Maximal Phrase Based Opinion Extraction of Product and Movie Reviews

Supriya B. Moralwar[1], Sachin N. Deshmukh[2]

Abstract— **In Recent Years, the use of smart phones and mobile applications has been increased, as it generate large amount of data which can be in text or speech format, we have to deal with that data so there is need for sentiment classification to use that data in useful way. In this paper we are using phrases and words as a feature for sentiment classification of product reviews. Here the goal of this paper is to find a statistical way representative words and phrases that are used typically in positive and negative reviews. Here we are using two dataset first is Product Review and another is Movie Review and we are comparing the results of these datasets. The approach is based on machine learning approach; here machine learning techniques are used to classify the sentiment into positive or negative instead of predefined lexicon based approach. The motivation is that potentially each word or phrase could be considered as expressing something positive or negative in different case.**

***Index Terms*— Sentiment Analysis, Opinion Extraction, Machine Learning, Phrase Extraction, Distinctiveness of Phrase, Scoring Algorithm, Support Vector Machine (SVM).**

I. INTRODUCTION

As human being it really matters "What other people thinks", it is an important type of information for most of people during the decision-making process [1]. Before the awareness of www (World Wide Web), many people asked their relatives or friends about any product or object to recommend or help them while purchasing any product or about anything for example we requested colleagues for giving reference letters regarding the job. But in recent years use of internet is increasing and it is easily possible to find out about the opinions and experience of those people which are not known for us and conversely, opinions are available on web which are given by people. It generates the big data. It is very difficult to categories such data as positive or negative data. Before using this type of data, we need to process that data so that it could be used in a helpful way. Opinion mining or sentiment analysis tries to find out the different ways in which people can express their sentiments or opinion about particular product, object or event. It is used to classify the sentiment.

In this paper we perform document-level sentiment analysis which is performed on whole document. In this type of analysis a single review about a single topic is considered as document. Here product reviews and movie reviews are used as dataset. The experiments are performed on Multi-Domain Sentiment Dataset (Blitzer et al., 2007)[2] consist of 26 different products such as books, beauty, music, etc having 10 attributes such as unique_id, review_text, rating, etc which are downloaded from Amazon.com. The data has been split into positive and negative reviews. There are near about 100,000 reviews in this dataset. Second dataset used is movie reviews which are a collection of movie reviews used for various opinion analysis tasks, here reviews are split into positive and negative classes as well as reviews split into subjective and objective sentences by Pang & Lee [3].

From that dataset we are extracting the phrases and words which can be used as a feature for sentiment classification of product reviews. In this work, we are not using the general predefined polarity lexicons which contain predefined set of positive and negative words. Very often the same word or phrase could express something positive in one situation and something negative in other [2]. We need to identify the words and phrases, which are typically or maximally used in positive and negative documents of some domain. After having those words or phrase we use these to classify new sentiment documents from the same type of documents, from which we are extracting the phrases.

In this work, there is no need for traditional approaches of preprocessing such as stop word removal and stemming. Due to this approaches their might be chances of lose of information. In this work, we use Latent Semantic Analysis (LSA) as a technique for dimensionality reduction [4]. As LSA is a technique in Natural Language processing (NLP), for analyzing relationships between a set of documents and the term they contain by producing a set of concepts related to that documents. LSA is used to create the weighted clusters of words, which are derivationally related to each other, so the need of stemming is reduced by LSA. The goal is to design and develop Phrase Extraction Algorithm and Scoring Algorithm.

II. RELATED WORK

Mikio Yamamoto and Kenneth W. Chruch et al. (2001) [5], used the suffix array as a data structure for indexing. By using suffix arrays we can compute term frequency and document frequency for all substring in the corpus.

Gaston Burek and Dale Gerdemann (2009) [2], proposed technique for phrase extraction and perform experiment using K Nearest Neighbor classification technique to classify online discussion forums.

Funk et al. (2008) [6] this paper describe how to classify product and company reviews into one of the 1-star to 5-star categories. The features to the learning algorithm (also SVM) are simple linguistic features of single tokens. They report best results with the combinations root & orthography, and only root.

Another interesting related work is that of Turney et al. (2002) [7], He uses an unsupervised learning algorithm to classify a review as recommended or not recommended. The algorithm extracts phrases from a given review, and

ISSN: 2278 – 1323

determines their point wise mutual information with the words excellent and poor. He points out that the contextual information is very often necessary for the correct determination of the sentiment polarity of a certain word.
Kushal Dave et al.(2003) [8], He proposed a classifier which is based on information retrieval techniques for feature extraction & scoring.
Diana Inkpen et al [9], proposes two methods. First valence shifter which are of 3 types: negations, intensifiers & diminishes. Second he uses SVM for sentiment classification.
Jeonghee Yi et al. (2003) [10], Proposed Sentiment Analyzer that extracts sentiment about a subject from online text documents and uses NLP techniques to extracts the sentiments.
Qiang Ye [11], compared three supervised machine learning algorithms Naïve Bayes, Support Vector Machine and character based N-gram model for sentiment classification. SVM and N-gram model results are good than naïve bayes model.
Xavier Glorot [12], propose a deep learning approach which learns to extract a meaningful representation for each review in an unsupervised manner and Deep Learning is based on algorithms for discovering intermediate representations built in a hierarchical manner.

III. METHODOLOGY

A.Suffix Array

Suffix array data structure is used as database indexing technique. Suffix array will be used to compute number of statistics such as term frequency, document frequency and to find location of substring in a long corpus. Suffix array viewed as representation of suffix trees.
In this work we can construct suffix arrays and we have to find location of substring in a large corpus. For that we construct an algorithm to create a suffix array and sort it according to alphabet. A suffix array s is an array of all N suffixes which are sorted alphabetically. A suffix, s[i], is known as semi-infinite string, is a string that starts with position i and continue to the end of the corpus.
The algorithm, suffix_arr takes a corpus and length N as an input, and gives suffix array s as output.

```
Suffix_arr ← function (corpus, N) {
Let s be array of integers from 0 to N-1 suffix starting from s[i] on the corpus.
Sort s in alphabetical order of suffixes denoted by integer.
Return s.}
```

Algorithm 1: compute suffix array and sorting of suffix array.

By using this algorithm we can compute suffix array and sort them according to alphabet.

B. Longest Common Prefixes (LCPs)

The LCP array is a data structure introduced in order to improve the running time of their string search algorithm. The LCP array is auxiliary data structure to compute suffixes array. Here we compute the lcp values by comparing the suffixes and skip a prefixes based on a known lower bound for the lcp value obtained. LCP array contains N+ 1 integer. Each element of LCP indicates the length of the common prefix between s [i-1] and s[i]. The first element of LCP vector is zero. We pad zero to simplify the code. Due to Padding zeros it avoids the need to test for certain end condition.
By using suffix array SA and inverse suffix array SA^{-1} it is easy to compute LCP array. LCP vector is used to compute location of substring of corpus [13].
Input: text T[0...n], suffix array SA[0…n], inverse suffix array SA^{-1} Output: LCP array LCP[1…n]

```
1. l ← 0
2. For i ← 0 to n - 1 do
3. k ← SA^-1[i]
4. j ← SA[k - 1]
5. while T[i+l] = T[j +l] do l←l+1
6. LCP[k] ←l
7. If l > 0 then l← l – 1
8. Return LCP
```

Algorithm 2: LCP array construction

The above algorithm is faster than the straightforward implementation when the corpus contains long repeated substrings.

C.Distinctiveness

As we have created a suffix array and we have LCP vector of each phrase now we are interested in finding the distinctive phrases. The distinctive phrases are those phrases, which predominantly occur in one particular type of documents than another. These phrases are not evenly distributed across any particular category. The common measure to compute distinctive phrase is tf-idf measure. But it is problematic because idf refers to compute the terms that cluster in a small number of documents, rather than classification technique.
Another way to select distinct phrase is to divide our data into four categories as excellent, good, fair and poor. After that they cluster excellent and good into one class and fair, poor in another. We are not interested to cluster good and poor or excellent and fair. This approach leads to problem of burstiness.
In this work, I have compute or extract distinctive phrases those which are ranked to the top of suffix array while creating suffix array and computing LCP vector. Only the top most phrase of each review from each class is considered as distinctive.

D.Phrase Extraction Algorithm

The idea behind algorithm is that if a phrase is distinctive for a particular category, it does not matter how long the phrase is, as it helps for distinguishing one type of documents from another type of documents, that phrase should be extracted. To extract phrase using this type of technique, the whole collection of document should be represented as one long string. Then each phrase is divided into substring of a string. But this technique is very expensive to compute statistics such as term frequency and document frequency, to over this limitation yamamoto and church gives a solution to this problem by dividing substring into equivalence classes and then they perform operations on that.

ISSN: 2278 – 1323

E. Score of Phrases

Once we have extract distinctive phrases we need to score them so that they can be used for classification. In scoring algorithm, we can give score, how many times that phrase occur. Occurrence of phrase is calculated. As we classify reviews according to their occurrence i.e. the phrases which occur maximally in text are extracted as a feature for sentiment classification. For that purpose we need to score phrases which are extracted from both positive and negative reviews.

Suppose we have 10 positive phrases and 2 negative phrases which are extracted from reviews the occurrence vector of this is <10, 2>.

F. Construction of Vector

After getting the phrases which are distinct and occurs maximal in corpus we have to construct the vector of all phrases. We can construct vector by giving binary values zero or one. For that the phrases which are used for classification or those which occur maximally and are distinct we have to assign one value for that phrase and the phrases which are not of use for sentiment classification are represented by zero.

IV. EXPERIMENTAL WORK

For experiments, we are using SVM as a machine learning technique for classification. We can implement SVM Algorithm using LibSVM package. Support Vector Machines (SVMs) are a popular machine learning method for classication, regression, and other learning tasks. The Web address of the LibSVM package is at http://www.csie.ntu.edu.tw/~cjlin/libsvm [14]. LibSVM tool is currently one of the most widely used SVM software.

We are performing experiments on product reviews downloaded from amazon and movie reviews by Pang and Lee. First we extract phrases and then those phrases are taken as input for sentiment classification using LibSVM.

LibSVM takes training data as input which are in the form of vector, it have SVM_Train method to train SVM as it takes training data as input and automatically generates SVM model and finds number of support vector and other parameters such as kernel parameters and epsilon value, etc. After that we have SVM_Predict method which takes model, testing data as input and give accuracy as output. For cross-validation we apply the k-fold cross validation test, with k = 10.

We have performed experiments on multi-domain dataset which consist of reviews from different domain such as book and music. As we performed cross validation test with 10 k-fold. For books, music and movie review domain we perform five different experiments, each time using different subset of extracted phrases. It is very interesting to notice the results of experiments as they give different results. In book review domain, it gives better accuracy while performing 2nd experiment and gives accuracy of 96.77%. In music review dataset, the best accuracy achieved is 96.67 and in movie review dataset accuracy values are increased up to 98%. It shows that we have better accuracy while working with Movie review domain instead of Book and Music Review Domain. Following Tables illustrate the same thing.

Table I: Domain Book

Experiments	Accuracy
Exp 1	79.43%
Exp 2	96.77%
Exp 3	85.54%
Exp 4	96.36%
Exp 5	92.67%

Table II: Domain Music

Experiments	Accuracy
Exp 1	89.57%
Exp 2	89.68%
Exp 3	96.46%
Exp 4	96.36%
Exp 5	96.67%

Table III: Domain Movie

Experiments	Accuracy
Exp 1	98.00%
Exp 2	96.17%
Exp 3	95.38%
Exp 4	92.85%
Exp 5	95.17%

Above table shows accuracy of respective domain according the experiment performed.

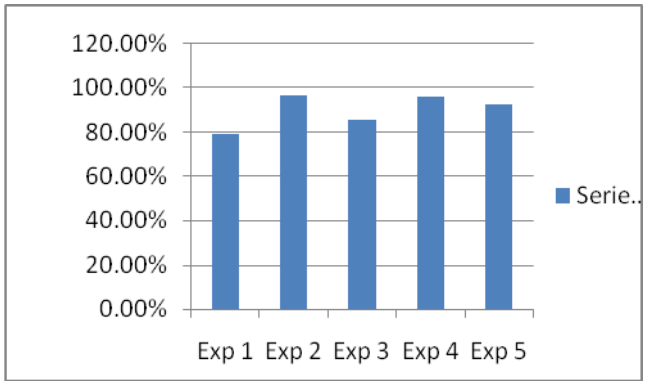

Fig 1: Plot of Accuracy of Book Review Domain

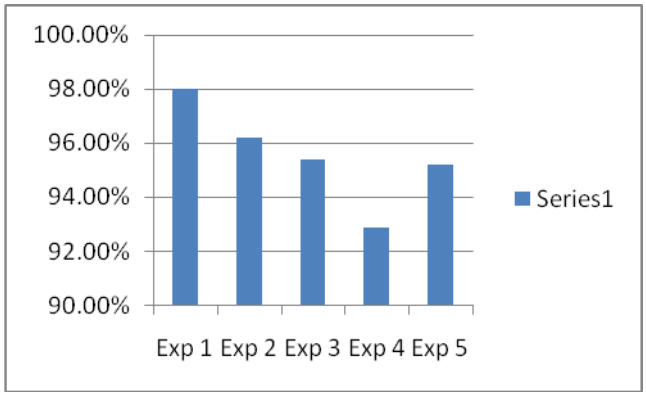

Fig 2: Plot of Accuracy of Music Review Domain

ISSN: 2278 – 1323

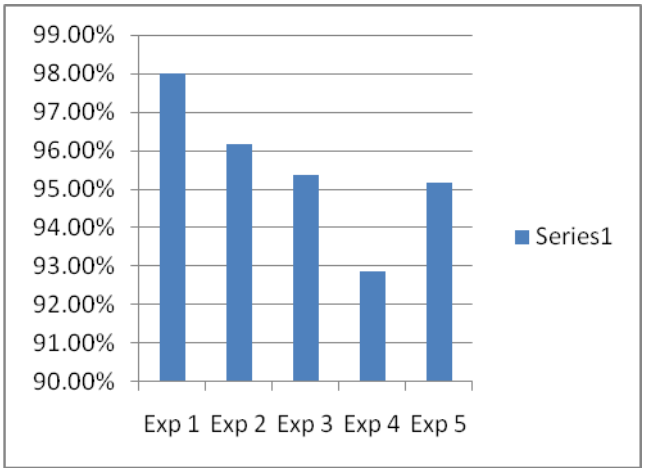

Fig 3: Plot of Accuracy of Movie Review Domain

Above Figures shows the Accuracy of corpus against the Experiment Performed on that.

V. CONCLUSION

In this work we presented different experiments on classifying product reviews of domains books and music by Blitzer et al., and movie reviews by Pang and Lee under the categories positive reviews and negative reviews using distinctive (maximally occurring) phrases as features for sentiment classification. For each domain best results were achieved with all extracted distinctive phrases as features. This approach gives better results while performing with movie reviews rather than with books and music reviews and gives accuracy about 98%. LibSVM tool is used to perform Support Vector Machine Algorithm.

ACKNOWLEDGMENT

The author will like to thank the university authorities and department of computer science and information technology Dr. B.A.M.U Aurangabad for providing the infrastructure to carry out the work. This work is supported by university commission.

REFERENCES

[1] Bo Pang and Lillian Lee. 2008, "Opinion mining and sentiment analysis", Foundations and Trends in Information Retrieval, Vol. 2, Nos. 1-2 (2008) 1-135.

[2] Maria Tchalakova, Dale Gerdemann, Detmar Meurers, "Automatic Sentiment Classification Of Product Reviews Using Maximal Phrases Based Analysis", Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT 2011, pages 111-117, 24 June, 2011, Portland, Oregon, USA 2011 Association for Computational Linguistics

[3] Bo Pang, Lillian Lee., and Shivakumar Vaithyanathan. 2002,"Thumbs up? Sentiment classification using machine learning techniques", In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79-86.

[4] Gaston Burek and Dale Gerdemann. 2009, "Maximal phrases based analysis for prototyping online discussion forums postings", In Proceedings of the workshop on Adaptation of Language Resources and Technologies to New Domains (AdaptLRTtoND), Borovets, Bulgaria.

[5] Mikio Yamamoto and Kenneth W. Church. 2001, "Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus", In Computational Linguistics, 27(1):1-30.

[6] Adam Funk, Yaoyong Li, Horacio Saggion, Kalina Bontcheva, and Christian Leibold. 2008," Opinion analysis for business intelligence applications", In Proceedings of First International Workshop on Ontology-supported Business Intelligence (OBI2008) at the 7th International Semantic Web Conference (ISWC), Karlsruhe, Germany.

[7] Peter D. Turney. 2002,"Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews", In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417-424.

[8] Kushal Dave, Steve Lawrence and David M. Pennock. 2003, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews", In *Proceedings of WWW*, pp. 519-528.

[9] Alistair Kennedy and Diana Inkpen, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters", University of Ottawa, Ottawa, ON, K1N 6N5, Canada.

[10] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, Wayne Niblack, "Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques", Dept. of Computer Science, University of Texas, Austin, TX 78712, USA.

[11] Qiang Ye a,b,*,Ziqiong Zhang a, Rob Law b, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches".

[12] Xavier Glorot, Antoine Bordes, Yoshua Bengio, "Domain Adaptation for Large-Scale Sentiment Classication: A Deep Learning Approach", Appearing in Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 2011,Copyright 2011 by the author(s)/owner(s).

[13] www.cs.helsinki.fi/u/tpkarkka/opetus/11s/spa/lecture10.pdf.

[14] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A Library for Support Vector Machines", Department of Computer Science National Taiwan University, Taipei, Taiwan.

AUTHORS PROFILE

Supriya B. Moralwar is pursing Masters in Computer Science and Engineering from Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad -431001, Maharashtra India.

Assistant Prof. Sachin N. Deshmukh
Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad -431001, Maharashtra, India.

ISSN: 2278 – 1323