A Semi-Automated Review Classification System Based on Supervised Machine Learning

Mrs. Mukta Y. Raut
Department of Computer Engineering
MIT Academy of Engineering, Alandi.
Pune, India.
rautmukta90@gmail.com

Dr. Sunita S. Barve
Department of Computer Engineering
MIT Academy of Engineering, Alandi.
Pune, India.
ssbarve@comp.maepune.ac.in

Abstract The field of opinion mining is expanding rapidly with the widespread use of internet for e-commerce and social interaction. One of the interesting use of opinion mining is in the field of online producer-consumer industry. The primary goal of the work presented in this paper is to perform a semi-automated sentiment classification on online product reviews for product evaluation using machine learning. We also aim to induce simplicity in sentiment classification; by using a method called Dual Sentiment Analysis, we relegate the need of using complex human annotations or very high end linguistic tools to solve the polarity shift problem in opinion classification. We also propose use of a pseudo-opposites dictionary based on our training corpus which is domain consistent with the training dataset.

Keywords Opinion mining, sentiment classification, polarity shift, machine learning.

I. INTRODUCTION

The upgrowth in usage of online social platforms for sentiment expression results in the generation of tons of data regarding various aspects of social media and online trading including product or service evaluations and feedbacks of consumers. This data has tremendous potential of mutual growth and development of social web and ecommerce. But as of today, very little of the available data is utilize in real time. Most of this data is collected in heterogeneous forms such as text reviews, feedbacks, star ratings, numerical ratings, different grading systems, and so on. Due to this data staging for transferring data into a single schema is of key importance. Moreover, the tremendous volume of this data and the speed at which it is generated makes it very difficult to perform manual analysis of the data for knowledge discovery.

This gave rise to the use of machine learning techniques, statistics and natural language processing to extract, identify, or otherwise characterize the sentiment content of a text unit. We term this process is collectively called as opinion mining or sentiment analysis. The automated machine learning techniques developed for sentiment analysis lets users and researchers concentrate on knowledge discovery and decision making rather than data staging and data analytics. The sentiment analysis can be carried out in two ways: supervised machine learning and unsupervised machine learning. While unsupervised learning techniques provide us complete automation, unsupervised techniques help us customize the machine learning as per our requirements. In our proposed system, we use supervised approach to train our classifier. This gives us freedom of choosing the reviews relevant to our domain and construct a domain specific corpus-based dictionary. In this paper, we present a simplified technique to classify reviews into positive, negative and neutral class. The method is also effective to solve the polarity shift problem encountered in automated opinion classification methods.

Polarity shift is a dialectal anomaly. Its occurrence can reverse the actual polarity of the text to its opposite (i.e. Positive sentence is classified as negative). There are three types of polarity shifts:

Negation: the polarity of text shifts from negative to positive or vice versa.

Explicit contrast: presence of two opposing opinions about a same entity in the neighboring sentences.

Sentiment inconsistency: can occur in long text or reviews where a part of review is expressing opposite polarity than that of the review as a whole; i.e. when reviewer dislikes entity as a whole but appreciates one sole aspect of it.

The most common and simple method to represent text for sentiment classification is Bag-of-Words model (BOW Model). In this model, a vector of independent words is used to store the review text. While doing so. The order of sentence is changed randomly; syntactic structures are disrupted. Some of the sentiment information can be lost as a result. Also, many sentiments opposite text are considered to be very similar in BOW model. Due to these reasons, standard machine learning algorithms often underperform in handling polarity shift when using BOW model for text representation.

In this paper, we propose a system which uses a data expansion technique called dual sentiment analysis [1]. In this we expand the original dataset by reversing the given reviews to their antonyms. By doing so we enrich the training process of our classifier. The classifier is trained twice while processing one review text, which adds to accuracy of sentence classification. The major contributions of the work presented here are as follows:

978-1-5090-4264-7/17/$31.00 ©2017 IEEE

1. We can eliminate the need of very large training dataset to gain stable classification output.
2. We achieve better review classification accuracy as compared to similar existing systems.
3. In case of feedbacks with mixed sentiment, the proposed method works efficiently by showing the probability of a given review of being positive, neutral or negative.
4. The work highlights significance of considering the neutral class in opinion classification.

The proposed system can be used as a stand-alone application for product evaluation. It can also be embedded into an existing online service portal as automatic rating module. The same system can be used for several domains. The modification required for using the system for different domains is loading domain specific training dataset and pseudo-opposites dictionary containing domain consistent words.

The paper is organized into the following sections: section II gives the overview of the related work. In section III we present the architecture of the proposed system. Section IV consists of algorithm used for reversing the reviews and feature extraction. Section V performs analysis of the designed system in regards with performance an accuracy, and lists observations. Section VI states conclusion and future work.

II. RELATED WORK

Different approaches are proposed by researchers to solve the polarity shift problem [9], [10], [11]. Many methods often need either extra human annotations or they have to use complex linguistic and semantic features. Due to this fact, such systems have limitations if intended to be used in small to medium scale business intelligence problems. In [12], a step by step sentiment classification system is shown. In this, the neutral class is considered before considering positive and negative class in the review text. [2] is an ongoing project analyzing political sentiments. It illustrates a method to develop corpora for detecting and analyzing sentiments in social media with help of an Italian project senti-TUT. The project consists of research about sentiments and irony in online political discussions. The corpus been built for this project refers to a structure of words or sentences representing certain properties and used for lexical, grammatical or other linguistic analysis. Corpus development in OMSA is a three-step method which includes collection,
annotations and analysis each of which is strongly dependent on the other [2]. Collection consists of choosing a dataset to compose the corpus and collecting the methodologies that can be applied to it. The annotation step includes defining scheme and applying it to the collected data. Annotation helps in
utilization of unstructured data for machine learning. Analysis of the developed corpus is then carried out. The annotated data then acts as training and test dataset in statistical machine learning tools for sentiment classification. we use feature selection and extraction methods to train the classifier. A big setback in this process is presence of noisy, irrelevant redundant attributes [3]. In [3], Abbasi et al. have given a rule based text selection methodology. It is a multivariate text feature selection method called Feature Relation Network (FRN) that considers semantic information and also leverages the syntactic relationships between n-gram features.

III. PROPOSED SYSTEM

In our system, we use a method called as Dual Sentiment Analysis [1]. Its sole purpose is dataset expansion. In this method, the given review is reversed in sentiment polarity. This gives us two training reviews each time we process a single review text:
(1) original review and
(2) polarity reversed review. To reverse the original review, we use a set of rules as follows:
(1) To reverse the text, we pick all the directional words followed by negation. Then all the directional words are reversed. Negation words are removed.
(2) To reverse the label, the original class label is reversed to its opposite by reversing directional words. The valence terms collected from both the original and polarity reversed review are used to predict the correct class of the reviews (positive, negative or neutral). The system is composed of Data Acquisition Module, Statistical Analysis Module and Performance Evaluation Module. Figure 1 gives the overall architecture of our system.

A. *Data Acquisition Module:* Data collection involves extracting this data from different sources into a single data structure. This data needs to be cleaned for redundancy, consistency, integration and other such properties as per requirement. This stage is also called as text preparation. To obtain trained dataset, we manually classify these reviews into positive, negative and neutral class and label them likewise. The corrective action is needed after the dataset is trained once, to assign increased weights to positive reviews and lower rates to negative reviews. The reversed review is obtained by aid of pseudo-opposites list stored in knowledge-base. This dictionary contains a list of words paired with their sentiment-opposite words. We also maintain a separate list of negation words. This output is then fed to second module.

Significance of Reversing the Reviews: By reversing the review, we get two reviews to be fed in the training dataset; the original review and the reversed review. Both the reviews are classified according to their polarity and then stored in the training dataset. So, while processing a single review, the classifier is trained twice, once for original review and second time for reversed review. Also two different valence words are stored in the knowledge bases while processing a single review.

B. *Statistical Analysis Module:* Classifier used to train the dataset is Naive Bayes classifier. It calculates the positivity and negativity of a sentence in form of posterior probability. It works on the Bayes theorem which states that the posterior probability of an event is directly

proportional to the product of the prior probability and the likelihood of the event. Here, Posterior probability is the probability of a review being positive or negative. Likelihood refers to the number of times the given valence words occur in the given class. Prior probability is the number of times the valence words occur in the training dataset.

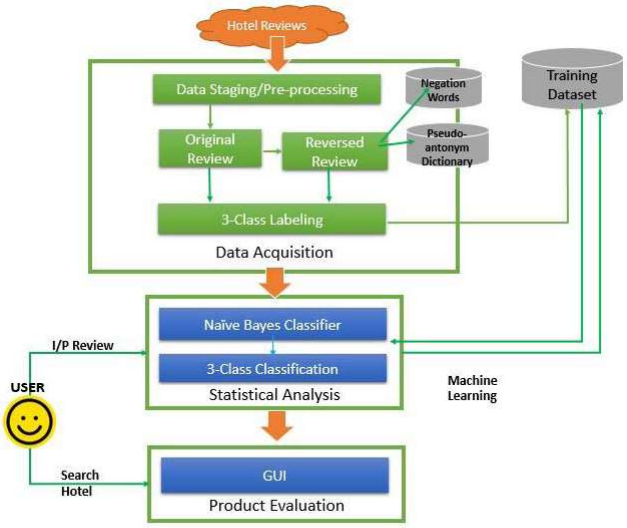

Figure 1: Architecture of Proposed System

We calculate the posterior probability of the review being positive, negative or neutral by using following formula:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \qquad (1)$$

Where,

$P(c|x)$ = posterior probability of given predictor

P(c) = prior probability of class

$P(x|c)$ = likelihood, i.e. the probability of predictor of given class

P(x) = prior probability of predictor

$P(x|c)$ is the conditional probability that word x belongs to class C. For text classification, it can simply be calculated by calculating the frequency of word x in class C relative to the total number of words in class c. To calculate the probabilities, we use following variation of the Naïve Bayes formula,

$$P(d_i|C) = \frac{count(d_i|C)}{\sum_i count(d_i|C)} \qquad (2)$$

In general, we find the ratio of occurrence of the given valence word in particular class to all the valence words in that class. For example, how many times fantastic occurs in positive class times probability of word fantastic occurring in training corpus divided by total number of reviews.

Significance of using Naïve Bayes Classifier: The Naïve Bayes classifier can be trained in very short amount of time. It can perform as accurate as we can train it. Also we do not require a huge amount of data to train the classifier. Naïve Bayes is a generative classifier. Thus, as the system using Naïve Bayes matures (i.e. learns), it performs better and better. By using Naïve Bayes, we can also control redundancy in our test samples to some extent. This is because it considers the presence or absence of a particular valence word, rather than calculating the occurrences of that word. We can implement the Naïve Bayes using the basic system requirements. That s why Naïve Bayes is considered as the baseline system for many sentiment analysis research problems.

C. *Product Evaluation Module:* The main idea of sentiment analysis is to convert text reviews into mathematical form. Classified reviews from module 2 are input into the product evaluation unit. Using this information, we draw a pie-chart for each product to be evaluated. Thus, after the completion of analysis, the text results are displayed on pie chart.

IV. ALGORITHM

In this section, we give the algorithm for reversing the reviews and extracting the valence words from the reviews. In the below algorithm, negativeKeywords and oppositeKeywords are both obtained from a list of negative keywords prepared manually and a pseudo-opposites dictionary created, respectively. The reviews are classified into positive, negative, or neutral in Dual Prediction step (DPs).

The steps of the Algorithm are as follows:

1. INPUT: string<original review>
2. INPUT: common_opposites dictionary
3. INPUT: stopwords, negative_vocabulary_words, positive_vocabulary_words
4. INITIALIZE: double EPSILON = 0.001; private variable CATEGORY_NEGATIVE = negative; private variable CATEGORY_POSITIVE = positive;
5. Proceed to reverse review by reversing the original review
6. INITIALIZE: string sentence, string reversedReview, ArrayList<string> rs
7. Print +reverse string
8. Call class DualTraining
9. Sentence = sentence.replace(".", " .");
10. get negativeKeywords
    String negativeKeywords[ ] = { "no", "did not", "didn't", "hadn't", "hasn't", "haven't", "isn't", "is not", "do not", "don t", "doesn't", "isn't", "can't", "are not", "cannot", "wouldn't", "would not", "shouldn't", "should not", "weren't", "were not", "wasn't", "was not", "not", "couldn't" };
11. for int i=0, i<arryList.size,
    Remove negativeKeywords
    //remove all stopwords, extract features
12. INITIALIZE: string sCurrentLine, featureList
13. Specify path of file containing stopwords
14. while (sCurrentLine = br.readLine) is not equal to null
15. if sCurrentLine = featureList

Then return feaureList
//reversing original review
16. Assign = as split operator
17. if arrayList.get(i).equalsIgnoreCase(column[0])
then replace column 0 with column 1
else if : if
arrayList.get(i).equalsIgnoreCase(column[1])
then replace column 1 with column 0
return arraylist
18. while line = br.readLine() is not equal to null
antonym_synonym.add(line);
return reversed sentence
// proceed to classify the original and reversed user review
19. End

V. SYSTEM ANALYSIS

A. Evaluation Metrics

The software resources used for evaluation purpose are Microsoft Excel spreadsheet, Microsoft Azure plug-in, AFINN dictionary with Eclipse Luna as JAVA development

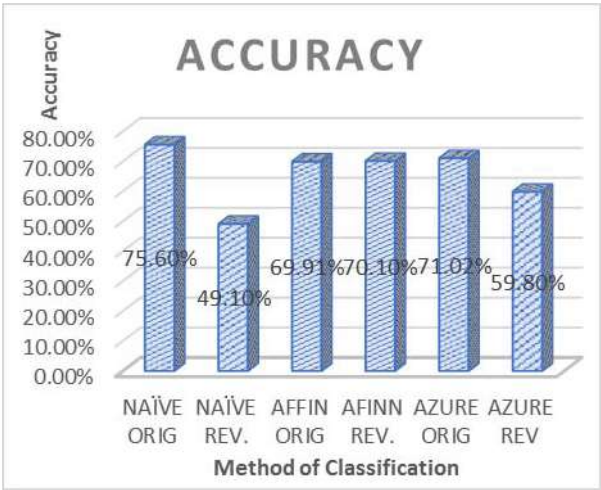

Figure 2: Accuracy of Naïve Bayes, AFINN and Azure

tool. Our proposed system performs sentiment classification using Naïve Bayes classifier. For feature extraction, we use two lexical lists, one named negative_vocabulary_words, and positive_vocabulary_words as second list. The words which do not have any sentiment value are stored into a list named stopwords. While performing statistical analysis on a review, the stopwords are removed from the review. To reverse the reviews, we use a pseudo-antonym dictionary named as common_opposites. This is a domain specific dictionary which is learnt from the training corpus.

Baseline: This is a pre-classified dataset. It contains reviews collected from two travel and hospitality sites Goibibo and MakeMyTrip. The reviews are labeled as positive, negative and neutral and are scored by the consumers. We compare our system with two of the following machine learning platforms:
*1. Microsoft Azur*e: Microsoft Azure Machine Learning (Azure ML) service is a unit of Cortana Intelligence Suite. It offers prediction and analysis of sentiment text and data interaction using natural language and speech through Cortana. For comparison purpose, we use its version that comes as a plug-in in Microsoft Excel spreadsheet.
2. AFINN Dictionary: AFINN is a list of English words rated for valence with an integer between minus five (negative) and plus five (positive). The words have been manually labeled by Finn Årup Nielsen in 2009-2011. The file is tab-separated.The test data consists of 14 datasets with number of reviews increasing in multiples of 15. Thus, first dataset contains 15 reviews, second contains 30 reviews, third contains 45 reviews, so on till 225 reviews. Analysis is performed on original reviews as well as reversed reviews.

B. Performance Evaluation

Accuracy of the System: To calculate accuracy of the systems, we consider six cases of review classification:

1. TP (True Positive): review is positive and is classified as positive.
2. TN (True Negative): review is negative and is classified as negative.
3. RN (True Neutral): review is neutral and is classified as neutral.
4. FP (False Positive): review is positive and is classified as negative/neutral.
5. FN (False Negative): review is negative and is classified as positive/neutral.
6. FRN (False Neutral): review is neutral and is classified as positive/negative.

The formula used for calculating accuracy of the systems is as follows:

$$\text{Accuracy} = \frac{(TN + TP + RN)}{(TN + TP + FP + FN + TNL + FRN)} \quad (3)$$

Table 1 represents accuracy of our system with original and reversed reviews using test datasets, and the accuracy of azure and AFINN dictionary using the same datasets (original and reversed). We can see that the accuracy of our system (Naïve Bayes classifier) is highest with original reviews, followed by Azure. But when reversed datasets are used, AFINN dictionary gives highest accuracy, followed by Azure and Naïve Bayes.

Table 1: Accuracy Achieved

Sr. No.	Methodology used	Accuracy Percentage
1ñ	Naive Bayes	75.60
2	AFINN	69.91
3	Azure	71.02

The results of table 1 are represented in figure 2.

Accuracy of Naïve Bayes: Table 2 shows accuracy of the Naïve Bayes classifier used for original reviews and reversed reviews. For Original reviews, the Accuracy Graph remains steady throughout the increasing number of samples in the dataset. Thus, for
implementing Naïve Bayes efficiently, we do not need a bigger training dataset to increase the classification accuracy.

In reversed reviews, however, Naïve Bayes performs poorly.

Table 2: Accuracy of Naive Bayes Original and reversed

Sr.No.	**No. of samples in dataset**	**Naive ORIG**	**Naive REV.**
1	15	80.00	40.00
2	30	73.33	56.67
3	45	77.78	57.78
4	60	76.67	56.67
5	75	73.33	56.00
6	90	73.33	51.11
7	105	72.38	50.48
8	120	73.33	48.33
9	135	75.56	48.15
10	150	76.00	46.00
11	165	76.22	45.45
12	180	76.67	45.00
13	195	75.90	44.10
14	210	76.67	44.76
15	225	76.79	45.98

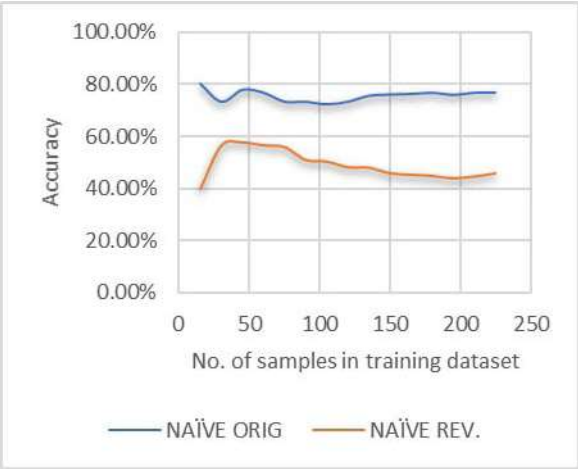

Figure 3: Accuracy Naïve Bayes.

Accuracy of AFINN Dictionary: The graph for original and reversed reviews declines slightly with increase in number of training samples in dataset. Thus the AFINN dictionary performs better with smaller datasets.

Table 3: Accuracy of AFINN Original and AFINN Reversed

Sr. No.	No. of Samples in Training Datasets	AFINN original accuracy in percentage	AFINN reversed
1	15	86.67	73.33
2	30	73.33	83.33
3	45	75.56	82.22
4	60	70.00	75.00
5	75	70.67	70.67
6	90	70.00	70.00
7	105	66.67	67.62
8	120	67.50	66.67
9	135	66.67	65.19
10	150	68.00	65.33
11	165	67.88	65.45
12	180	66.49	66.67
13	195	66.15	67.18
14	210	66.51	67.14
15	225	66.52	65.63

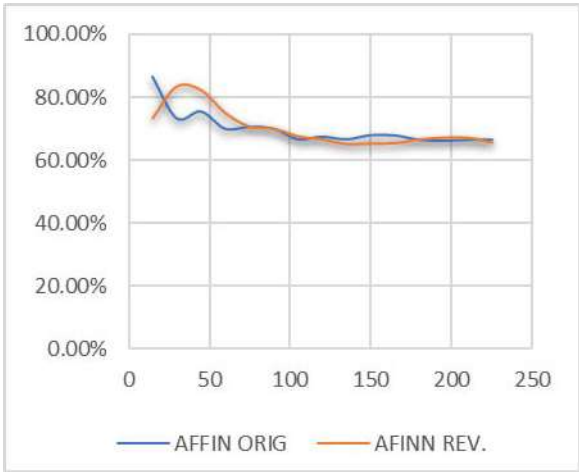

Figure 4: Accuracy of AFINN

Accuracy of Azure: The accuracy of original reviews declines slightly with increase in number of training samples. Accuracy is highest with smaller number of training samples.

The main reason of reduced accuracy of Azure Machine Learning is due to exclusion of Neutral class in classification. The accuracy is high when reviews are classified only as positive and negative. However, there are sentences which lack sentiments (Vasilis Vryniotis, blog.datumbox.com), although they contain one or more valence words. If we force such sentences into either positive or negative class, it leads to overfitting of training data (Vasilis Vryniotis, blog.datumbox.com). Overall classification accuracy is affected by this and we get more sentences which are false negative or false positive, reducing accuracy when cross checked against manual classification, which we consider as baseline.

Table 4: Accuracy of Azure Machine Learning

Sr.No	No. of samples in dataset	AZURE ORIG	AZURE REV
1	15	93.33%	60.00%
2	30	70.00%	66.67%
3	45	75.56%	68.89%
4	60	76.67%	71.67%
5	75	73.33%	61.33%
6	90	71.11%	60.00%
7	105	66.67%	59.05%
8	120	65.00%	57.50%
9	135	67.41%	57.46%
10	150	66.67%	57.33%
11	165	66.67%	56.36%
12	180	68.89%	55.56%
13	195	66.67%	56.41%
14	210	68.57%	54.76%
15	225	68.75%	54.02%

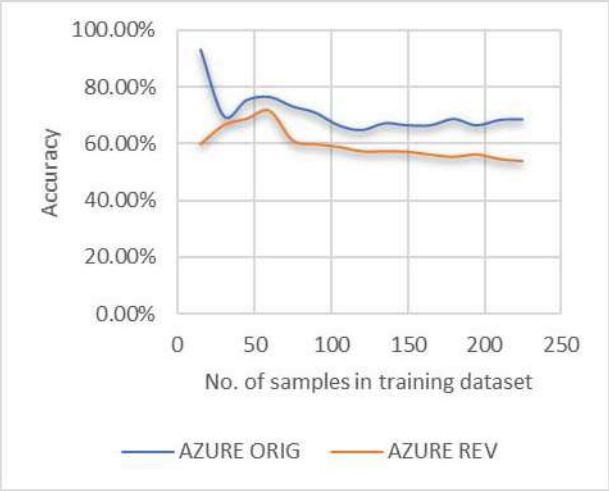

Figure 5: Accuracy of AZURE Original and AZURE Reversed

Comparison between Naïve Bayes-AFINN-Azure:

In Figure 6, we show the comparison between our system, AFINN, and Azure in review classification. As we can see, the graph indicating classification accuracy of Naïve Bayes is stabilized after a certain training interval. This suggests that the Naïve Bayes classifier converges quickly and so requires a small amount of training data as compared to other two. The accuracy remains low until the classifier is completely trained for all the valence words in the training corpus. However, Naïve Bayes do not consider the relations between them.

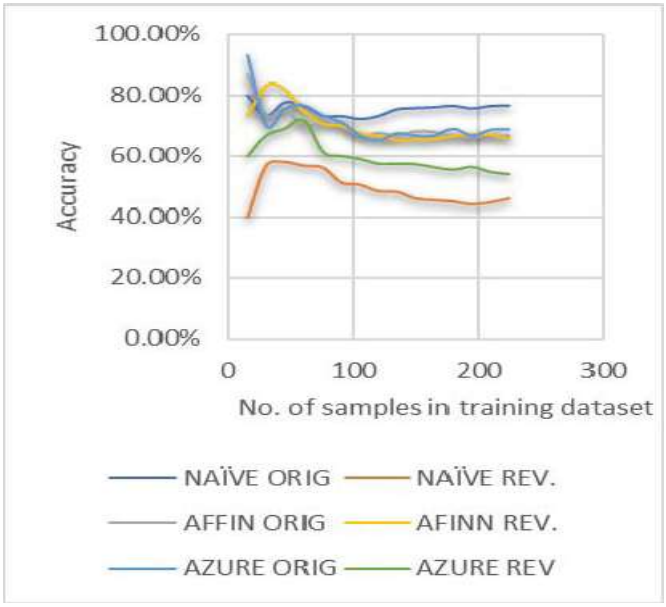

Figure 6: Accuracy Comparison between Naïve Bayes AFINN-Azure.

VI. OBSERVATIONS

1. The proposed algorithm does not depend on size of dataset for classification accuracy after a specific training interval.
2. The accuracy of system remains stable and does not show significant variation with change in number of reviews in training and test dataset.
3. The algorithm works well with the reviews with dual sentiment and helps in deciding the score of positivity or negativity of a review.
4. More efficient method needs to be developed for staging of training data, which will further improve accuracy.
5. The consideration of Neutral Class in classification prevents overfitting or underfitting of training and test data.
6. Although reverse reviews help in training the dataset faster, we cannot use them for increasing accuracy. This is because, when the review is reversed by the algorithm, the syntactic structure is broken and the review becomes partially incoherent.
7. Sentiment Analysis for product evaluation is difficult even when restricted to straight support vs opposition judgments in formal environments.

VII. CONCLUSION

The Naïve Bayes classifier gives most accurate results in sentiment classification of product reviews. The use of corpus-based domain specific dictionary for statistical analysis (i.e. common_opposites dictionary) gives better accuracy as compared to AFINN dictionary which gives positive/negative scores to the valence words and calculate the scoring of the review. Also by using Naïve Bayes classifier, we don t need a

very big training dataset, rather a small dataset with selective reviews.

REFERENCES

1. Abbasi A., France S., et al., (2011), "Selecting Attributes for Sentiment Classification Using Feature Relation Networks", IEEE Transactions on Knowledge and Data Engineering, 23(3): 447-462.

2. Bosco C., Patti V., et al., (2013), "Developing Corpora for Sentiment Analysis: The Case of Irony and Senti-TUT", Published by The IEEE Computer Society, 55-64.

3. Cambria E., (2015), "Affective Computing and Sentiment Analysis", Published by The IEEE Computer Society, 102-107.

4. Dadwar M., Hu C., (2014), "Scope of Negation Detection in Sentiment Analysis", Human Media Interaction Group University of Twente Enschede, Netherlands.

5. Fang X., Zhan J., (2015), "Sentiment Analysis using Product Review Data", Fang and Zhan Journal of Big Data 2:5.

6. Pang B., Lee L., (2008), "Opinion Mining and Sentiment Analysis", Found Trends Inf. Retrieval 2(1-2):1135.

7. Tang D., Wei F., et al., (2016), "Sentiment Embeddings with Applications to Sentiment Analysis", IEEE Transactions on Knowledge and Data Engineering,28(2): 496-509.

8. Xia R., Xu C., et al., (2015), "Dual Sentiment Analysis: Considering Two Sides of One Review", IEEE Transactions on Knowledge and Data Engineering, 27(8).

9. Liu B. (2012), Sentiment Analysis and Opinion Mining, Series Synthesis Lectures on Human Language Technologies (16).

10. Mills M., Bourbakis N. (2014), "Graph-Based Methods for Natural Language Processing and Understanding: A Survey and Analysis," in IEEE Transactions on Systems, Man, and Cybernetics: Systems 44(1):59-71.

11. Pang B. And Lee L. (2008), Opinion Mining and Sentiment Analysis , Found. Trends Inf. Retrieval, 2,(1-2):1135.

12. Patten T., Call C., Mitchell D., Taylor J. Lasser S. (2016), "Defining the Malice Space with Natural Language Processing Techniques," 2016 Cyber security Symposium (CYBERSEC), Coeur d'Alene, ID, USA 44-50.

13. Pukish M., Rycki P., Wilamowski B.(2015), "PolyNet: A Polynomial Based Learning Machine for Universal Approximation, in IEEE Transactions on Industrial Informatics 11(3):708-716.

14. blog.datumbox.com