contains more than one entity then the opinion word close to the entity has smaller value of dis (wj , ei) and indicates this word makes more contribution to that entity’s sentiment scores. The scores(s) is normalized by the number of the opinion words, n, in the sentence to reflect the sentiment scores distributions of opinion words. So, normalized sentiment score will be:

$$Score(s)N= Score(s)/n \qquad (2)$$

2) *Entity Sentiment Aggregation Function:* In the given set of tweets, an entity appears in the set of sentences s = {s1, s2,..., si}. We use co-occurrence of an entity and a sentiment word in the same sentence to mean that the sentiment is associated with that entity. This is not always accurate, particularly in complex sentences [2]. Still the volume of text we process enables us to generate accurate sentiment scores.

The Mathematical Model of the system will be given as-
Score={wj,ei,WL,SentOri,s}where,

- wj=Opinion word
- ei=Entity
- SentiOri=Semantic Orientation
- WL=Wordnet List
- s=Sentence that contain an entity

For a given entity ei, which may appear in multiple sentences {s1, s2, ..., si}, the normalized sentiment score for this entity in a sentence sk is score(ei , sk)N. The total sentiment scores of this entity will be aggregated by Entity Sentiment Aggregation Function that is depicted as below:

$$score(e_i) = \sum_{(s_k : s_k \in s)} score(s_k)_N \qquad (3)$$

This score is normalized by the number of the sentences, m, and then the final sentiment score for an entity will ranges in the interval [+1, −1].

$$score(e_i)_N = score(e_i)/m \qquad (4)$$

In regard to sentiment intensity (or strength) for a given entity, ei, appears in the sentences, the following heuristic rule is applied:

$$intensity(e_i) = \begin{cases} \text{SP} & \text{if } (+0.5 < score(e_i)_N < +1) \\ \text{P} & \text{if } (0 < score(e_i)_N < +0.5) \\ \text{Neu} & \text{if } (score(e_i)_N = 0) \\ \text{Neg} & \text{if } (-0.5 < score(e_i)_N < 0) \\ \text{SN} & \text{if } (-1 < score(e_i)_N < -0.5) \end{cases}$$

a. SN (Strong Negative) Sentences about the entity ei contain purely negative words or phrases or only allowed a slightly positive word.
b. N (Negative) Sentences contain mainly negative phrases and words. There may be a few positive words, but the negative words or phrases outweigh the positive ones.
c. Neu (Neutral) Sentences have a mediocre or balanced sentiment. The positive and negative words or phrases seem to balance each other, or it is neither positive nor negative overall. Even if there are more negative phrases, the positive ones use a stronger language than the negative ones.
d. P (Positive) Sentences have mainly positive terms. There may be some negative ones; however, the positive ones are stronger and outweigh the negative ones.
e. SP (Strong Positive) Sentences have purely positive words expressing strong affirmative feelings with no complaints. It may have the smallest negative words, but the sentence has mostly great-sounding words or phrases.

IV. PROPOSED SYSTEM AND IMPLEMENTATION DETAILS

The proposed system consists of three modules:

• Feature selection module is build for extracting the opinionated words from each sentence.
• Sentiment identification module that associates expressed opinions with each relevant entity in each sentence level.
• Sentiment aggregation and scoring module is build for calculating the sentiment scores for each entity. In our proposed system, firstly the tweets are taken from the twitter in database and then for each sentence in tweet, POS tagging and stemming is performed. A Part-Of-Speech Tagger (POS Tagger) is

a software package that reads text and assigns parts of speech tags to each word, such as noun, verb, adjective, etc. In this paper we focus on five POS tags: NN, JJ, DT, NNS and VBG, for nouns, adjectives,

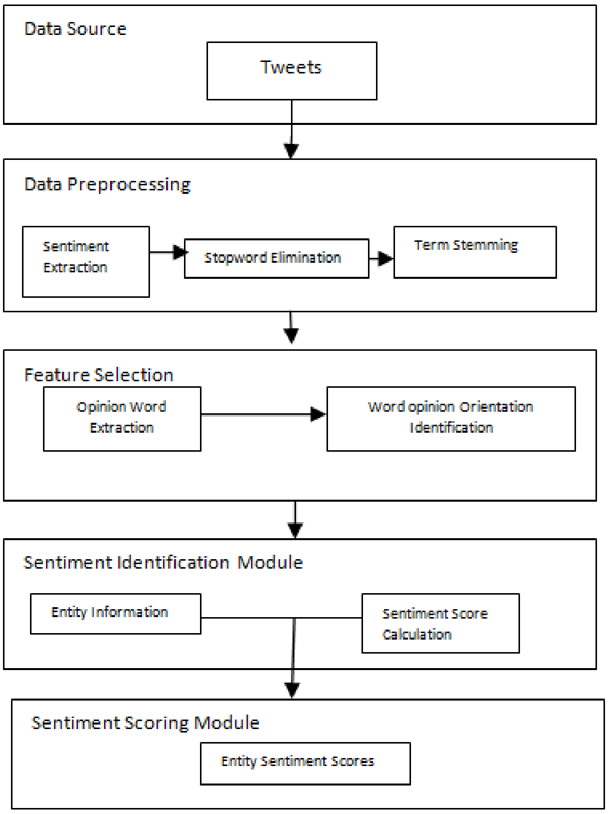

Figure 1: Modules of proposed system

determiners, plural nouns and verb gerunds respectively [13]. Stemming is used to select one single form of a word instead of different forms. The goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. In this work we use the Stanford software package for both POS tagging and stemming.A typical tweet contains word variations, emoticons, hashtags etc. The objective of the preprocessing step is to normalize the text into an appropriate form to extract the sentiments. Below are the preprocessing steps used:

POS Tagging: POS Tagger gives part of speech tag associated with words. POS tagging is done using NLTK.

Stemming: Stemmer gives the stem word, non stem words are stemmed and replaced with stem words. For example, words like 'loved', 'loves', 'loving', 'love' are replaced with 'lov'. This would aid the engine to do the word match from the text to the lexicon. Stemming is done using NLTK.

Exaggerated word shortening: Words which have same letter more than two times and not present in the lexicon are reduced to the word with the repeating letter occurring just once. For example, the exaggerated word "NOOOOOO" is reduced to "NO".

Emoticon detection: Emoticon has some sentiment associated with it. Twitter NLP is used to extract emoticons along with the sentiments in the Twitter data.[14]

Hashtag detection: The hashtag is a topic or a keyword that is marked with a tweet. Hashtag is a phrase starting with # with no space between them. Hashtags are identified and sentiments are extracted from them.

Stop Words: All the stop words (like a, an, the, is etc.) and discourse connectives are discarded.

In Figure 2 we have shown the workflow of our model, in which XML, JavaScript,Java, Servlets, JSON are used and for database Twitter Developers' API are used. Tools & Software: Windows 7, Eclipse, Twitter REST API are used.

In the propose system, the algorithm detects all words that belong to the emotional dictionary and extracts their polarity and intensity. We modify the initial term scores with additional, prose-driven functionalities such as: negation detection (e.g., "good" versus "not good"), capitalization detection (e.g., "bad" versus "BAD"), exclamation and emoticon detection (e.g., "happy!!" and ":-)") intensifiers (e.g., "liked" versus "liked very much") and diminishers (e.g., "excellent" versus "rather excellent"), and finally orientation is calculated the polarity of the entity is calculated among two also by which an user can make an opinion with generic domain .Below is workflow of the propose system.

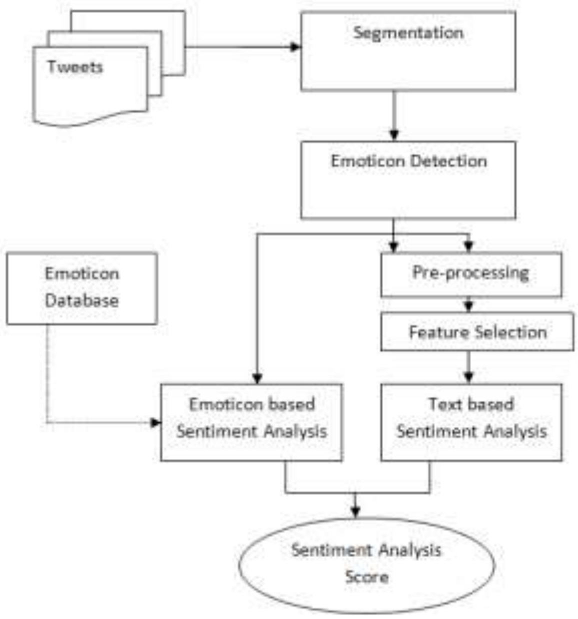

Figure 2: Workflow of model

V. LEXICON-BASED CLASSIFICATION

The proposed classifier is a typical example of an unsupervised approach, because it can function without any reference corpus and does not require any training. The classifier is based on estimating the intensity of negative and positive emotion in text, that is, the output of the classifier is one of {0, +1, −1}. The notion that both negative and positive emotion is present in a text may seem somewhat peculiar. The level of valence in each scale is measured in two independent ratings {Cpos, Cneg}; one for the positive dimension (Cpos =
{1, 2 . . . , 5}) and one for the negative (Cneg = {−1, . . . ,−5}), where higher absolute values indicate stronger emotion and values {1,−1} indicate lack of (i.e., objective text). For example, a score like {+3, −1} would indicate the presence of only positive emotion, {+1,−4} would indicate the presence of (quite strong) negative emotion and {+4,−5} would indicate the presence of both negative and positive emotion. For
example, the sentence "I hate the fact that I missed the bus, but at least I am glad I made it on time :-)" expresses both negative and positive emotion, where the latter is considered dominant. We solve conflicts of equality (e.g., {+3, −3}) by taking into consideration the number of positive and negative tokens and giving preference to the class with the largest number of tokens. A document is classified as objective if its scores are {+1,−1}. Note that the {Cpos, Cneg} ratings are only used as an intermediate step in making the final prediction. The algorithm is based on the emotional dictionary from the "Affective Norms for English words" and which was derived from a number of psychological studies and maintains an extensive dictionary list along with human assigned emotional categories for each lemma. Given a document d, the algorithm detects all words that belong to the emotional dictionary and extracts their polarity and intensity. We modify the initial term scores with additional, prose-driven functionalities such as: negation detection (e.g., "good" versus "not good"), capitalization detection (e.g., "bad" versus "BAD"), punctuation and emoticon detection (e.g., "happy!!" and ":-)") intensifiers (e.g., "liked" versus "liked very much") and diminishers (e.g., "excellent" versus "rather excellent"), to produce the final document scores and manually checked for duplicates and conflicts. The modules function in the following way: the neighborhood of every word that is present in the text and belongs to the ANEW lexicon is scanned for "special" terms, such as negators (e.g., "not") intensifiers (e.g., "very") or diminishers (e.g., "little"). The reason where the detection is carried out because of the vast majority of informal textual communication contains significant spelling errors, making any such attempt very difficult and additionally seriously limiting the domain of applicability of the proposed solution. If an intensifier or diminisher word is found, then the original emotional value of the word is modified by the respective modifier score which is either added or subtracted (in case of a diminisher) to the absolute value of the term. For example, if "bad" has an initial value of −3 then "very bad" would be modified to -4. Similarly, "somewhat good" would be judged as +2, taking into consideration that "good" has an original value of +3. If a negation term is found then the absolute value of the emotional term is decreased by 1 and its polarity is reversed. For example "not bad" would be +2. The intuition behind the reduction by one (instead of a simpler reversal of signs) is that although the polarity of a term is reversed with the usage of negation, the full original emotional weight of a term (such as "bad" in the given example) is not fully transferred to the other class and thus the reduction by one. Simply put, one does not typically use "not bad" if one means "good." Last, for the capitalization detection module, if a word, larger than two characters (in order to avoid false positives caused by normal article capitalization after a fullstop), that is written fully in capital letters is detected within the neighborhood of an emotional word, including the actual emotional word, then the weight of the word is modified in the same manner as if an intensifier with a weight of 1 was detected. The exclamation detection module functions in the same manner. In contrast, emoticons are considered as explicit indicators of expression.That are assigned specific weights, that is, +3 for positive emoticons and -3 for negative.The score of a document on the Cpos and Cneg scales is the maximum positive and negative number produced, respectively. As previously stated, for binary positive/ negative prediction the class with the highest absolute value is considered dominant. Algorithm 1 presents the full details of the classifier in pseudocode.

Algorithm 1: Lexicon-Based Classifier

INPUT: Affective Dictionary Wordnet
INPUT: AbbrevationList, NegationList, IntensifierList
INPUT: ExclamationList, DiminisherList, EmoticonList
INPUT: Document d = {w1,w2 ...wn} to be classified
Initialize Cpos ← +1, Cneg ←−1, PosInstances← 0, NegInstances← 0
for all wi ∈ d do
if wi ∈ AbbrevationList then wi← FullForm(wi)
endif
if wi ∈ Wordnet ∪ EmoticonList then
tempwi<-EmotWeight(wi)

if EmotWeight < = 5
then tempwi= Strong positive(+5)
else tempwi=LOL
endif
if EmotWeight=> 5
Then tempwi= Strong Negative(-5)
endif

```
if wk ∈ NegationList then
if tempwi<0 then
tempwi←−tempwi − 1
else
tempwi←−tempwi + 1
    end if
 end if
if wk ∈ IntensifierList then
   if  tempwi<0 then
tempwi← tempwi−IntenseWeight(wk)
else
     tempwi←−tempwi +IntenseWeight(wk)
             end if
       end if
if wk.length ≥ 2 AND wk =ALLCAPITALS then if
tempwi ≤ 0 then
tempwi ← tempwi− 1
else
tempwi←−tempwi + 1
end if end if
if wk ∈ DiminisherList then
    if tempwi<0 then

tempwi ← tempwi + DiminishWeight(wk)
else
tempwi←−tempwi− DiminishWeight(wk)
   end if
end if
if wk ∈ PunctuationList then

   tempwi← tempwi− 1
else
    tempwi← tempwi + 1
     end if
          end if
       end for
 end if
if tempwi  > 5 then
tempwi ← 5
end if
if tempwi < −5 then tempwi←−5
end if
if tempwi > 0 then
PosInstances← PosInstances+ 1
else
NegInstances← NegInstances+ 1
endif
if tempwi > 0 AND tempwi > Cpos then
Cpos ← tempwi
end if
if tempwi < 0 AND tempwi < Cneg then
Cneg ← tempwi
end if end for
if Cpos = Cneg = 1 then return objective else if Cpos
>Cneg then return positive else if Cpos <  Cneg then
return negative
else if Cpos =  Cneg then
if PosInstances > NegInstances then return positive
else if PosInstances < NegInstances then return
negative else if PosInstances = NegInstances then
return objective      end if
end if
```

In the proposed system Wordnet list of words, which readily provide emotional weights for token on 1-9 scales, to the list of utilized lexicon used by the algorithm. The proposed system will be capable of predicting capabilities of the algorithm to wordnet word list, thus providing more emotionally comprehensive analysis of textual communication.

VI RESULT ANALYSIS

The result of the proposed solution is evaluated by comparison of the Accuracy of Sentiment Analysis. Accuracy of the Sentiment Analysis is identified using below formulae:

$$\text{Accuracy} = \frac{\text{Total Number of True Sentiments}}{\text{Total Number of Tweets}}$$

As shown in the graph below the sentiments of Tweets without considering polarities of Emoticons.

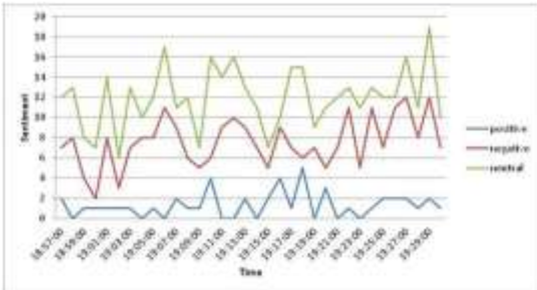

Figure 1: Sentiment Analysis without Emoticons

On the other hand, when Emoticon Sentiment Analysis is employed, the graph below shows sentiments and difference in the opinion.

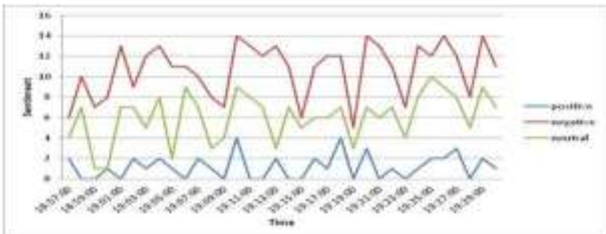

Figure 2: Sentiment Analysis with Emoticons

Calculating accuracy, there is need to predict which ones are positive ,negative and neutral and when I get the actual results it will be sum up the times I was right or wrong. There are four ways of being right or wrong:

- TN / True Negative: case was negative and predicted negative
- TP / True Positive: case was positive and predicted positive
- FN / False Negative: case was positive but predicted negative
- FP / False Positive: case was negative but predicted positive
- TNL / True Neutral: case was negative and predicted neutral

- FNL / False Neutral: case was false and predicted neutral

- Accuracy= $\frac{(TN+TP+TNL)}{(TN+TP+FP+FN+TNL+FNL)}$

The graph below compares the Accuracy (in percentage) between Sentiment Analysis without Emoticon Polarities and with Emoticon Polarities. The below formulae is used to arrive at the accuracy:

- Accuracy = Total Number of True Sentiments / Total Number of Tweets

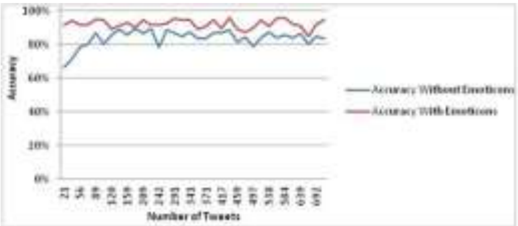

Figure 3: Accuracy

ACKNOWLEDGEMENT

I thank Mrs. Mayura Kinikar for guiding and helping for the project information and valuable comments. She helped in a broad range of issues from giving me direction, helping to find the solutions, outlining the requirements and always having the time to see me.
I have furthermore to thank Prof. R. M. Goudar, M.E. Coordinator Department of Computer Engineering, to encourage me to go ahead and for continuous guidance. I also want to thank Prof. Uma Nagraj for all her assistance and valuable guidance.

VII. REFERENCES

[1] Myriam Munezero, Calkin Suero Montero,Erkki Sutinen, and John Pajunen, "*Are They Different? Affect, Feeling, Emotion,Sentiment, and Opinion Detection in Text*", IEEE transactions on affective computing, pp. 101-111 VOL. 5, NO. 2, APRIL-JUNE 2014

[2] Shulong Tan, Yang Li, Huan Sun, Ziyu Guan, Xifeng Yan, Jiajun Bu,Chun Chen,and Xiaofei He, "Interpreting the Public Sentiment Variations on Twitter", IEEE transactions on knowledge and data engineering,pp. 1158-1171, VOL. 26, NO. 5, MAY 2014

[3] Georgios Paltoglou, Mathias Theunis, Arvid Kappas, and Mike Thelwall, "Predicting Emotional Responses to Long Informal Text", IEEE transactions on affective computing,pp.106-116 vol. 4, no. 1, january-march 2013

[4] Changbo Wang, Zhao Xiao, Yuhua Liu, Yanru Xu, Aoying Zhou, and Kang Zhang, "SentiView: Sentiment Analysis and Visualization for Internet Popular Topics" ,IEEE transactions on human-machine systems,pp.620-631,vol. 43, no. 6, november 2013

[5] Alena Neviarouskaya and Masaki Aono, "Sentiment Word Relations with Affect,Judgment, and Appreciation",IEEE transactions on affective computing,pp. 425-439 vol. 4, no. 4, october-december 2013

[6] Alexander Hogenboom, Daniella Bal, Flavius rasincar, Malissa Bal, Franciska de Jong,Uzay Kaymak, *"Exploiting emoticons in sentiment analysis*". In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Pp. 703-710, 2013

[7] Weichselbraun, A. ; Gindl, S. ; Scharl, "Extracting and Grounding Contextualized Sentiment Lexicons",In Intelligent Systems, IEEE Trans.,Vol.28, No. 2, pp: 39 - 46,March-April 2013

[8] Geor gios Paltoglou and Mike Thelwall. "*Twitter,MySpace,Digg: Unsupervised Sentiment Analysis in Social Media"* ACM Transactions on Intelligent Systems and Technology, Vol. 3, No. 4, Article 66, Publication date: September 2012.

[9] Xujuan Zhou,Xiaohui Tao,Jianming Yong and Zhenyu Yang. "*Sentiment Analysis on Tweets for Social Events*". In Proceedings of the IEEE 17th International Conference on Computer Supported Cooperative Work in Design ,pp. 557-562,2013.

[10] S. Brody and N. Elhadad. "*An unsupervised aspect-sentiment model for online reviews*". In HLT '10: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 804-812, Morristown, NJ, USA, 2010. Association for Computational Linguistics.

[11] K. Dave, S. Lawrence, and D. Pennock. "*Opinion extraction and semantic classification of product reviews*". In Proceedings of the 12th International World Wide Web Conference (WWW), pp. 519-528, 2003

[12] E.Schonfeld. Mining the thought stream. TechCrunch Weblog Article, *http://techcrunch.com/2009/02/15/mining-the-thought- stream//, 2009.*

[13] V. Hatzivassiloglou and J. M.Wiebe. "*Effects of adjective orientation and gradability on sentence subjectivity*". pp. 299-305, 2000.

[14] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. "*Twitter power: Tweets as electronic word of mouth*". J. Am. Soc. Inf. Sci., 60(11):2169-2188, 2009.

[15] M. M. Bradley and P. J. Lang. "*Affective norms for english words (anew): Stimuli,instruction manual, and affective rating*"s. Technical report, Center for Research in Psychophysiology, University of Florida, Gainesville, Florida, 1999.

[16] J. Kamps, M. Marx, R. Mokken, and M. de Rijke. "*UsingWordNet to measure semantic orientation of adjectives*". In Proceedings of the 4th International Conference on Language Resources and Evaluation, pages 1115-1118, 2004.

[17] B. Liu. "*Handbook of Natural Language Processing, chapter Sentiment Analysis and Subjectivity*". Second edition, 2010.

[18] S. Milstein, A. Chowdhury, G. Hochmuth, B. Lorica, and R. Magoulas. "*Twitter and the micro-messaging revolution*": Communication, connections, and immediacy140 characters at a time. An OReilly Radar Report. 54 pp., November 2008.

[19] B. Pang and L. Lee. "*Thumbs up? sentiment classification using machine learning techniques*". In Proceedings of EMNLP, pp. 79-86, 2002.

[20] J.Wiebe. "Learning subjective adjectives from corpora".In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pages735-740. AAAI Press, 2000.

[21] J.Wiebe and E. Riloff. "Creating subjective and objective sentence classifiers from unannotated texts". In CICLing2005, pages 486-497,2005.

[22] D. T. Wijaya and S. Bressan. *"A random walk on the red carpet: rating movies with user reviews and page rank".* In CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management, pages 951-960. ACM, 2008.

[23] T. Wilson, J. Wiebe, and P. Hoffmann. *"Recognizing contextual polarity in phrase-level sentiment analysis".* In HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 347-354, Morristown, NJ, USA, 2005.

[24] W. Zhang, C. Yu, and W. Meng. *"Opinion retrieval from blogs".* In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, CIKM '07, pages 831-840. ACM, 2007.

[25] Xiaowen Ding, Bing Liu, and Philip S. Yu. "*A Holistic Lexicon-Based Approach to Opinion Mining".* WSDM'08, CIKM '07, pages 831-840,Palo Alto, California, USA.February 11-12, 2008.

Waghode Poonam B has received B.E[Computer Engineering] from University of Pune in 2010 & pursuing M.E[Computer Engineering] From M.I.T, AOE, Alandi. Her research interest includes Data Mining

Mayura.U.Kinikar has received M.E [Comp Science & Engineering] from Government College of Engineering Aurangabad. Her research interest includes Data Mining.She is working as Asst. Prof. at M.I.T, AOE, Alandi, Pune.