International Journal of Mathematics And its Applications
Volume 4, Issue 4 (2016), 303–312. (Special Issue)
ISSN: 2347-1557
Available Online: http://ijmaa.in/

International Journal of Mathematics And its Applications

Sentiment Analysis for Online Product Reviews using NLP Techniques and Statistical Methods

Research Article*

S.Muthukumaran[1], P.Suresh[1] and A.Victoria Anand Mary[2]

1 Research and Development Centre, Bharathiar University, Coimbatore, Taminadu, India.

2 Department of Mathematics, St. Joseph's College of Arts & Science, Manjakuppam, Cuddalore (Tamil Nadu), India.

Abstract: Opinion mining or sentiment analysis is the computational study of opinions or emotions towards aspects or things. The aspects are nothing but attributes or components of the individuals, events, topics, products and organizations. Opinion mining has been an active research area in Web mining and Natural Language Processing (NLP) in recent years. Online reviews are now popularly used for judging quality of product or service and influence decision making of users while selecting a product or service. Due to innumerous number of customer reviews on the web, it is difficult to summarize them which require a faster opinion mining system to classify the reviews. Reviews is proposed as there are many user generated product reviews available online many researchers have explored various supervised and unsupervised machine learning techniques for binary classification of reviews. 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).

1. 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 peoples 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.

* *Proceedings : National Conference on* `Recent Trends in Applied Mathematics` *held on 22 & 23.07.2016, organized by Department of Mathematics, St. Joseph's College of Arts & Science, Manjakuppam, Cuddalore (Tamil Nadu), India.*

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

2. 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.

3. 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.

4. 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 θ_W (similar to the corpus language model) and a topic-specific language model θ_T.

$$\theta = \alpha\theta_w + \beta\theta_T$$

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, θ_W and θ_T have multinomial distributions, $\theta_W = (\theta_{W_1}, \theta_{W_2}, \ldots, \theta_{W_k})$, $\theta_T = (\theta_{T_1}, \theta_{T_2}, \ldots, \theta_{T_k})$, and $\theta = (\theta_1, \theta_2, \ldots, \theta_k)$, 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 θ_W can be calculated directly as: Then the q_i's are given by:

$$q_i = \begin{cases} \frac{f_i}{\lambda} - \frac{\alpha}{\beta} p_i, & \text{if } 1 \le i \le t; \\ 0, & \text{otherwise.} \end{cases}$$

$$\lambda = \frac{\sum\limits_{i=1}^{t} f_i}{1 + \frac{\alpha}{\beta} \sum\limits_{i=1}^{t} p_i}$$

Algorithm that computes the exact maximum likelihood estimation of the multinomial distribution of q in the following mixture model of multinomial distributions, $p = (p_1, p_2, \ldots, p_k)$, $q = (q_1, q_2, \ldots, q_k)$, and $r = (r_1, r_2, \ldots, r_k)$.

$$r* = (r+1)\frac{n_{r+1}}{n_r}$$

For feature term selection, compute θ_{T_i} as

$$\theta_{T_i} = \begin{cases} \frac{f_i}{\lambda} - \frac{\alpha}{\beta}\theta_{W_i}, & \text{if } 1 \leq i \leq t; \\ 0, & \text{otherwise.} \end{cases}$$

$$\lambda = \frac{\sum_{i=1}^{t} f_i}{1 + \frac{\alpha}{\beta}\sum_{i=1}^{t}\theta_{W_i}}$$

Then sort candidate feature terms in decreasing order of θ_{T_i}. Feature terms are those whose θ_{T_i} score satisfy a predefined confidence level. Alternatively we can simply select only the top N terms.

5. Sentiment Pattern Based Analysis

For each sentiment phrase detected SA determines its *target* and final polarity based on the sentiment pattern database Section. *SA* first identifies the *T-expression*, and tries to find matching sentiment patterns [17]. Once a matching sentiment pattern is found, the *target* and sentiment assignment are determined as defined in the sentiment pattern. Some sentiment patterns define the *target* and its sentiment explicitly. Suppose the following sentence, sentiment pattern, and subject is given

SA first parses the sentence and identifies:

−matching sentiment pattern: ⟨`"take" OP SP`⟩

−subject phrase (SP): `this camera`

−object phrase (OP) : `excellent pictures`

−sentiment of the OP: `positive`

−T-expression : ⟨`camera, take, excellent picture`⟩

From this information, *SA* infers that the sentiment of *source* (OP) is positive, and associates positive sentiment to the *target* (SP): (camera, +). During the semantic relationship analysis, *SA* takes *negation* into account at the sentence level: if an adverb with negative meaning (such as not, never, hardly, seldom, or little) appears in a verb phrase, *SA* reverses the sentiment of the sentence assigned by the corresponding sentiment pattern.

For example, *SA* detects negative polarity from the following sentence: This camera is not good for novice users. We anticipated the shortcomings of the purely statistical approaches, and in this paper we show that the analysis of grammatical sentence structures and phrases based on NLP techniques mitigates some of the shortcomings. We designed and developed *Sentiment Analyzer* (*SA*)

5.1. 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 *Review Seer*[3]. To our knowledge, *Review Seer* 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.

	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%

Table 1. **Characteristics of the initial SA**

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-of speech(POS) tags for nouns and adjectives respectively defined by Penn Treebank [10].

Corpus	No. of Sentences	No. of Words	No. of Distinct Words	Average Length of the Sentences
1-2 years	41786	63810	3019	1.23
2-3 years	115114	324341	8414	2.15
3-4 years	60317	243244	8479	4.62
4-5 years	19747	74719	4465	4.71
5-6 years	4542	29422	938	6.96
6-7 years	3383	21477	841	6.88

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

Corpus	SEM	Re-trained tagger
1-2 Years	82%	85%
2-3 Years	70%	80%
3-4 Years	73%	88%
4-5 Years	75%	90%
5-6 Years	80%	92%
6-7 Years	87%	90%
Average	77.83%	87.5%

Table 2. **Impact of the re-learning on the accuracy of the distinct age groups**

In a dependency relation R, if there exist a such that $POS(w_1) = JJ^*$ and $POS(w_2) = NN^*$ and w_1 and w_2 are not stop-words then w_2 is considered as the feature and w_1 as an opinion. Thereafter, the relationships advmod(w_1, w_3) and neg$(w_1, w_4)$are searched [4]. If both exist together then $(w_4 w_3)$ will be the modifier or if advmod(w_1, w_3) only exist then the modifier will be w_3 or if only neg(w_1, w_4) exist then modifier will be w_4. In a dependency relation R, if there exist relationships nsubj(w_1, w_2), dobj(w_1, w_3) and nn(w_3, w_4) such that $POS(w_1) = VB^*$ and $POS(w_2) = POS(w_3) = POS(w_4) = NN^*$ and w_1, w_2, w_3 and w_4 are not stop-words then $(w_4 w_3)$ is considered as the feature and w_1 as an opinion. Thereafter, the relationships advmod(w_1, w_5) and neg(w_1, w_6) are searched.

In a dependency relation R, if there exist relationships amod(w_1, w_2) and amod(w_1, w_3) such that $POS(w_1) = JJ^*$ and

$POS(w_2) = POS(w_3) = NN^*$ and w_1, w_2 and w_3 are not stop-words then (w_3w_1) is considered as the feature and w_2 as an opinion. Thereafter, the relationships advmod(w_2, w_4) and neg(w_2, w_5) are searched. If both exist together then (w_5w_4) will be the modifier or if advmod(w_2, w_4) only exist then the modifier will be w_4 or if only neg(w_2, w_5) exist then modifier will be w_5.

6. 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|

Text j ← 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

6.1. 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: $d_1 = halong = 1$, $cashier = 0$, $stream = 0$, $muddy = 1, \ldots$ I and $f(d_1) = RIV$ Performance can be measured as in text categorization.

6.2. 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

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

$$p(w) = \sum_{z} p(z) \prod_{n=1}^{N} p(w_n|z) \tag{2}$$

$$p(\theta, z, w|\,\alpha, \beta) = p(\theta|\,\alpha) \prod_{n=1}^{N} p(\,z_n|\,\theta) p(\,w_n|\,z_n, \beta) \tag{3}$$

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

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

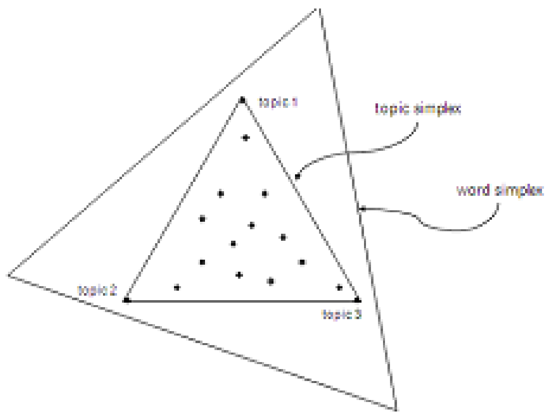

Figure 1. **A geometric interpretation**

6.3. 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 [9]. 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 $m = \{m_1, m_2, m_3, \ldots, m_n\}$,

Database: Naive Table NT

***Output*:** Positive messages $p = \{p_1, p_2, \ldots, \}$,

Negative messages $n = \{n_1, n_2, n_3, \ldots\}$,

Neutral messages $nu = \{nu_1, nu_2, nu_3, \ldots\}$

$M = \{m_1, m_2, m_3 \ldots\}$

Step 1: Divide a message into words $m_i = \{w_1, w_2, w_3 \ldots\}$, $i = 1, 2, \ldots, n$

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 z_{i-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 x_i equal to one, it means that w_{i-1} and w_i 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 3. **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 as-pects 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

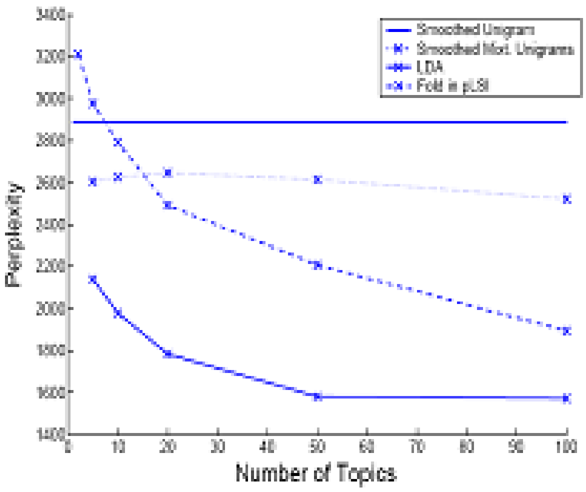

Figure 2. **Classifier efficiency**

7. Conclusion

In this paper, we proposed LDMA model, a model which extends the original topic modeling approach, LDA, by considering the underlying structure of a document and order of words in document. LDMA ignores the bag of words assumption of LDA to extract multi-word and n-gram aspects and topics from text data. We find that ignoring this basic assumption allows the model to learn more coherent and informatics topics. The key benefit of our proposed method is to study the sentiment analysis with minimal support as a composite problem that can be solved by succeeding partitions. In the supervised learning, the objective was to calculate the sentiment score of product features by aggregating opinion polarities of opinion words around the product features. We considered several techniques that could improve the classification performance. Therefore, choosing the features that are directly related to sentiment analysis is important, because it can performance and time and space efficiency.

References

[1] I.Hemalatha, G.P.Saradhi Varma and A.Govardhan, *Sentiment Analysis Tool using Machine Learning Algorithms*.

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

[4] I.Hemalatha, G.P.Saradhi Varma and A.Govardhan, *Sentiment Analysis Tool Using Machine Learning Algorithms*, 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*, Safe and Secure Software Reuse, Springer, (2013), 208-223.

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

[7] C.Iacob and R.Harrison, *Retrieving and analyzing mobile apps feature requests from online reviews*, In MSR '13 Proceedings of the 10^{th} Working Conference on Mining Software Repositories, (2013), 41-44.

[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, (2013), 582-591.

[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, (2013), 125-134.

[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, (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)(1995), 39-41.

[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, 52(2013), 201- 213.

[14] T.Chinsha and S.Joseph, *A syntactic approach for aspect based opinion mining*, Proceeding of iEEE 9th International Conference on Semantic Computing (ICSC'15), (2015), 24-31.

[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, 41(13)(2014), 5995-6008.

[16] M.Z.Asghar, *A Review of Feature Extraction in Sentiment Analysis*, Journal of Basic and Applied Scientific Research, 4(3)(2014), 181-186.

[17] M.Eirinaki, S.Pisal and J.Singh, *Feature-based opinion mining and ranking*, Journal of Computer and System Sciences, 78(4)(2012), 1175-1184.

[18] J.Isabella, *Analysis and evaluation of Feature selectors in opinion mining*, Indian Journal of Computer Science and Engineering, 3(6)(2012).

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

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

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

[22] J.Zhu, H.Wang, M.Zhu and B.K.Tsou, *Aspect based opinion polling from customer reviews*, IEEE Transactions on Affective Computing, 2(1)(2011), 37-49.

[23] P.Turney, *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, (2002).

[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), 783-792.

[25] Y.Lu, C.Zhai and N.Sundaresan, *Rated aspect summarization of short comments*, in Proceedings of the 18^{th} international conference on World Wide Web. ACM, (2009), 131-140.

[26] B.Snyder and R.Barzilay, *Multiple aspect ranking using the good grief algorithm*, in Proceedings of NAACL HLT, (2007), 300-307.

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