ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 4, June 2013

An Approach for Online Analysis using Expectation Maximization

Arti Buche[1], Dr. M. B. Chandak[2], Akshay Zadgaonkar[3]

Student, Dept. of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, India[1]

Head of Department, Dept. of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, India[2]

Director, Zadgaonkar Software Pvt. LMT, Nagpur, India[3]

ABSTRACT: Opinion rich web resources such as discussion forums, review sites and blogs which are bulky and are available in digital form. For the purpose of customer and business perspective, the task of scanning these reviews manually is computational burden. Hence, to process reviews automatically and summarizing them in suitable form is more efficient. The distinguished problem of producing opinion summary addresses is how to determine the mood, sentiment or opinion expressed in the review with respect to a numerical feature value. In this paper, the focus is on the main task of opinion mining called as opinion summarization. The extraction of product feature, technical feature value and opinion are critical for opinion summarization as they affect the performance significantly. The proposed approach consists of a software system in which mining of product feature, technical feature value and opinion is performed. The main motto of this software system is to recognize the technical feature value depending on which the reviews are summarized. This software is helpful for humans to understand the technical values expressed in the reviews.

Keywords: Opinion Mining, opinion summarization, product reviews, product features, technical features.

I. Introduction

In recent years, we have witnessed that opinionated postings in social media have helped reshape businesses, sway public sentiments and emotions, and rapid growth of ecommerce as online shopping has increased which have profoundly impacted on our social and political systems. The customers can post their reviews regarding a product on merchant sites such as amazon.com, cnet.com etc. These customer reviews then become the source of information useful for both the customers and manufacturers. For development, consumer relationship management and marketing purpose, these customer reviews are highly valuable from product manufacturers' point of view.

The text processing is mainly done by Natural language Processing. The reviews are written in natural language scheme. The author's review from natural language textual information can be determined by several existing methods. Some machine learning model is employed with varying degree of effectiveness. Opinion mining is one of the types of natural language processing in which the public moods, attitude or sentiments are tracked.

Opinion mining is basically a technique of extraction and detection of subjective information from text documents. The main challenge in opinion mining is evaluation a product and sentiments expressed which is in the form of feature and can be labeled as positive or negative. [1]

The feature expressions which are to be grouped as domain synonyms help in generation of effective opinion summary. As text for an opinion mining application, consists of hundreds of feature expressions, it becomes tedious for human users to handle them. Scanning the large amount of documents require much time, therefore, automatic assistance is required for extracting the relevant information. Machine learning is employed for this purpose. [2]

In this paper, our goal is to develop an approach to establish relationship between the product candidate features (the topic of the sentiment), technical feature value (product numerical value) and the opinion word (sentiment). The summary is produced depending on these characteristics of the reviews. The paper is organized in following sections: section 2 describes the related work on mining product features and opinion extraction, section 3 proposed algorithm, section 4 evaluation measures to calculate accuracy of the system, section 5 Experimental results and section 6 conclusions and future work.

II. Related Work

The rich and unique source of data for the purpose of companies and people across disciplines are available mostly on blogs, micro blogs and reviews. The major contribution of this data is for the improvement of quality services and enhancement of deliverables. Generally for opinion mining the sources of opinions are mostly blogs and reviews sites where millions of product reviews are posted by the customers. [3, 4, 5] The techniques developed to perform opinion mining tasks are surveyed and analyzed as follows:

A. Sentiment Classification

Sentiment classification which is also called as polarity classification is the task of assigning labels to the opinionated documents as overall positive or negative opinion. The commonly used techniques for sentiment classification are machine learning algorithms and analyzing the text depending on the three levels: the document level, sentence level and feature (aspect) level.

In machine algorithms the knowledge and corresponding knowledge organization is used to analyze and interpret the acquired knowledge. There are three types of machine learning approaches such as supervised learning in which a function is generated to map inputs to desired outputs as labels as they are labeled by human experts. Eg: Naïve Bayesian Classifier. In unsupervised learning, clusters which are set of inputs are not known during training and are classified using syntactic patterns which express opinions. Eg: Part-of-speech tagging. While in semi supervised learning combination of both labeled and unlabelled inputs is used. [1, 8]

In opinion mining, the analysis of text is done on three levels. In document level classification, the whole document is used for extracting informative text. But the document level can be confusing and may complicate extraction as document categorization may contain conflicting sentiments. The sentence level classification is fine grained level in which polarity of sentences is calculated as positive negative or neutral. The problem with sentence level classification is co-reference problem. While in feature level, analysis of features is done which can be product attributes for determining sentiment of the document. The polarity is identified by extracting such features and it more fine grained model among all.

The classification at document level or sentence level is generally insufficient as they do not assign targets or identify opinion targets. Even if it is assumed that each document evaluates as a completely positive document or negative document. So for complete analysis, the identification of features is needed to decide whether opinion is positive or negative on each feature. [6, 7, 8]

B. Text Classification

The main challenge is to classify text which is available in massive volume on different websites, internet news etc. The text can be classified using classifiers which uses statistical learning algorithms to classify the text. The well known probabilistic classifier is Naïve Bayesian classifier in which the parameters to be estimated using labelled training data. These estimated parameters are used by the algorithm to further classify new documents. Due to high variance of such estimated parameters, the accuracy of the method suffers as it has small labelled training data set. [9]

Expectation Maximization (EM) algorithm was implemented which improves these parameter estimates. EM is the class of iterative algorithms which uses maximum likelihood or maximum posterior estimation for problems with incomplete data. It has two steps of computation: E (Expectation) step in which statistics over completions is computed and in M (Maximization) step the likelihood of data re-estimation is done. [9, 10]

C. Grouping Features

The challenge of grouping the features is tedious for human users so automatic assistance is required which can be in the form of regular expressions or clustering. In regular expressions the set of possible strings that can be matched and English sentences or email addresses. Regular expression language is small and restricted; hence the task of string processing becomes complicated using regular expressions. While in clustering hundreds of feature expressions are discovered which form cluster on basis of distributional similarity which rely on pre-existing knowledge. [11, 12]

D. Feature Extraction

For extracting the features from the customer reviews can also be information extraction task. The opinion mining facilitates extraction by some specific problem characteristics. The fundamental characteristic is that all opinions have target. Such target can be a feature or topic to be extracted from the review. Thus, the feature extraction is performed in order to extract the opinion expressions from the review. The extracted opinion expressions can either be positive or negative. The feature extraction can be done by using following approaches:

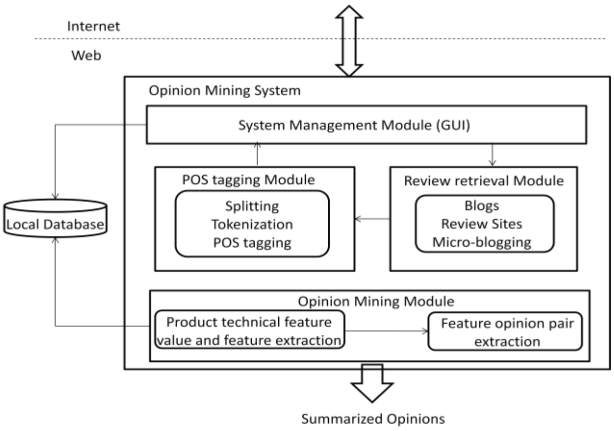
- Extraction based on frequent nouns and noun phrases
- Extraction by exploiting opinion and target relations
- Extraction using supervised learning
- Extraction using topic modeling [8]

E. Evaluation Measures

The evaluation measures commonly used in natural language processing are precision and recall. Precision is basically fraction of relevant retrieved instances while recall is fraction of retrieved relevant instances. Precision and recall are based on understanding and relevance measure. [12]

III. Proposed System Architecture

In this section, we describe our method to produce product feature summary extracted from customer reviews using semi supervised EM. The summary produced by the system is displayed in the form of tree view which consists of customer review sentiment and system processed sentiment. The crux methods are described in the following sections and the figure depicts architecture overview of the system.



In this section, we describe our method to produce product feature summary extracted from customer reviews using semi supervised EM. The summary produced by the system is displayed in the form of tree view which consists of customer review sentiment and system processed sentiment. The crux methods are described in the following sections and the figure depicts architecture overview of the system.

A. System Management Module

The system management module consists of system graphical user interface. The user interface mainly consists of two forms. The first form includes block of reviews and the reviews can be added dynamically. For adding the reviews, vocabulary generated by the system is to be considered which consists of a specific structure in which the review should be added such that every review should have some numerical value embedded in it. It also consists of a control which learns the complete block of reviews. After learning the reviews, a particular review is selected and analysed. Its summary is displayed graphically in tree view in second form.

The system proposed uses Open NLP library as machine learning based tool kit for natural language processing of a text and SharpNLP which is collection of tools of natural language processing. It provides common NLP tasks such as a splitter, a part-of-speech tagger, a tokenizer, a chunker, a parser, a name entity extraction and interface to WordNet lexical base which are driven by maximum entropy models processed by SharpEntropy library. Open NLP is based on maximum entropy based model and perceptron based machine learning. In addition, the SharpWordNet is provided by SharpNLP which is WordNet database library. In the proposed system, the Open NLP tasks performed are splitting, tokenizing and POS tagging. [1]

B. POS tagging Module

The text analysis systems mainly consist of a tagger as an important component. The significance of part-of-speech (POS) tagging for analysis and language processing is, they provide much amount of information about words and their tags. The tagger categorizes the given text into a set of lexical or part-of-speech tags such as noun, verb, adjective, adverb etc. The POS tags assigned to each word are the symbolic representation of such categorized word such as (NN) noun, (VB) verb, (JJ) adjective, (RB) adverb etc. Most commonly used tagset is the Penn tree bank tagset which consists of 45 tags. [14]

- Splitting

The sentence splitter is useful for obtaining an array of words from the given sentence. The basic and simple sentence splitter used is ('.') But it is the limited way of dividing the sentences of a paragraph of a text. Therefore, to handle most cases correctly, the extended splitters used are ('.') ('!') ('?') The input text is scanned and whenever it comes across any of these characters, it should decide whether or not it is the end of the sentence. To decide this, maximum entropy model comes into the picture.

The system uses an Open NLP tool namespace called as OpenNLP.Tools.SentenceDetect namespace with an object EnglishMaximumEntropySentenceDetector object, such that functionality packaged into classes so that it performs intelligent sentence splitting. In this functionality, the end-of-sentence positions which are possible are generated and related to the set of predicates. A set of predicates are generated by relating the possible end-of-sentence marker with characters before and after.

The maximum entropy model evaluates this set of predicates. The characters are separated off into a new sentence whenever the end result indicates optimum sentence break. The new sentences separated from the characters include the suitable characters and their position marked by the end-of-sentence marker. In the example below shows the OpenNLP task of splitting the given text.

Eg: The Nokia 808 PureView has 41-megapixel camera. The camera is amazing.

After applying the OpenNLP.Tools.SentenceDetect namespace to the given example along with the EnglishMaximumEntropySentenceDetector object and SentenceDetect method called. The constructor EnglishMaximumEntropySentenceDetector takes one argument and a string containing the file path to sentence detection maximum entropy model file. In the above example, the text shown is passed through SentenceDetect method; the resultant array will contain two elements: 'The Nokia 808 PureView has 41-megapixel camera.' and 'The camera is amazing.'

- Tokenizing Sentences

One of the steps in NLP tasks is to identify basic units called tokens which cannot be decomposed further in the processing. The tokens are nothing but the English words with the combination of which a sentence in a text is constructed. It is obvious, proper analysis or generation cannot be carried out without segregating these basic units. The simplest way to recognize the words in the given text is to use space marks as explicit delimiters. But these space marks may misled by overlooking distinguish complex units such as English idioms or fixed expressions.

The system uses a Tokenize method of EnglishMaximumEntropyTokenizer object. Initially each white space characters are splitted into candidate tokens. Then each candidate token is examined and if it is two characters long or contains only alphanumeric characters, and then it is considered as a token. Otherwise, each position of the character is examined to check if it should be splitted at that position into more than one token. This set of predicated is generated by considering various features such as numbers or letters on each side of characters, characters before and after split, and so on. The maximum entropy model is used to evaluate this set of predicates. There are two possible outcomes of the model, "T" for a split and "F" for non-split. If the outcome is "T" then the characters present on left of the split position are separated off as a new token. Thus, this tokenizer splits the words given in the sentence such that the words consist of contractions for e.g. "don't" is splitted into "do" which is recognized as a verb and the contraction "n't" is considered as "not" which is recognized as adverb such that the preceding verb "do" is modified. [14]

For example consider the given sentence

Samsung Galaxy Note2 has 8 megapixel camera which is very good and 1.5 GHZ processor which is not good.

After applying EnglishMaximumEntropyTokenizer object the result generated will be

Samsung | Galaxy | Note2 | has | 8 | megapixel | camera | which | is | very | good | and | 1.5 | GHZ | processor | which | is | not | good |.

- Part-Of-Speech Tagging

The words in the sentence are assigned part-of-speech, this task of assigning is called as part-of-speech tagging abbreviated as POS tagging. The array of tokens obtained from the tokenization process is fed to the POS tagger. The result generated is also an array of tags of same length as that of tokenizer array such that the index of tag array matches with the index of token array. The POS tags are coded abbreviations, which follow the scheme of PennTree bank, which is a linguistic corpus developed by University of Pennsylvania. The AllTags ( ) method provides all possible list of tags that follow PennTree bank description. The POS tagger was trained by the maximum entropy model which used the text from the oldest Wall Street Journal and Brown Corpus. The POS tagger is controlled by providing it with a POS lookup list. The constructor used by the system are EnglishMaximumEntropyPosTagger constructors, to specify the POS lookup list, there are two possible alternatives either by a POSLookupList or a by file path. The lookup list includes a text file with a word and its possible POS tags on each line, such that if a tagged word is found in the lookup list, the possible tags specified by the list are restricted by the POS tagger such that it selects the correct tag. It basically splits an input paragraph into sentences, each sentence is tokenized, and then Tag method POS tags the sentence. [15]

For example: Samsung Galaxy Note2 has 8 megapixel camera which is very good and 1.5 GHZ processor which is not good.

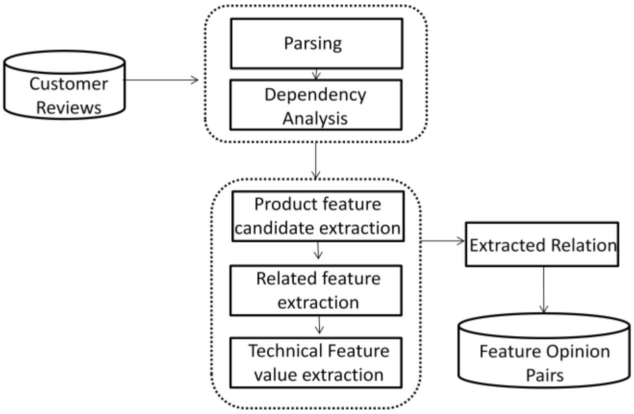
The part-of-speech tag is assigned by maximum entropy model in which each token is followed by a "/".

Samsung/NNP Galaxy/NNP Note2/NNP has/VBZ 8/CD megapixel/NN camera/NN which/WDT is/VBZ very/RB good/JJ and/CC 1.5/CD GHZ/NN processor/NN which/WDT is/VBZ not/RB good/JJ. /

C. Opinion Retrieval Module

The opinion retrieval mainly consists of extraction of the product candidate feature and related opinion feature extraction from the opinions expressed in the customer reviews. The product candidate feature consists of a brand name, a model name, a property, a part, a feature of a product and technical feature value of a product. The following sections describe the core methods used by the system.

The reviews are processed initially through a parser and then dependency analysis is performed on it.


Customer Reviews
Parsing
Dependency Analysis
Product feature candidate extraction
Related feature extraction
Technical Feature value extraction
Extracted Relation
Feature Opinion Pairs

- Parsing and Dependency Analysis

The NLP algorithms task is to perform parsing and produce a parser tree. A parser splits the customer review into a split tree which is a syntactic structure. The parse tree generated is constituency based parse tree which includes phrase grammar structure. [1]

The system generates the parse tree using a class named as EnglishTreebankParser class in which the most probable the ranked parse tree is generated. The object created is the root node in the tree such that the objects which generated are the best guess for the customer review. The tree generated can be traversed using GetChildren () method and has a property the Parent property. The parse node has tagset which has all tags of Penn Treebank found in Type property and which is equal to a MaximumEntropyParser. The output of parser is the textual representation of a parse tree graph.

For example: Samsung Galaxy Note2 has 8 megapixel camera which is very good and 1.5 GHZ processor which is not good.

(TOP (S (NP (NNP Samsung) (NNP Galaxy) (NNP Note2)) (VP (VBZ has) (NP (NP (CD 8) (JJ megapixel) (NN camera)) (SBAR (WHNP (WDT which)) (S (VP (VBZ is) (ADJP (RB very) (JJ good))))) (CC and) (NP (NP (CD 1.5) (NN GHZ) (NN processor)) (SBAR (WHNP (WDT which)) (S (VP (VBZ is) (RB not) (ADJP (JJ good)))))))) (. .)))

- Feature Extraction

The system extracts the features based on frequent nouns and noun phrases which occur in the review after using a POS tagger. The opinions expressed by many customers are first indentified for mining of product feature candidate, related opinion and technical feature value extraction.

The required extractions for the system are performed using semi-supervised learning which uses Expectation Maximization (EM) algorithm. This opinion retrieval module is the “E-step” of EM algorithm and consists of following extractions:

1) Product Feature Candidate Extraction

The product feature candidate extraction is done by considering the tagged sentence and after parsing the noun phrases in the review indicate the product feature candidate. A particular linguistic pattern is used for the noun phrases such as: NN, NN IN DT NN, and NN JJ NN where NN JJ DT are the Penn Treebank tags. [1]

For example: Samsung Galaxy Note2 has 8 megapixel camera which is very good and 1.5 GHZ processor which is not good.

Product candidate feature: camera, processor

2) Related Feature Extraction:

Related feature is the opinion expressed in the review. The system uses adjectives and adverbs as opinion words which are searched in the parse tree generated previously.

For example: Samsung Galaxy Note2 has 8 megapixel camera which is very good and 1.5 GHZ processor which is not good.

Related feature opinion: very good, not good

3) Technical feature value Extraction:

The reviews also consist of a numerical value which describes the product feature candidate. The numerical value is the technical feature value which is extracted by considering the cardinal number which is abbreviated as CD in tag set.

For example: Samsung Galaxy Note2 has 8 megapixel camera which is very good and 1.5 GHZ processor which is not good.

Technical feature value: 8, 1.5

The pseudo code for above extractions is as follows:

Pseudo code:

```
Input: IT: set of tag
        TS: tokenize sentence (noun list)

Output: Featurevalue, Technical feature value, ReviewSentiment

if (IT=Proper noun ("NNP") or IT=noun("NN"))  //  Check for the nouns in the tag set.
{
If (IT=CardinalNumber ("CD") or IT=determiner ("DT"))
// check the predecessor of nouns has a numerical value and determiner
                    {
                        Featurevalue=TS [i-1] element
                    }
}
if (IT=adjective ("JJ") or IT=verb("VBN"))// check if the tag is adjective or verb
  {
ReviewSentiment = IT;
    if (IT= Adverb ("RB"))// check the adjective or verb is preceded by an adverb
           {
                    Reviewsentiment=TS [i-1] element + TS [i] element
           }
    else if (IT= Cardinal Number ("CD")) // check if the adjective or verb is preceded by a cardinal number
           {
                    Featurevalue= TS[i-1] element
                    Technical feature value=CD
                    Reviewsentiment= sentiment string
           }
      }
```

D. *Opinion Mining Module*

The crux of the opinion mining module is to summarize the reviews. The reviews are generated by considering all the extractions performed in the opinion retrieval module. For effective summary generation, grouping of feature expressions is important such that they are domain synonyms. Clustering is the natural technique used to discover hundreds of feature expressions from text for an opinion mining application.

The system uses semi-supervised learning in which the features extracted from the opinion retrieval module are used as a set of labelled and unlabelled expressions. The partition of these features into labeled and unlabelled set is done on the basis of soft constraints of natural language called as sharing words and lexical similarity. For semi supervising learning, Expectation Maximization algorithm based on Naïve Bayesian is formulated. Expectation maximization is used to avoid the errors which might occur during labelling process and to re-assign classes to labeled set.

The proposed EM algorithm, the expectation step (E-step) computes expected statistics over completions rather than explicitly forming probability distribution over completions. The system's E-step consists of storing the extracted product candidate feature, related feature opinion and technical feature value. [1, 10]

Similarly, for the maximization step (M-step) consists of model re-estimation which can be thought of as 'maximization' of the expected log-likelihood of the data. In the system, M-step consists of following step:

- The stored technical feature values of particular product feature candidate are clustered in one group.
- Statistical calculations are carried out on those technical feature values as they need to be grouped into three different classes as best, average and poor so that the summary for the particular product feature candidate is generated.

- For grouping the technical feature values, the statistical method called standard deviation is used. Standard deviation is basically shows how much variation exists from the average (mean) or expected value so that the values get distributed into classes. The standard deviation formula is as follows:

For grouping the technical feature values, the statistical method called standard deviation is used. Standard deviation is basically shows how much variation exists from the average (mean) or expected value so that the values get distributed into classes. The standard deviation formula is as follows:

$$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})$$

Where {x1,x2…..xn}are the technical feature values extracted from the reviews and x¯is the mean value of these technical feature values, while the denominator N stands for the number of reviews and S is the standard deviation.

- In the proposed system, the standard deviation is calculated using largest technical feature value, smallest technical feature value and mean of technical feature value. The smallest standard deviation value calculated among these is considered and the product feature candidate is assigned to that particular class which can be best, average or poor which is considered as processed opinion by the system.
- The summary is generated from the above statistical calculations is in tree view form in which the sentiment processed by the system depending on the technical feature value extracted is rated into three classes as good, average and poor.

IV. Evaluation Measures

The system proposes a frequent and quick evaluation metric of calculating BLEU score. The idea behind proposing this metric is close the machine translation to a human translation it is better. According to a numeric metric, the quality judgement of machine translation is done by measuring its closeness with one or more references of human translation. Bleu score requires two ingredients: a numeric metric (translation closeness) and a corpus of human reference translations.

The keystone of this metric is the precision measure. The precision is given by the following formula.

$$P_n = \frac{no.\ of\ candidate\ translation\ words\ occur\ in\ reference\ translation}{total\ number\ of\ words\ in\ candidate\ translation}$$

Sometimes machine translation systems can generate high precision words which results in uncertainty. Therefore, modified precision is formulized.

Initially the precision is calculated using the formula stated above. A multiplicative factor called brevity penalty is introduce which matches high scoring candidate translation with reference translation in word, in length and in word order. The brevity penalty is calculated as:

$$BP = \begin{cases} 1, & if\ c < r \\ e^{1-r/c}, & if\ c\ \leq r \end{cases}$$

Where r is length of reference corpus and c is length if candidate translation. Thus the bleu score is calculated as follows:

$$Bleu = BP \times e^{(W_n \log P_n)}$$

Where Wn=1/N which is uniform weight.

As the system uses unigram its brevity penalty is 1 and Wn=1.

- Evaluation Results

The system evaluation is done by considering 20 reviews and their summary generated by the system with the summary available for testing.

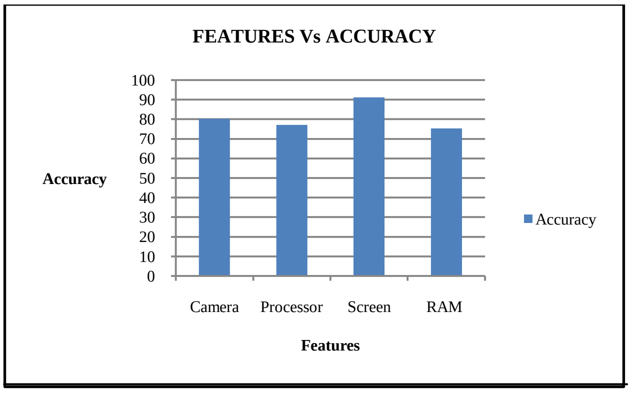
- Bleu Score Evaluation

Bleu score has range from 0 to 1. The score of few translations will be 1 unless they are identical to reference translation and score will be 0 if they do not match at all. The system evaluation is done by considering 20 reviews and accordingly the percentage accuracy of the system is calculated by taking average of the bleu score of individual review and the accuracy of the features encountered in the reviews is then calculated. [16]

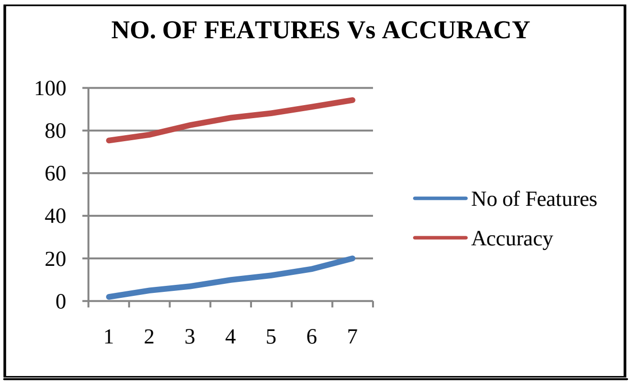
Accuracy of the features = Average of Bleu score * 100

Features	Average Bleu Score	Accuracy
Camera	0.8024	80.24%
Processor	0.7708	77.08%
Screen	0.9112	91.12%
RAM	0.7532	75.32%

The following graph shows the accuracy of the features extracted by the system

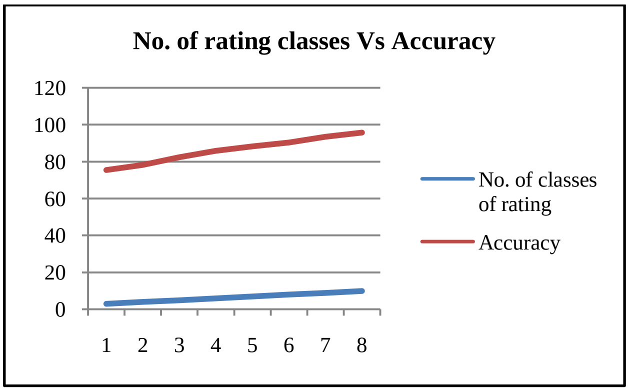


The system has considered four features. The machine is learned by using these four features.



The complete system accuracy increases with the determination of number of features. The system accuracy increases with the increase in number of feature extracted.

The system accuracy also increases with the increase in the number of rating classes of the summary generated by the system.



V. Conclusion And Future Work

This paper studied the problem of analyzing the online reviews of customer on products and generating the summary for those reviews by using modified Expectation Maximization algorithm based on Naïve Bayesian which summarizes review depending on features and technical feature value extracted from the reviews. OpenNLP library, machine learning based toolkit is used by the system for the processing of natural language text. Experimental results produced by the system shows the accuracy of the proposed algorithm.

References

1. Zhongwu Zhai, Bing Liu, Hua Xu and Peifa Jia, Clustering Product Features for Opinion Mining, WSDM'11, February 9–12, 2011, Hong Kong, China. Copyright 2011 ACM 978-1-4503-0493-1/11/02...$10.00

2. Gamgarn Somprasertsri and Pattarachai Lalitrojwong, Mining Feature-Opinion in Online Customer Reviews for Opinion Summarization, Journal of Universal Computer Science, vol. 16, no. 6 (2010), 938-955 submitted: 15/9/09, accepted: 4/3/10, appeared: 28/3/10 © J.UCS

3. G.Vinodhini and RM.Chandrasekaran, Sentiment Analysis and Opinion Mining: A Survey, Volume 2, Issue 6, June 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

4. Singh and Vivek Kumar, A clustering and opinion mining approach to socio-political analysis of the blogosphere, Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference.

5. Alexander Pak and Patrick Paroubek, Twitter as a Corpus for Sentiment Analysis and Opinion Mining

6. Ainur Yessenalina, Yisong Yue and Claire Cardie, Multi-level Structured Models for Document-level Sentiment Classification, Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 1046–1056, MIT, Massachusetts, USA, 9-11 October 2010. c 2010 Association for Computational Linguistics

7. V. S. Jagtap and Karishma Pawar, Analysis of different approaches to Sentence-Level Sentiment Classification, International Journal of Scientific Engineering and Technology (ISSN : 2277-1581) Volume 2 Issue 3, PP : 164-170 1 April 2013

8. Bing Liu, Sentiment Analysis and Opinion Mining, Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012.

9. K. Ming Leung, Naive Bayesian Classifier,Copyright c 2007 mleung@poly.edu.

10. Chuong B Do & Serafim Batzoglou, What is the expectation maximization algorithm?, 2008 Nature Publishing Group http://www.nature.com/naturebiotechnology

11. Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun and Tom Mitchell, Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning, 39, 103–134, 2000.2000 Kluwer Academic Publishers. Printed in The Netherlands

12. Sunghwan Sohn,1,* Manabu Torii,2,* Dingcheng Li,1 Kavishwar Wagholikar,1 Stephen Wu,1 and Hongfang Liu1, A Hybrid Approach to Sentiment Sentence Classification in Suicide Notes, Biomed Inform Insights. 2012; 5(Suppl. 1): 43–50. Published online 2012 January 30. doi: 10.4137/BII.S8961

13. YuanbinWu, Qi Zhang, Xuanjing Huang, LideWu, Phrase Dependency Parsing for Opinion Mining, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1533–1541, Singapore, 6-7 August 2009. c 2009 ACL and AFNLP

14. Daniel Jurafsky & James H. Martin, Speech and Language Processing: An introduction to speech recognition, computational linguistics and natural language processing, Copyright c! 2006, All rights reserved. Draft of July 30, 2007

15. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

16. G.Vinodhini and RM.Chandrasekaran, Sentiment Analysis and Opinion Mining: A Survey, Volume 2, Issue 6, June 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

17. Nidhi Mishra and C.K.Jha, PhD., Classification of Opinion Mining Techniques, International Journal of Computer Applications (0975 – 8887) Volume 56– No.13, October 2012

18. Nilesh M. Shelke, Shriniwas Deshpande, PhD. and Vilas Thakre, PhD., Survey of Techniques for Opinion Mining, International Journal of Computer Applications (0975 – 8887) Volume 57– No.13, November 2012

19. David Osimo and Francesco Mureddu, Research Challenge on Opinion Mining and Sentiment Analysis, Osimo, D. et al., 2010. The CROSSROAD Roadmap on ICT for Governance and Policy Modeling

20. Bo Pang and Lillian Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008) 1–135 2008 Bo Pang and Lillian Lee

21. David Vadas and James R. Curran, Parsing Noun Phrases in the Penn Treebank, 2011 Association for Computational Linguistics

22. David Vadas and James R. Curran, Adding Noun Phrase Structure to the Penn Treebank, Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 240–247 Prague, Czech Republic, June 2007. c 2007 Association for Computational Linguistics

23. Dae Hoon Park and Wei Peng, Generate Adjective Sentiment Dictionary for Social Media Sentiment Analysis Using Constrained Nonnegative Matrix Factorization, Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media 2011

24. Lin D. Automatic retreival and clustering of similar words, 1998: Proceedings of ACL 768-774

25. Fellbaum C, WordNet: An electronic lexical database. 1998: MIT press Cambridge, MA

26. Akshi Kumar and Teeja Mary Sebastian, Sentiment Analysis: A Perspective on its Past, Present and Future, 2012

27. Boris Kraychev and Ivan Koychev, Computationally Effective Algorithm for Information Extraction and Online Review Mining, 2010

28. Khairullah Khan and Baharum B. Baharudin, Identifying Product Features from Customer Reviews using Lexical Concordance, Research Journal of Applied Sciences Engineering and Technology, 2012

29. Zhai Z, Liu B, Xu H, and Jia P, Grouping Product Features Using Semi-supervised Learning with Soft-Constraints, in Proceedings of COLING. 2010

30. Gamgarn Somprasertsri and Pattarachai Lalitrojwong, Mining Feature-Opinion in Online Customer Reviews for Opinion Summarization, Journal of Universal Computer Science, vol. 16, no. 6 (2010), 938-955 submitted: 15/9/09, accepted: 4/3/10, appeared: 28/3/10 ©.

BIOGRAPHY

Arti Buche is pursuing Masters in Technology in Computer Science and Engineering Department from Shri. Ramdeobaba College of Engineering and Management, Nagpur.