Polarity Categorization with Fine Tuned Pipeline Process of Online Reviews

Prabha Natarajan[1], Vignesh Sankaran[2], B.Santhi[3], G.R.Brindha[4]
[1, 2, 3, 4] Department of Information and Communication Technology, School of Computing, SASTRA University
Thanjavur, India.
[1]prabhanatraj@gmail.com
[2]vigneshshankaran@gmail.com

***Abstract*: The development of Web 2.0 concept increased the web storage by offering information sharing from anywhere in the world. But how to use this content effectively and efficiently is the challenging task which is the important research in the field of Sentiment Analysis and Opinion Mining. This paper focus on these online data to process the web content using a pipeline processing which is applied to online reviews about products and generating a polarity checking tool for the user to provide them decision support information. Most of the research focuses on classification of polarities instead of pre-processing of data. But our idea is fine tuned pipeline processing will help us give better categorization. Classification has been achieved with many techniques, mainly depends on Machine Learning. This study also focuses on ranking using different classification techniques.**

Keywords: Opinion Mining, Machine Learning, Classification, SVM, Naïve Bayesian

I. INTRODUCTION

Various applications for opinion mining have made it the need for research at present. Today, Internet contains a large amount of data. Opinion mining deals with opinions of Internet users in terms of categorizing, extracting, and summarizing about a particular product, person or service. Currently researchers are focusing more on classification of polarity feature present in reviews. With the abundant growth of web 2.0 concept, such as blogs, forms and social networks people share their views and experiences through reviews and online discussions for decision making. In the current scenario, people need automated software for almost everything. Today, to buy a product irrespective of its value either by cost or anything, it becomes a normal one to check the online reviews and discussions in forums available all over internet. But still, looking for those sites and processing information from them remains a great task because of voluminous data availability. Each and every site might contain a large quantity of data that a common user will face difficulty finding relevant sites and extracting opinions in them. Hence automated opinion mining system becomes a vital one.

The focus of our project is to categorize the polarity of the online reviews with fine-tuned process. Classification has been achieved with many techniques, most of which depends on Machine Learning Techniques. Also, this paper gives a study on ranking and evaluating results based on the comparison of ranking classification using machine learning approach i.e. In Short, we deal with the following two modules.

1. Polarity Categorization - whether the user comments out positive or negative opinion.
2. Ranking - using Naïve Bayesian and SVM & comparing which works fine on our preferred dataset.

II. RELATED WORK

As the web information keeps increasing, the need for opinion mining arises. Many of the researchers contribute on opinion mining research. Researcher Li Chen et al.[1] deals with feature-level opinion mining he describes how the Conditional random field model was adopted for feature-level opinion mining. Another researcher Stephen Shaoyi Liao et al.[2] deals with extracting comparative relations from customer opinion data. He explains that the two-level CRF model can extract the comparative relations of customer opinion data with good accuracy. Magdalini Eirinaki et al. [3] presents an algorithm not only deals with overall sentiment score but also with semantic orientation of review that lead to a certain opinion. RushdiSaleh et al.[4] speaks about the SVM classification of product mining in various domains. They have dealt with different weighing algorithms like TF-IDF and several n-gram techniques and have found that SVM is a great tool to deal with opinion mining tasks. Daniele. O'Leary [5] gives a brief on blog mining. He has worked on mining opinion and sentiment from blogs. He has also made a relationship between information in online discussions and blogs.Earlier, Qingliang Miao, Qiudan Li, Ruwei Dai [6] proposed a sentiment mining and retrieval system which extracts data from product reviews by making use of data mining and information retrieval mechanism. They have proposed a mining mechanism named temporal opinion quality (TOQ) and evaluated results based on that. Also, Huifeng Tang[7] has taken survey for almost all like subjectivity detection, Sentiment Classification, Supervised and unsupervised classification. His paper deals with almost all about opinion mining and machine learning mechanism. Still Earlier, Bo Pang, Lillian Lee, Shivkumar Vaidyanathan [8] have given a detailed explanation about Classification of a particular dataset using Naïve Bayesian and Support Vector Machines.

A. *Module 1*

First Process deals with categorizing the online reviews whether it is positive or negative. The Process flow is given below.

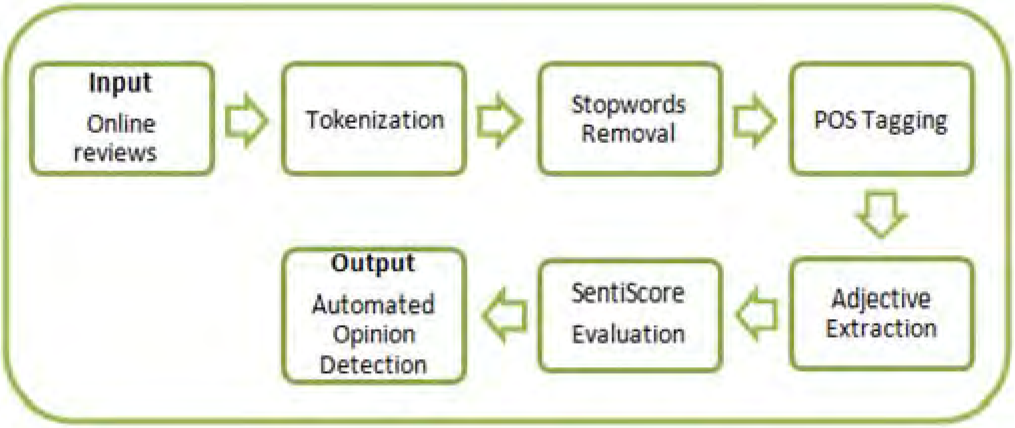

Fig.1. Process Flow Diagram for Opinion Detection

1) Methods Used:

1. Tokenization: The given input (online reviews) is split into individual units called tokens. Ex. Viswaroop is a onetime watch for sure. This becomes ‘Viswaroop’, ’is’, ’a’, ’onetime’, ’watch’, ’for’, ’sure’.
2. Stop words Removal: Rarely used words such as Prepositions, Questioning words and punctuations are removed. The above example becomes ‘Viswaroop’, ‘onetime’, ‘watch’, ‘sure’ after removing ‘is’, ‘a’ and ‘for’.
3. POS Tagging: Remaining words in the corpus are tagged using POS Tagger. This becomes ‘viswaroop,NNP’, ‘onetime,NN’ , ’watch,NN’, ‘sure,NN’.
4. Adjective Extraction: Adjectives remain in the corpus gets extracted.
5. Senti Score Evaluation: The words remain in the corpus are given positive, negative scores using SentiWordNet.After that, Overall average is taken out and the opinion score will be obtained.

B. *Module 2*

Our next process is to rank the data using Machine Learning Techniques. Here we are going to classify the dataset with SVM and Naive Bayesian Classifiers using two different algorithms. Our proposed algorithm frank is compared with already existing count score algorithm with the above said classification models. The Comparison we are following is given below. Also, the performance is commented out on Results and analysis Section.

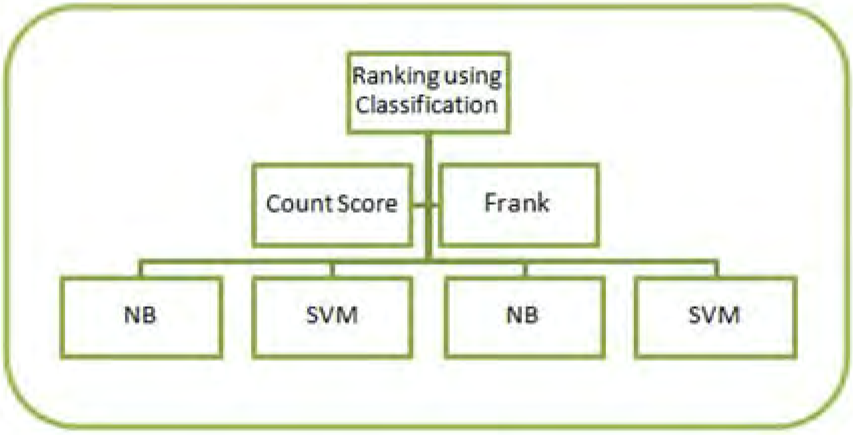

Fig.2. Ranking using Classification

1) Feature set:

We have taken most common words from the training data as feature set for polarity ranking.We have ranked these data based on the following two algorithms.

i) Count score- the weights are calculated by number of times the word appears in the document.
ii) FRank- It is based on the presence or absence of term denoting 1 or 0 respectively.

2) Frank Algorithm:

```
Split movie_reviews
    Set in doc_words
 Remove rare_words (doc_words)
      For   all word in doc_words do
          If word match with stop word set
                    Remove word from doc_words
                    Set in document_words
 Define feature set
      For every word in feature_set do
           If word in feature_set
                 Add 1 with existing term presence
                    Else
                 Add 0 with existing term presence
 Write in document_words
```

III. CLASSIFICATION PROCESS

We have implemented two commonly used machine learning techniques namely naïve Bayesian and SVM but with two different algorithms.

1. A naïve Bayesian classifier is a basic probabilistic classifier, which is based on bayes theorem with independence assumptions. Generally it requires a small set of data to calculate its parameters. This classifier maps a number of input features to a set of classes. This model says that all the features are fully independent. Consider classifying text by their content, say correct and incorrect data. Let text be taken from a number of set of documents denoted as sets of words where the possibility of occurrence that the ith word of a given text occurs in a text from set *C* can be written as

$$p(w_i|C)$$

Then the possibility that a given document T having all the words $\mathbf{w_i}$, given a set *C*, is

$$p(T|C) = \prod_i p(w_i|C)$$

2. SVM: Here we have considered using the linear SVM classifier. If an input is given to a linear SVM classifier, it predicts to which of the two possible classes the input belongs and gives an output accordingly. i.e., it predicts whether a new example falls into any one of the two categories. With high dimensionality of data, it gets easier to separate. It builds a set of hyper planes in a high dimensional space, which is useful for classification. A linear SVM is framed by a set of support vectors and weights. Let them denoted by **s** and **w** respectively. The output with N support vectors $s_1, s_2, \ldots, s_N$ and weights $w_1, w_2, \ldots, w_N$ will then be given by,

$$F(x) = \sum_{i=1}^{N} w_i (s_i, x) + b$$

IV. COMPARISON OF SVM AND NB

The Naive Bayes Classifier is a popular algorithm because of its simplicity, fast computational efficiency. It is easy and fast to train and classify. It is very well suited to problems involving normal distributions. But, it does not hold good for large dataset .i.e. it fails to solve a complex problem. Its main drawback is that it can't learn ambiguous features. Now coming to SVM, it is relatively slow to train and evaluate. Also, it is sensitive to noisy data. Advantages are high accuracy, great computational efficiency. But for a huge dataset they cannot be run on a normal computer with a strong processor.

V. RESULTS AND DISCUSSION

This section deals with the comparative analysis of different instances in both the classification models namely Naïve Bayesian and SVM. The analysis includes four types of measurement namely Accuracy, Precision, F1-Measure, and Recall .Table 1 and 2 includes the values of above said measurement.

A. Tabular description

TABLE I
USING COUNT SCORE

Corpus Taken	Classification model	Accuracy	Precision	Recall	FMeasure
N=40	NB	0.76	0.76	0.80	0.78
	SVM	0.87	0.53	0.50	0.51
N=100	NB	0.75	0.77	0.80	0.76
	SVM	0.87	0.47	0.44	0.45
N=400	NB	0.70	0.72	0.80	0.76
	SVM	0.87	0.48	0.46	0.46

TABLE II
USING FRANK ALGORITHM

Corpus Taken	Classification model	Accuracy	Precision	Recall	FMeasure
N=40	NB	0.77	0.75	0.80	0.77
	SVM	0.90	0.53	0.60	0.56
N=100	NB	0.79	0.70	0.76	0.73
	SVM	0.86	0.50	0.55	0.58
N=400	NB	0.79	0.72	0.77	0.74
	SVM	0.90	0.52	0.55	0.58

B. Pictorial Representation

The above said values are imported in the chart for further analysis.

1. Fig.3 gives the performance of already existing count score approach. We have taken three different instances, Say N=40,100 and 400. Though Naïve Bayesian stands good and gives a good rank SVM tops the three cases, performance falling in 88 - 90 in all the cases.
2. Fig.4 provides an overview of our proposed algorithm Frank. In our approach, both Naïve Bayesian and SVM provide a very good performance as this overcome the drawback of count score approach. Both NB and SVM gets high performance with our Frank algorithm. Also, in this SVM does the best performance.
3. Fig.5 is a comparison of accuracy between the existing and the proposed algorithm. It can be seen clearly that accuracy almost remains the same with near cases like N=40 and 100.But When it increases it gets even more accuracy in both NB and SVM. While comparing the performance of both the classifiers with the pre-process of proposed FRank and existing Count Score, Frank provides maximum accuracy both in NB and SVM. Among these two classifiers maximum accuracy is attained by SVM for movie domain dataset.
4. Though SVM lacks efficiency than NB in determining precision, recall and FMeasure, it has got an optimal solution for bringing on high accuracy. This can be seen in fig.4

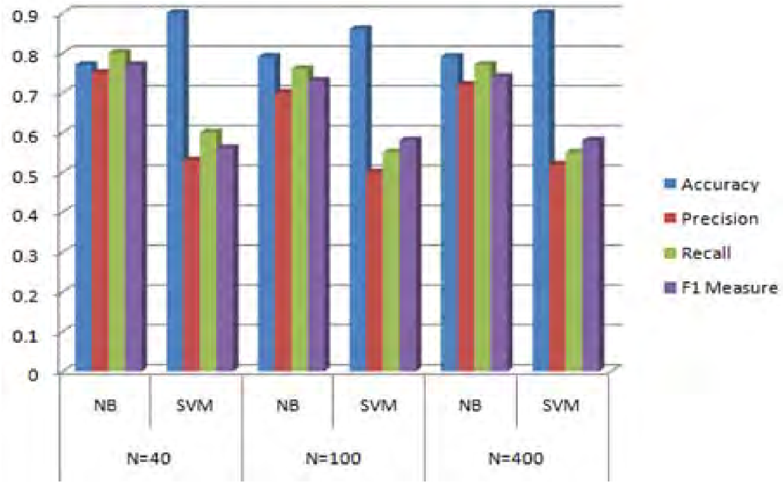

Fig.3. Count Score Approach

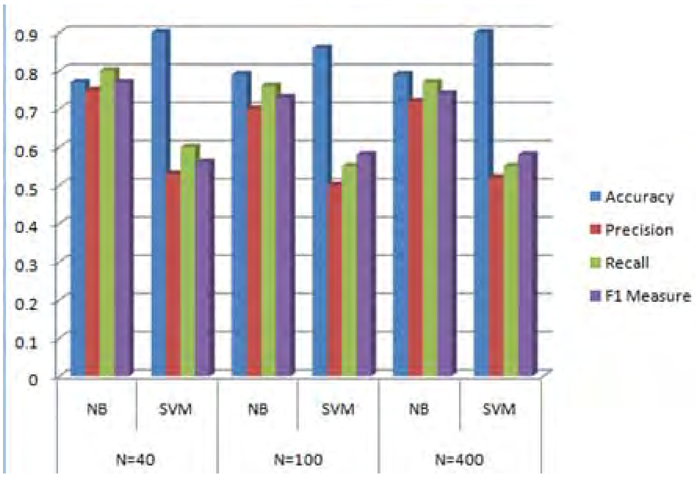

Fig.4. Frank Approach

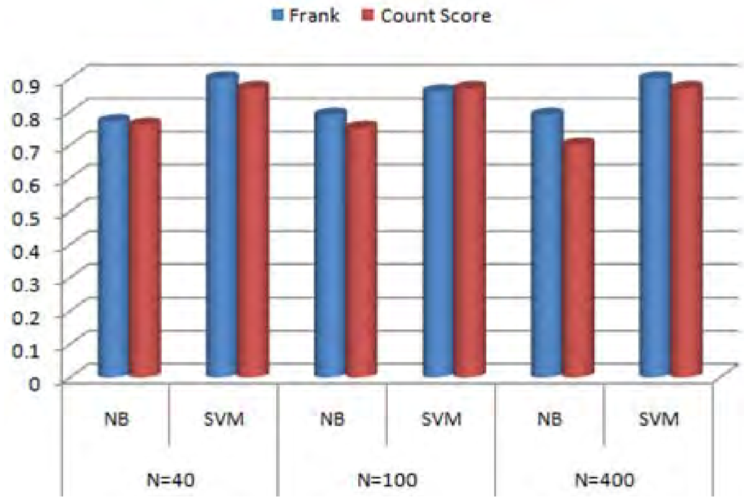

Fig.5. Accuracy Difference between existing and proposed method

CONCLUSION

This paper gives an overview of both automated opinion detection system and ranking using different classification models. We have also worked on comparative analysis of different classification models with different algorithms. Also, maximum accuracy is obtained in our proposed ranking algorithm Frank in both the classification models naive Bayesian and SVM than the existing approach. Coming to detection of opinion mining system, though this is a simple process, we can say it will be even more efficient if handling negation is given more preference. Also, it takes quite a long time for these processes. Our work still continues to reduce the timing constraint.

REFERENCES

[1] Li Chen, Luole Qi, Feng Wang “Comparison of feature-level learning methods for mining online consumer reviews” *Expert Systems with Applications, Volume 39, Issue 10, August 2012, Pages 9588-9601*

[2] Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, Yuxia Song “Mining comparative opinions from customer reviews for Competitive Intelligence” *Decision Support Systems, Volume 50, Issue 4, March 2011, Pages 743-754*

[3] M. Rushdi Saleh, M.T. Martín-Valdivia, A. Montejo-Ráez, L.A. Ureña-López “Experiments with SVM to classify opinions in different domains ” *Expert Systems with Applications, Volume 38, Issue 12, November–December 2011, Pages 14799-14804*

[4] Qingliang Miao, Qiudan Li, Ruwei Dai “AMAZING: A sentiment mining and retrieval system” *Expert Systems with Applications, Volume 36, Issue 3, Part 2, April 2009, Pages 7192-7198*

[5] Daniel E. O'Leary ”Blog mining-reviews and extensions: From each according to his opinion” *Decision Support Systems, Volume 51, Issue 4, November 2011, Pages 821-830*

[6] Huifeng Tang, Songbo Tan, Xueqi Cheng “A survey on sentiment detection of reviews” *Expert Systems with Applications, Volume 36, Issue 7, September 2009, Pages 10760-10773*

[7] Bo Pang, Lillian Lee, and Shivakumar Vaidyanathan”*Sentiment classification using machine* learning techniques”. Proceedings of EMNLP, pp. 79--86, 2002

[8] Magdalini Eirinaki, Shamita Pisal, Japinder Singh ”Feature-based opinion mining and ranking” *Journal of Computer and System Sciences, Volume 78, Issue 4, July 2012, Pages 1175-1184*