Text Analysis for Product Reviews for Sentiment Analysis using NLP Methods

Text Analysis for Product Reviews for Sentiment Analysis using NLP Methods

[1] S.Muthukumaran, [2] Dr.P.Suresh

Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore, Tamilnadu, India.
Research Supervisor, HOD, Dept. of Computer Science, Salem Sowdeswari College, Salem, Tamilnadu, India

Abstract

This paper explains different methods for sentiment analysis and showcases an efficient methodology. It also highlights the importance the product reviews are of utmost importance for the buyers to decide based on their concerns regarding product's various aspects for example a monitor, processor speed, memory etc. Hence this sentiment analysis of product review provides nearly accurate statistics regarding a product, providing an ease to the customers for analyzing the product and zero down his/her search for an online product. The key focus here is efficient feature extraction, polarity classification thereby summarizing positive and negative or neutral polarity. The proposed work is able to collect information from various sites and perform a sentiment analysis of a user reviews based on that information to rank a product. Also these reviews suffer from spammed reviews from unauthenticated users. In this paper, we can show that the statistical methods are often combined with traditional linguistic rules and representations. In view of these facts, we argue that the Naive Bayes classification model and Hidden Markov Models is applied to analyze the polarity of the sentiment on online product reviews due to its computational simplicity and stochastic robustness.

Keywords: *Sentiment analysis (SA), Opinion mining, Machine learning, Naive Bayes (NB), Support Vector Machine (SVM), online reviews, Hidden Markov (HMM).*

I. INTRODUCTION

Sentiment Analysis, or Opinion Mining, describes a Natural Language Processing problem that attempts to differentiate opinionated text from factual text and, once text is assumed to be opinionated, classify it as expressing a negative opinion, a neutral opinion, or a positive one. Another sub problem that closely relates to Sentiment Analysis is that of representing opinionated content in a comprehensive manner.

Opinion Mining and Sentiment Analysis, an extension of data mining, is a natural language processing and text analytic technique [9] that determines people's emotion or feeling or attitude towards some topic by processing huge unstructured internet content. Opinion Mining extracts the sentiments or opinions bearing words present in the free text while Sentiment analysis determines sentiment polarity, whether positive or negative or neutral by analyzing each opinionated word or phrase. Sentiment Analysis summarizes the opinion of a writer or speaker about a particular topic and it can be done at word or aspect level, sentence level and document level. Sentiment can be classified by various ways: supervised classification techniques, unsupervised classification techniques and hybrid classification techniques or by combining the above two approaches. We *present* Sentiment Analyzer (SA) that extracts sentiment (or opinion) about a subject from online text documents.

Instead of classifying the sentiment of an entire document about a subject, SA detects all references to the given subject, and determines sentiment in each of the references using natural language processing (NLP) techniques. Our

Sentiment analysis consists of 1) a topic specific feature term extraction, 2) sentiment extraction, and 3) (subject, sentiment) association by relationship analysis. SA utilizes two linguistic resources for the analysis: the sentiment lexicon and the sentiment pattern database

II. RELATED WORK

Opinion Mining and Sentiment Analysis involves extraction of sentiment words from user reviews and automatic classification and summarization of sentiments. The sentiment words present in the free text can be identified by considering the following: adjectives or adverbs [14], uni-grams [15] or n-grams [16] with their frequency of occurrence, the POS (parts of speech) tagging of words [17] the negation of words [18]. The automatic text classification can be done through various machine learning techniques. The machine learning technique may be supervised learning technique such as Naive Bayesian [18], support vector machines [19], Artificial Neural Networks [21] or unsupervised learning technique [16, 19, 20] or hybrid approaches [22, 24]. The hybrid methods combine the supervised and unsupervised techniques to yield maximum accuracy in sentiment classification. However, the classification and summarization of sentiments expressed by the user in the free text or documents is more comprehensive process and it is quite different from simple text mining approaches. It is not an extractive summary or classification of entire documents by simply considering topic-indicative words or phrases. Instead, Sentiment classification involves tasks like generation of semantic feature-set, sentiment words or opinionated word identification corresponding to the features, determination of the semantic polarity orientation of the feature-opinion pairs, and find out overall sentiment by aggregating the mined results [1, 2, 23]. Second, the association of the extracted sentiment to a specific topic is difficult. Most statistical opinion extraction algorithms perform poorly in this respect as evidenced in [3]. They either i) assume the topic of the document is known *a priori*, or ii) simply associate the opinion to a topic term co-existing in the same context. The first approach requires a reliable *topic* or *genre classifier* that is a difficult problem in itself. A document (or even a portion of a document small as a sentence) may discuss multiple topics and contain sentiment about multiple topics.

III. Proposed Opinion Mining Framework

This section discusses the design of proposed fuzzy based opinion mining and sentiment analysis system, which automatically extracts features, opinions and linguistic hedges (modifiers) from unstructured user-generates reviews and based on their sentiment orientation, it classifies reviews as “positive reviews”, “negative reviews” and “neutral reviews”. During sentiment score calculation of opinionated word, it considers the affect of linguistic edges or modifiers on those opinionated words. For instance, “x is good” describes no hedge; “x is very good” describes an intensifying or concentrating hedge and in “x is not good” describes an inverting or modifying hedge.

The cleaned and parsed documents obtained from the pre-processing stage are given to the feature extraction stage. The features extraction stage extract the features by using various rules and the irrelevant features are filtered out by considering frequency of occurrence features. If the max frequency measure of any feature is below the threshold limit then it will be eliminated.

The K-Means clustering algorithm present in the Weka data mining software is used to classify the datasets using the generated matrix from the feature extraction stage which had already been converted into ARFF format as input. The System consists of three major steps as described in Figure 1: pre-processing step, feature selection step, and classification and summarization step .The system used a non-supervised sentiment classification approach for sentiment classification and it is evaluated using dataset of online customer reviews of mobile phones. The noises are usually in the form of spelling mistakes, grammar mistakes, mistakes in punctuation, incorrect capitalization, and usage non-dictionary words such as abbreviations or acronyms of common terms and so on. The main reason for this is these reviews are mostly written by non-experts and in short informal texts. After downloading the datasets from internet, the proposed system cleaned the documents by removing the html tags present in the document and it correct spelling errors. The texts are tokenized into tokens and the stop-words are detected and removed. Since words like preposition, digits, articles and proper nouns like name of cell phone etc. are considered as valueless in the sentiment analysis, hence these words are included in the stop word list. The sentences generated in this pre-processing can be parsed automatically by any linguistic parser. The proposed system used Stanford Linguistic parser for POS tagging of each word present in the sentences. POS tagger parses each sentences and tags each term with its part of speech. The definite noun phrases referring to topic features do not need any additional constructs such as attached prepositional phrases or relative clauses, in order for the reader to establish their referent. Thus, the phrase “the battery,” instead of “the battery of the digital camera,” is sufficient to infer its referent. There are expressions used to completely change sentence polarity, from positive to negative or vice-versa For example the sentence don't like this camera" is negative, also there is the term like" that alone is a positive marker. This kind of construct have to be well treated with great attention because not all the time negations are used to change opinion. Handle negations represents a limitation of Sentiment Analysis, derived directly from the complexity of written languages.

language. In the presented work there are considered and managed some simple case of sentiment shifting", to mitigate the negations problem. Some example of researched patterns (in form of POS Tag sequences) handled to identify a negation have been: BNVip3B, BNB, BNRiAs, BNVip3Ss, BNPCVis3**.**

IV. Feature Term Extraction

This method is based on the mixture language model by Zhai and Lafferty[23]: they assume that an observed documents *d* is generated by a mixture of the query model and the corpus language model. In our case, we may consider our language model as the mixture (or a linear combination) of the general web language model *θW* (similar to the corpus language model) and a topic-specific language model *θT*

$$\theta \quad = \quad \alpha\theta_W + \beta\theta_T$$

where α, β are given and sum to 1. α indicates the amount of background noise when generating a document from then topic-specific model. θ, θW and θT have multinomial distributions, θW = (θW1, θW2 , ...θWk), θT = (θT1, θT2 , ...θTk), and θ = (θ1, θ2, ...θk), where k is the number of words in the corpus. Intuitively, by calculating the topic-specific model, θ_T , noise words can be deleted, since the topic specific model will concentrate on words occurring frequently in topic-related documents, but less frequently in the whole corpus. The maximum likelihood estimator of $\theta_W$can be calculated directly Algorithm that computes the exact maximum likelihood estimation of the multinomial distribution of q in the following mixture model of multinomial distributions, p = (p1, p2, ...pk),
q = (q1, q2, ...qk), and r = (r1, r2, ...rk).

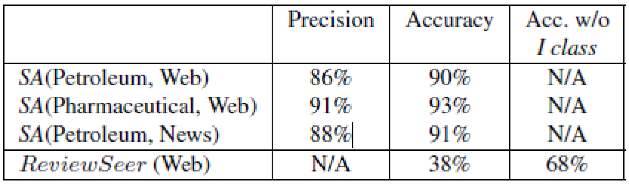
V. Sentiment Pattern based Analysis.

In they have developed an algorithm for predicting semantic orientation. Algorithm designed for isolated adjectives, rather than phrases containing adjectives or adverbs. They used four step supervised learning algorithm to infer the semantic orientationof adjectives from constraints on conjunctions. In that they got accuracy for classification of adjectives ranging from 78 % to 92 % depending on amount of training data. In they developed such system that generates sentiment timelines. It tracks online discussions on movies and generate plot which contains number of positive sentiment and negative sentiment messages over time. They used specific domain lexicons for movies. It is used instead of a hand-built lexicon. This work is used in automatic review rating, tracking advertising campaigns, tracking public opinion for politicians, tracking financial opinions by stock traders, tracking entertainment and technology trends by trend analyzers. In it is concerned with subjectivity tagging. They evaluated objectively present factual information. This paper identifies strong clues of subjectivity using the results of a method for clustering words according to distributional similarity. In 10-fold validation results, features based on both similarity clusters and the lexical semantic features are shown to have higher precision than features based on each alone *SA* first parses the sentence and identifies: In they have developed an algorithm for predicting semantic orientation. The documents are cleaned by removing the html tags present in the document and by correcting spelling errors. The texts are then tokenized into tokens and the stop-words are detected and removed. Stanford Linguistic parser is used for POS tagging of each term. By applying the six rules, the features, opinions and modifiers are extracted. By applying a threshold frequency limit of 3, the irrelevant terms are filtered out. The dataset used for this project is the Flip kart Reviews Database. The reviews in the dataset are consists of the attributes such as: Reviewer ID, Product ID, Review Text, Rating and time of the review. The main source of data used is the product reviews from Amazon. The reviews for a few popular phones have been obtained by building a web crawler. The web crawler has been written in Python using a scraping library called Beautiful Soup. Along with the review text, some additional data clusters and the lexical semantic features are shown to have higher precision than

VI Product Review Dataset.

We ran *SA* on the review article datasets .The review articles are a special class of web documents that typically have a high percentage of sentiment-bearing sentences. For each subject term, we manually assigned the sentiment. Then, we ran *SA* for each sentence with a subject term and compared the computed sentiment label with the manual label to compute the accuracy. The result is compared with the collocation algorithm and the best performing algorithm of *ReviewSeer*[3]. To our knowledge, *ReviewSeer* is by far the latest and the best opinion classifier. The collocation algorithm assigns the polarity of a sentiment term to a subject term[21], if the sentiment term and the subject term exist in the same sentence. If positive and negative sentiment terms coexist, the polarity with more counts is selected.

Table 1.0 Characteristics of the initial SA



	Precision	Accuracy	Acc. w/o *I class*
SA(Petroleum, Web)	86%	90%	N/A
SA(Pharmaceutical, Web)	91%	93%	N/A
SA(Petroleum, News)	88%	91%	N/A
ReviewSeer (Web)	N/A	38%	68%

5.2 Pretreatments

The characteristics of each of our initial sub-corpora are presented in the table. There are differences between them: the corpus for the age of "6-7 years" is the smallest one. To balance the corpora of the different age groups, we have sampled them according to the number of words: this feature is more reliable than the number of sentences, because the length of the sentences is a key factor which significantly varies from one age to another (see the following). To have comparable sub-corpora, the number of words is thus more reliable than the number of sentences.

BNP restricts the candidate feature terms to one of the following base noun phrase (BNP) patterns *NN*, *NN NN*, *JJ NN*, *NN NN NN*, *JJ NN NN*, *JJ JJ NN*, where *NN* and *JJ* are the part-ofspeech(POS) tags for nouns and adjectives respectively defined by Penn Treebank[10] .As we want to perform statistical measures on the morpho-syntactic labels, labeling errors must be reduced as much as possible. In [TDEW13], it has been shown that to learn a good tagger by supervised machine learning, it is more efficient to have a small annotated corpus similar to the target data than to have a large too different training set. So, we decided to use the labelled sentences which have been manually corrected for the evaluation of SEM as training data

In a dependency relation R, if there exist a such that POS(w1) = JJ* and POS(w2) = NN* and w1and w2 are not stop-words then w2 is considered as the feature and w1 as an opinion. Thereafter, the relationships advmod(w1,w3) and neg(w1,w4)are searched[04]. If both exist together then (w4 w3) will be the modifier or if advmod(w1,w3) only exist then the modifier will be w3 or if only neg(w1,w4)exist then modifier will be w4

In a dependency relation R, if there exist relationships nsubj(w1,w2), dobj(w1,w3) and nn(w3,w4) such that POS(w1) = VB* and POS(w2) = POS(w3) = POS(w4) = NN* and w1, w2, w3and w4 are not stop-words then (w4 w3) is considered as the feature and w1 as an opinion. Thereafter, the relationships advmod(w1,w5) and neg(w1,w6)are searched.

In a dependency relation R, if there exist relationships amod(w1,w2) and amod(w1,w3) such that POS(w1) = JJ* and POS(w2) = POS(w3) = NN* and w1,w2 and w3 are not stop-words then (w3 w1) is considered as the feature and w2 as an opinion. Thereafter, the relationships advmod(w2,w4) and neg(w2,w5)are searched. If both exist together then (w5 w4) will be the modifier or if advmod(w2,w4) only exist then the modifier will be w4 or if only neg(w2,w5)exist then modifier will be w5.

VII Feature-wise Total Score Calculation

The SentiWordNet score calculated in the previous step is utilized for feature wise SentiWordNet score calculation[15]. Feature wise SentiWordNet score is calculated by using the following algorithm

```
NB_Learn(Dt, C)
 /* collect all tokens that occur in Dt */
T ← all distinct words and other tokens in Dt
 /* calculate P(cj) and P(tk|cj) */
 for each target value cj in C do
 Djt ← subset of Dt for which target value is cj
 P(cj) ← |Djt ||Dt|
 Textj ← concatenation of all texts in Djt
 n ← total number of tokens in Textj
 for each word tk in C do
nk ← number of times word tk occurs in Textj
 P(tk|cj) ← nk+1 n+|T |
 done
 done
```

VIII Task, Data representation, performance measures

WSD can be described as a categorisation task where senses (FIN, RIV) are labels (C) the representation of instances (D) comes from the context surrounding the words to be disambiguated. For T = {along, cashier, stream, muddy, . . . }, we could have: d1 = halong = 1, cashier = 0, stream = 0,muddy = 1, . . . I and– f(d1) = RIV Performance can be measured as in text categorization.

A. Dirichlet distribution

A k-dimensional Dirichlet random variable θ can take values in the (k-1)-simplex, and has the following probability density on this simplex from the following Equations

$$(1)\ p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k}\alpha_i)}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\cdots\theta_k^{\alpha_k-1} \text{ --}$$

→**(1)**

$$p(\mathbf{w}) = \sum_z p(z)\prod_{n=1}^{N} p(w_n \mid z) \text{ ----}\mathbf{(2)}$$

$$p(\theta,\mathbf{z},\mathbf{w}|\alpha,\beta) = p(\theta|\alpha)\prod_{n=1}^{N} p(z_n|\theta)\,p(w_n|z_n,\beta)$$

→3

$$p(\mathbf{w}|\alpha,\beta)=\int p(\theta|\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta)\,p(w_n|z_n,\beta)\right)d^k\theta$$

$$p(d,w_n)=p(d)\sum_{z} p(w_n\,|\,z)\,p(z\,|\,d) \dashrightarrow (4)$$

The dataset used for this project is the Flip kart Reviews Database. The reviews in the dataset are consists of the attributes such as: Reviewer ID, Product ID, Review Text, Rating and time of the review.

The main source of data used is the product reviews from Amazon. The reviews for a few popular phones have been obtained by building a web crawler. The web crawler has been written in Python using a scraping library called Beautiful Soup.

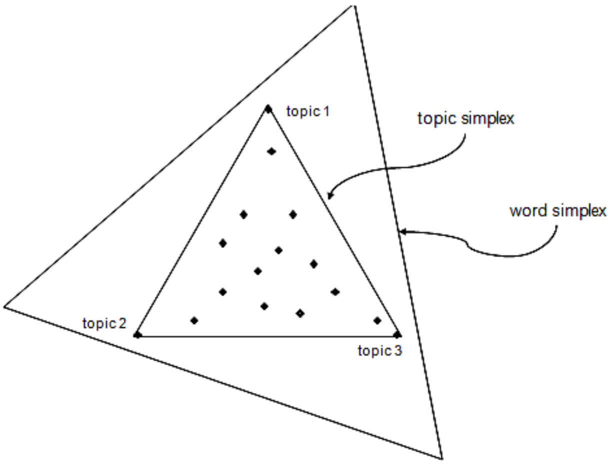


Fig 1.0 A geometric interpretation

B. Naive bayes classifier

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes[09]. It can solve diagnostic and predictive problems. This Classification is named after Thomas Bayes (1702-1761), who proposed the Bayes Theorem. Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data.

Input: messages m= {m1, m2, m3......mn},

Database: Naive Table NT

Output: Positive messages p= {p1, p2...},

Negative messages n= {n1, n2, n3....},

Neutral messages nu= {nu1, nu2, nu3...}

M= {m1,m2,m3...........}

Step: 1 Divide a message into words mi ={w1, w2, w3.........},i=1,2,....,n

Step 2: if wi NT Return +ve polarity and -ve polarity

Step 3: Calculate overall polarity of a word=log(+ve polarity)-log(-ve polarity)

Step 4: Repeat step 2 until end of words

Step 5: add the polarities of all words of a message i.e. total polarity of a message.

Step 6: Based on that polarity, message can be positive or negative or neutral.

Step 7: repeat step 1 until M NULL

LDMA, in addition to the two sets of random variables *z* and *w*, introduces a new set of variables *x* to detect an n-gram phrase from the text. LDMA assumes that the topics in a sentence form a Markov chain with a transition probability that depends on , a distribution zw, a random variable xi and the topic of previous word *zi-1*. Random variable x denotes whether a bigram can be formed with previous term or not. Therefore LDMA has the power to decide whether to generate a unigram, a bigram, a trigram or etc. Here we only consider generating unigrams and bigrams from LDMA. If the model sets *xi* equal to one, it means that *wi-1* and *wi* form a bigram and if it is equal to zero they do not.

We first start by preprocessing review document from the datasets. We extract the sentences according to the delimiters '.', ',', '!', '?', ';'. And then by removing Stop words and words with frequency less than three we extract a feature vector to represent review documents. By applying LDMA and the original LDA models examples of most probable aspects

	Total Tweets	Positive	Negative	Neutral
Canon	652	247	351	54
Nikon Coolpix	390	159	200	22
Nokia 6610	597	194	320	83
Apex	849	209	599	41
Creative Labs	1821	580	1103	138

Table 2.0 Classifier Evaluation

From the tables we can find that the LDMA model discovered more informative words for aspects or topics. In addition to the unigrams, LDMA can extract phrases, hence the unigram and bigram list of aspects are more pure in LDMA. Also LDMA associates words together to detect the multi-word aspects which are only highly probable in this model. Based on the results, LDMA can successfully find aspects that consist of words that are consecutive in a review document

IX. CONCLUSION

This paper shows that, the system performs very well in sentiment classification of user reviews with high accuracy. The implemented fuzzy functions to emulate the effect of various linguistic hedges such as dilators, concentrator and negation on opinionated phrases help the system to achieve more accuracy in sentiment classification and summarization of users' reviews in various aspects and various countries. As future work of this research, we can refine rule set to extract more dependency relations from datasets and that will help to improve the precision and recall values of the system by defining algorithms. From the analysis of review documents, it fails the system from defining correct dependency relations between word pairs and comparison results. If the system able to correct all the spelling and grammatical errors present in the review documents in the pre-processing step itself that will definitely improve the recall value of the System performance.

X. REFERENCES

[1]. Hemalatha, I., GP Saradhi Varma, and A. Govardhan. "Sentiment Analysis Tool using Machine Learning Algorithms."

[2]. Hemalatha, I., A. Govardhan, and G. P. Varma. "Machine Learning Methods in Classification of Text by Sentiment Analysis of Social Networks." *International Journal of Advanced Research in Computer Science* 2.5 (2011).

[3]. Hemalatha, I., GP Saradhi Varma, and A. Govardhan. "Preprocessing the Informal Text for efficient Sentiment Analysis." *International Journal* (2012).

[4]. G.P.Saradhi Varma,A.Govardhan, I.Hemalatha. "Sentiment Analysis Tool Using Machine Learning Algorithms." *Elixir International Journal, Elixir Comp. Sci. & Engg.* 58 (2013): 14791-14794.

[5].Y. Zou, C. Liu, Y. Jin, and B. Xie. Assessing Software Quality through Web Comment Search and Analysis. In Safe and Secure Software Reuse, pages 208–223. Springer, 2013.

[6]. N. Seyff, F. Graf, and N. Maiden. Using mobile re tools to give end-users their own voice. In Requirements Engineering Conference (RE), 2010 18th IEEE International, pages 37–46. IEEE, 2010.

[7]. C. Iacob and R. Harrison. Retrieving and analyzing mobile apps feature requests from online reviews. In MSR '13 Proceedings of the 10th Working Conference on Mining Software Repositories, pages 41–44.IEEE Press, May 2013.

[8]. L. V. Galvis Carreño and K. Winbladh. Analysis of user comments: an approach for software requirements evolution. In ICSE '13 Proceedings of the 2013 International Conference on Software Engineering, pages 582–591. IEEE Press, May 2013.

[9] D. Pagano and W. Maalej. User feedback in the appstore : an empirical study. In Proc. of the International Conference on Requirements Engineering - RE '13, pages 125–134, 2013.

[10] K. Schneider, S. Meyer, M. Peters, F. Schliephacke, J. Mörschbach, and L. Aguirre. Product-Focused Software Process Improvement, volume 6156 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, June 2010.

[11] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-oriented domain analysis (FODA) feasibility study. Technical report, DTIC Document, 1990.

[12] G. A. Miller. WordNet: a lexical database for English. Communications of the ACM, 38(11):39–41, 1995.

[13] A. Bagheri, M. Saraee and F. de Jong, 'Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews', Knowledge-Based Systems, vol. 52, pp. 20 1- 2 13,20 13.

[14] Chinsha, T. and S. Joseph. 'A syntactic approach for aspect based opinion mining', in the Proceeding of iEEE 9th International Conference on Semantic Computing (ICSC'15), pp. 24-3 1,20

[15] I. Pefialver-Martinez, F. Garcia-Sanchez, R. Valencia Garcia, M. Rodriguez-Garcia, V. Moreno, A. Fraga and J. Sanchez-Cervantes, 'Feature-based opinion mining through ontologies', Expert Systems with Applications, vol. 4 1, no. 13,pp. 5995-6008,20 14.

[16] Asghar, M.Z., et aI., 'A Review of Feature Extraction in Sentiment Analysis', Journal of Basic and Applied Scientific Research, vol. 4(3): pp. 18 1- 186,20 14.

[17] M. Eirinaki, S. Pisal and J. Singh, 'Feature-based opinion mining and ranking', Journal of Computer and System Sciences, vol. 78, no. 4, pp. 1 175- 1 184, 20 12.

[18] Isabella, J Analysis and evaluation of Feature selectors in opinion mining, Indian Journal of Computer Science and Engineering (IJCSE), Vol. 3 No.6 Dec 2012-Jan 2013

[19] Qi Su,Kun Xiang,Houfeng Wang,Bin Sun and Shiwen Yu(2006).Using Pointwise Mutual Information to Identify Implicit Features in Customer Reviews.ICCPOL ,LNAI 4285 ,pp.22-30,Springer(2006).

[20] Edison Marrese-Taylor , Juan D. Vel asquez , Felipe Bravo- M arquez Yutaka Matsuo (2013). Identifying CustomerPreferences about Tourism Products using an Aspect-Based Opinion Mining Approach, Procedia Computer Science 22 (2013) 182 191,Elsevier

[21] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques, In Proc. of EMNLP.

[22] Zhu J.,H.Wang,M,Zhu and B.K.Tsou.2011.Aspect based opinion polling from customer reviews. IEEE Transactions on Affective Computing,2(1):37-49.37

[23] Turney, p. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, In Proceedings of the 40th Annual Meeting of the Association for Linguistics, Philadelphia, Pennsylvania.

[24] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: a rating regression approach," in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010, pp. 783–792.

[25] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in Proceedings of the 18th international conference on World Wide Web. ACM, 2009, pp. 131–140.

[26]. B. Snyder and R. Barzilay, "Multiple aspect ranking using the good grief algorithm," in Proceedings of NAACL HLT, 2007, pp. 300–307.

[27]. S. Brody and N. Elhadad, "An unsupervised aspect-sentiment for online reviews," in Proceedings of ACL: HLT, 2010, pp. 804–812.