computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

NOVEL PRE-PROCESSING FRAMEWORK TO IMPROVE CLASSIFICATION ACCURACY IN OPINION MINING

Helen Josephine V. L. [1], Duraisamy S. [2]

[1] Research Scholar, Bharathiar University, Coimbatore,
Department of Computer Applications, CMRIT, Bangalore,
helenjose.cbe@gmail.com, https://sites.google.com/a/cmrit.ac.in/helen-josephine-v-l/
[2] Department of Computer Science, Chikkanna Government
Arts College, Tirupur, sdsamy.s@gmail.com, http://cgac.in/duraisamy/

Paper history:
Received 17 July 2018
Received in revised form 2 October 2018
Accepted 26 October 2018
Available online 31 December 2018

Keywords:
Pre-processing framework;
Opinion Mining;
Machine Learning;
Neural network.

Abstract: The growth of information technology led to the Internet development that in turn helped people in many ways. The major one is to express their views about the products and services through reviews, blogs, feedback, and comments on the website and in social media. The buyers are forced to go through investigation on these reviews/blogs, before choosing any product or service. Out of all online services, Mobile learning app places a vital role to increase the thirst for knowledge. But to identify the suitable mobile learning app, the opinions of the existing customers need to be mined. This research paper analyzes the mobile learning reviews which are available in the corpus. A novel preprocessing framework is proposed in this paper to improve classification accuracy in the dataset - mobile learning app review dataset. The corpus dimension is reduced using SVD through which, the data is prepared for mining. The classification accuracy is evaluated by applying Multinomial Naïve Bayes, Random Forest data mining algorithms and Learning Vector Quantization (LVQ), Elman Neural Network (ENN), Feed Forward Neural Network (FFNN) algorithms with the dataset obtained by the proposed processing method.

1. INTRODUCTION

Online learning has increased because of the availability of high bandwidth wireless expertise. The terrific development in the technology made the people express their views about the product and services in online platforms. Since the plentiful reviews are present on the Internet, the reviews/ blogs/feedbacks and comments need to be analyzed critically before choosing any product or service. Opinion mining is the useful research area that mines the product/service reviews state whether the opinions are positive/ negative/neutral. There are two broad categories of textual information namely subjective and objective. Objective statements reveal the facts of the product/services, but subjective statements exhibit the opinions and sentiments of the customers on product/service.

Application of natural language processing (NLP) helps to extract subjective information from the sources. NLP is suitable for document classification based on the topic, whereas, for opinion classification, opinion word and features have to be identified for the inspection. Supervised Machine learning methodology allows us to automatically learn the rules, efficiently form the training data and attempt to predict the reviews category whether it is positive/ negative/neutral. The accuracy of the result is based on the data quality. Since the real data which is obtained from the Internet contains errors, noise, ambiguity, duplicate and irrelevant information, cleaning or preprocessing the real data is essential to get the desired result.

Quality of input data determines the quality of the result, therefore, the preprocessing step is important and vital [1]. The major and important role of the preprocessing technique is used to remove noise and irrelevant data from the set. Stop word removal and stemming are the usual preprocessing steps of forming the corpus in text mining field [2].

In this paper, a novel preprocessing framework is proposed using basic and advanced methods to clean and filter that allows us to increase classification accuracy. It also compares the accuracy rates between existing preprocessing techniques and the proposed preprocessing methods with different classification algorithms. The research shows that classification of opinions does not only depend on opinion words and features words but also the corpus words which are commonly used in the review documents [3, 4]. This paper is organized into the following sections. Section II explains the methods and concepts in the proposed preprocessing algorithm, section III describes the results obtained and discusses them. Finally, section IV concludes the paper.

2. METHODOLOGY

Preprocessing is an essential and acute step in Text mining and sentiment analysis. The accuracy and effectiveness of the mining system are based on perfect preprocessing methods. It is crucial to select the influential and definite keywords that carry the meaning and impact the result and eliminate the words that do not contribute to differentiating between the documents. The pre-processing phase helps to conduct the study of the reform of the original unstructured textual data in a data mining ready structure [5]. An excellent outcome after applying data mining is based on an appropriate data preparation in the beginning. Important elements of the original data have to be detected and filtered out for further analysis. Unimportant and meaningless data need to be removed.

2.1. PRE-PROCESSING METHODS

1) Basic steps
    - Change all the collected reviews into lower case
    - Remove punctuations and extra white spaces
    - Remove repeated letters in a word like 'gooooooooooood', 'gooddddddddddddddd' for 'good'. Even though the lengthy words detect the strength of the opinion, these words are brought to the correct spelling.
2) Stopword Removal
    - Remove stop words like ’is’, ’the’, ’that,’ etc.
3) Stemming with spell check, Lemmatization
    - Stemming helps to match the similar words in a text document
    - Lemmatization supports a morphological analysis of the words.
4) Document Indexing
5) Inverse document frequency (IDF)
6) Outlier Removal

2.1.1 BASIC STEPS

(i) The collected review may contain upper case, lower case or sentence case words. First, the original text data should be converted into lowercase. Stop word removal and spell check tools input data are in the form of lower case. To get good result changing the case to lower is very essential.

(ii) Punctuations and extra white spaces will be removed in the next step. Since these are useless in text mining and opinion mining.

(iii) Removing repeated letters in a word should be brought into correct spelling

2.1.2. STOPWORD REMOVAL

Some words in reviews have nothing to do with the product or any polarity word. Most generally used words in English are ineffective in opinion mining. These words are known as Stop words. Stop words are language-related functional words which carry no meaningful information for mining [6]. They may be of the following types such as pronouns, prepositions, conjunctions. Some parts of speech in English such as pronouns, conjunctions, and prepositions have belonged to this category. Those words should be removed from the documents. Stopword dictionary contains all these words [10].

2.1.3. STEMMING WITH SPELL CHECK AND LEMMATIZATION

Stemming is one of the pre-processing methods used to find out the root/stem of a word. Stemming transform words into their stems. Stemming is powerful since the same stem or word root mostly describe the same or relatively close concepts in the text. So, the same stem or word may be discarded to reduce the dimension of the document term matrix [11]. For example, the words: material, materially, materialize, materialization, materialize, materiality all can be stemmed to the word 'MATERI'. The main goal of this method is to eliminate various suffixes, to decrease the number of words in order to have accurately matching stem. Different stemming algorithms are available, but the M.F Porters Stemmer algorithm [12] is the most commonly used algorithm in English. However, in the above examples, the correct form of the stemmed word is ‘MATERIAL'. But the Porters algorithm gives the word ‘MATERI'. To eliminate these problems, spelling is also checked after the stemming process [13].

Lemmatization is the key to this methodology in linguistics. To extract the proper lemma, it is necessary to look at the morphological analysis of each word. This requires having dictionaries for every language to provide that kind of analysis.

Example: For the word ‘studies'
Lemma will be ‘studies’. (correct)
The stem will be ‘studi’. (wrong)

2.1.4. DOCUMENT INDEXING

The main purpose of document indexing is to increase the efficiency. A selected set of terms will be used for indexing the document. Document indexing consists of choosing the appropriate set of keywords based on the whole corpus of documents and assigning weights to those keywords for each particular document [14]. So, each document is transformed into a vector of keyword weights. The weight normally is related to the frequency of occurrence of the term in the document and the number of documents that use that term.

2.1.5. INVERSE DOCUMENT FREQUENCY (IDF)

The Term Document Frequency is computed for a set of given review documents D_i=1 to n = $d_1, d_2, d_3, \ldots\ldots d_n$ and $t_1, t_2, t_3, \ldots\ldots\ldots t_j$ set of terms t_j j=1 to m. The term frequency is denoted by freq (d_i, t_j) representing the number of occurrences of term t_j in the document d_i i = 1 to n. The term-frequency matrix TF(d_i, t_j) measures term t_j association with regard to the ven document d_i and has a value of zero on document term for non-occurrence or a number otherwise[15, 16]. The number can be set as TF(d_i, t_j) = n, when term t_j appears in the document d_i or when a relative term frequency is used.

$$TF(d_i, t_j) = \frac{\text{No of Times term } t_j \text{ appears in a document } d_i}{\text{Total number of terms in the document } d_i}. \quad (1)$$

Table 1. Document term frequency matrix (DTFM)

Document/Term	*t*1	*t*2	*t*3	*t*4	*t*5	*t*6	*t*7	*t*8
*d*1	0	5	12	6	0	2	0	0
*d*2	4	7	0	19	0	1	20	11
*d*3	12	15	4	0	15	12	17	12
*d*4	21	4	9	6	12	2	3	0
*d*5	0	3	7	2	9	8	9	9
*d*6	0	12	15	2	7	5	4	11

Term frequency TF((d_i, t_j) is the total number of a term t_j in document d_i. Superior value of a Term Frequency shows the term t_j importance in a given document d_i. Terms presented in several documents were suppressed as these tended to stop words. The second component IDF (Inverse Document Frequency) handles this control:

$$IDF(t_j) = log\frac{|D|}{DF(t_j)}, \quad (2)$$

where

t_j = i_{th} term.
$|D|$ = the total count of documents.
$DF(t_j)$ = count of documents that contain term t_j.

If a term is present in all the documents, then numerator and denominator are equal in equation (2). In such case, the result of this $IDF(t_j)$= *log* 1, which is zero. But if the term presents relatively less number of the document, then $DF(t_j)$ < $|D|$. As a result, $IDF(t_j)$ = *log* (>1), which is a positive integer. Term presence vector was used in the calculation of *IDF*. TF-IDF identified important terms in given set of documents but as per Martineau and Finin top ranked index terms were not the top-ranked sentimentally polarized terms.

2.1.6. OUTLIER REMOVAL

To remove common and uncommon words, the product of Term Frequency and word importance is performed in the word vector. Then exclude the word which has product value less than 25 and greater than 75. Outlier removal consisting of the range of values shows the probability of the range that captures the true population. In order to achieve the confident values below 25 and above 75 are deleted from the word vector.

2.2 DATASET & FRAMEWORK

The proposed algorithm is applied for Mobile Learning system app reviews as data is downloaded from online. This dataset consists of reviews of the users from the Android Market Website, about the Mobile Learning system. In this research, mobile learners' opinions about the learning system app were considered. Three different types of reviews such as positive, negative and neutral are chosen for the analysis.

There are 300 reviews collected randomly from the website. Each reviewed document contains some stop words, numbers, non-alphabet characters, and vocabularies. Before training, the review documents are pre-processed by the existing pre-processing techniques and proposed novel pre-processing techniques. The result of the existing pre-processing method is called as Corpus I and proposed pre-processing method is called Corpus II.

Implementation is done in Python using Anaconda. Anaconda is the package, which provides Python and most of the libraries, which are used for machine learning pre-installed. Scikit learn library has been used, which contains the variety of machine learning library for the Python programming language.

The proposed pre-processing methods contain the subsequent steps: (i) Basic cleaning methods, (ii) Stop word removal, (iii) Stemming with spell check and Lemmatization, (iv) Document Indexing and Inverse document frequency to identify the rare terms than the usual terms, (v) The product of Term Frequency and Singular Value Decomposition of word importance, (vi) the outlying values removing the values between 25 and 75, are discarded in proposed preprocessing. The results of the proposed preprocessing method are called Corpus II.

The above-mentioned two corpora I and Corpus II) are further analyzed with classification algorithms, namely, multinomial naive Bayes, Random forest and three more neural network classification algorithms LVQ, Elman and FFNN. The respectively obtained results of the corpora are compared to note the accuracy and classification measures among the existing and proposed preprocessing methods. The framework of the proposed algorithm flowchart is represented in Fig. 1.

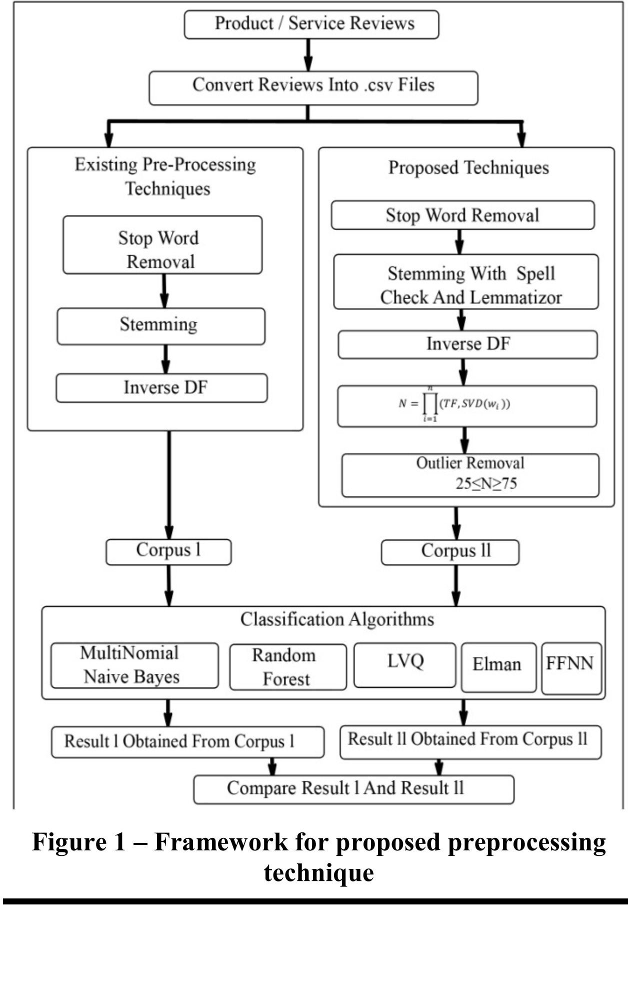
Product / Service Reviews
Convert Reviews Into .csv Files
Existing Pre-Processing Techniques
Stop Word Removal
Stemming
Inverse DF
Proposed Techniques
Stop Word Removal
Stemming With Spell Check And Lemmatizor
Inverse DF
$N = \prod_{i=1}^{n}(TF, SVD(w_i))$
Outlier Removal 25≤N≥75
Corpus I
Corpus II
Classification Algorithms
MultiNomial Naive Bayes
Random Forest
LVQ
Elman
FFNN
Result I Obtained From Corpus I
Result II Obtained From Corpus II
Compare Result I And Result II

Figure 1 – Framework for proposed preprocessing technique

Mobile learning system reviews are retrieved from online. The word vector ‘w' is created, which contains the term frequency with respect to its frequency of occurrence. In the basic step of preprocessing punctuation removal, spell check, stop word removal like ‘are' and ‘the'(stopword list) and stemming of words (i.e., bringing the word to the base form) are implemented.

The word frequency and word importance are calculated by SVD [14, 15]. To eliminate frequent and common words, the product of Term Frequency and word importance is performed in the word vector. Next step has excluded the word which has word count less than 25 and greater than 75. Outlier removal consisting of the range of values shows the probability of the range that captures the true population. In order to achieve the confident values the range between (<=25) and (>=75) are deleted from the word vector [16]. This proposed algorithm facilitates creation of new and clean data set to improve the accuracy of the classification algorithm.

Common and uncommon words will be identified through term frequency matrix and single value decomposition method. The product of TF and SVD will identify the outlier words. These outlier words will be considered as extra noise for the data set. Along with the existing pre-processing methods stemming with spell check and outlier word removal are considered as novel methods which have included in the proposed framework and yield better classification performance.

3. RESULTS AND FINDINGS

In the present work, a novel pre-processing framework has been proposed. It includes the existing techniques like stopword removal, stemming, SVM, TF-IDF and outlier removal based hybrid framework model is presented to process the mobile learning app reviews to classify the opinion. Implementation is done in Python using Anaconda. Anaconda is the package which provides Python and most of the libraries, which are used for machine learning pre-installed. Scikit learn library has been used which contains the variety of machine learning library for the Python programming language. It also provides the results in terms of classification accuracy, confusion matrix, precision, recall, and f-measure etc. Description of the implementation is shown in Table 2.

Two data mining classifiers multinomial Naive Bayes, Random Forest and three neural network classifiers accuracy are acquired and compared. To train the algorithm, a tenfold cross-validated method has been used. The efficiency of Multinomial Naïve Bayes, Random forest, LVQ, Elman neural network, and FFNN classifiers are compared for further analysis. Pre-processing techniques and feature extraction play the crucial role to obtain the accuracy in the classification algorithm:

Table 2. Description of Implementation

Features		Values
No. of reviews		300 (100- Positive, 100- Negative and 100- Neutral)
Training and Testing Data		Cross-validation, Random State = 1
Review classes		Positive, Negative, Neutral
Data Mining Classifier Applied to Corpus I & II	•	• Multinomial Navie Bayes, • Random Forest
Neural Network Classifier applied to Corpus I & II	•	• Linear Vector Quantization Neural Network (LVQ), • Elman Neural Network(ENN), • Feed Forward Neural Network(FFNN)

$$Classification\ Accuracy = \frac{No.of\ reviews\ classified\ correctly}{Total\ No.of\ reviews}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} * 100\,. \qquad [17]$$

The bar graph representation of the values is shown in Fig. 2 and Result Obtained with Data Mining Classifiers is shown it Table 3.

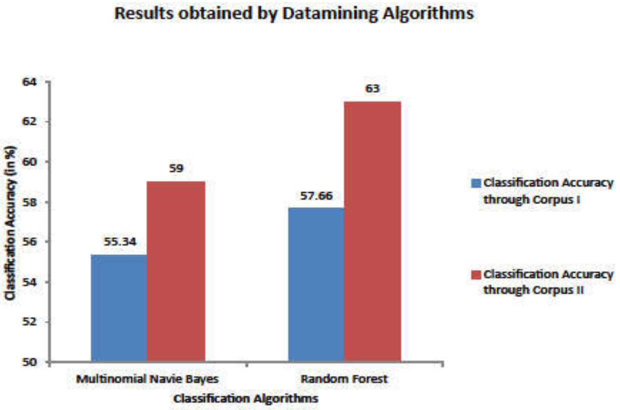

Figure 2 – Classification Accuracies with Data Mining Classifiers

Table 3. Result Obtained with Data Mining Classifiers

Different Data mining Classifiers	Classification Accuracy through Corpus I (in %)	Classification Accuracy through Corpus II (in %)
Multinomial Naive Bayes	55.34	59
Random Forest	57.66	63

1) The Linear Vector neural network algorithm has been used in opinion mining to produce the good results [18]. The classification accuracy for LVQ in Corpus I is 54.33%. But classification accuracy for LVQ in Corpus II (proposed new preprocessing techniques) is 60.67%.
2) Elman neural network is considered as feedforward networks added with the layer of recurrent connections with time delay structures [19, 20]. The classification accuracy for ENN in Corpus I is 72% whereas in Corpus II is 79%.
3) The feed-forward neural network is the simple and trouble-free artificial neural network developed in [21]. In FFNN, the data flows in only one direction, i.e., starting from the input nodes to the output nodes through hidden nodes (if they exist). The network will not allow any cycles or loop [22]. The summation and sigmoid activation function is used to calculate the output values. The process is iterated till the threshold value is obtained or the number of epochs it reached. This FFNN algorithm has been applied to the Corpus I and II dataset, which produce the classification accuracies 77 and 85 respectively. Parameters used in the FFNN algorithm are given in Table 4.

Table 4. Parameters used in FFNN

Parameters	Values
Number of neurons in the input layer	60
Number of neurons in hidden layers	30
Number of neurons in output layers	1
Number of Hidden layers	1
Number of epochs	500
Activation function	Sigmoid
Learning rate	0.1
Momentum	0.5

By testing with three different algorithms, it is revealed that the classification accuracies attained through proposed preprocessing method had improved the mean by 7.00%. Classification accuracies of three different neural network based classifier with the two corpora (Corpus I and Corpus II) are tabulated in Table 5. Fig. 3 diagrammatically shows the classification accuracies for the three neural network algorithms.

Table 5. Classification Accuracy Obtained by Neural Network Classifiers

Different Neural Network Classifiers	Classification Accuracy through Corpus I (in %)	Classification Accuracy through Corpus II (in %)
LVQ	54.33	60.67
Elman Neural Network	72.33	79.67
FFNN	77	82.33

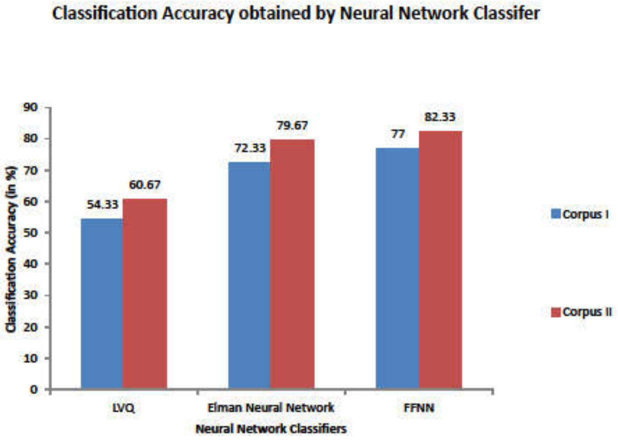

Figure 3 – Classification Accuracies Obtained through Neural Network Classifiers

Fig. 2 and Fig. 3 show the classification accuracy result of the data mining classifier multinomial naive Bayes and random forest and neural network classifier LVQ, Elman NN and FFNN with existing preprocessed methods and proposed processing methods. It graphically proves that the Corpus II, which resulted by proposed preprocessing methods has higher classification accuracies than Corpus I, which resulted by existing preprocessing methods.

Classification Accuracy alone is not a sufficient metric to examine the effectiveness of classifiers. Along with classification accuracy, Precision, Recall, and F-measure, their proximities need to be calculated and identified. Following formulae are used to calculate the precision, recall, and F-measure:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive},$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Positive},$$

$$F - Measure = \frac{2 * Precision * Recall}{(Precision + Recall)}.$$

Precision and Recall are the other metrics that can provide much greater insight into the performance characteristics of a binary classifier. Precision is inversely proportional to false positive, i.e., precision measures the exactness of a classifier. A higher precision ensures less false positives, while a lower precision implies more false positives. But Recall is inversely propositional to the false negative. Higher recall means less false negatives, while lower recall ensures more false negatives. Improving recall can often decrease precision because it gets increasingly harder to be precise as the sample space increases. A single metric formed by combining, Precision, Recall, and F-measure, which is the weighted harmonic mean of precision and recall [23]. The F-measure reflects the relative importance of recall versus precision [24].

Table 6 shows the precision, recall, and F-Measure for various classification algorithms obtained by existing preprocessing methods. Table 7 illustrates different classification performance metrics, namely, precision, recall and F-Measure for various classification algorithms obtained by proposed preprocessing methods.

Table 6. Classification measures for various algorithms obtained by existing preprocessing methods

Algorithm	Precision	Recall	F-Measure
Multinomial Naïve Bayes	0.566	0.556	0.561
Random Forest	0.587	0.576	0.581
LVQ	0.553	0.543	0.548
Elman NN	0.724	0.723	0.723
FFNN	0.773	0.77	0.771

The tabulated values showing the precision, recall, and F-Measure, which are obtained through Corpus II yield better results compared to Corpus I.

Table 7. Classification measures for various algorithms obtained by proposed method

Algorithm	Precision	Recall	F-Measure
Multinomial Naïve Bayes	0.609	0.586	0.597
Random Forest	0.642	0.626	0.634
LVQ	0.621	0.613	0.617
Elman NN	0.812	0.8	0.806
FFNN	0.832	0.826	0.829

Classification performance metrics – Precision is tabulated in Table 6 for existing pre-processing methods and Table 7 for proposed pre-processing techniques. Fig. 4 shows the bar chart of precision values obtained through Corpus I and Corpus II by applying various classification algorithms. Fig. 5 shows the bar chart of recall values obtained by Corpus I and Corpus II by applying various classification algorithms. Fig. 6 gives the graphical representation of f-measure values obtained through Corpus I and Corpus II by applying various classification algorithms.

Figures 4, 5, 6 graphically prove that classification performance metrics precision, recall, and f-measure of Corpus II are higher than the Corpus I values which are obtained by the existing pre-processing methods. The novel proposed preprocessing framework includes the TF, SVD and outlier removal along with the other existing methods.

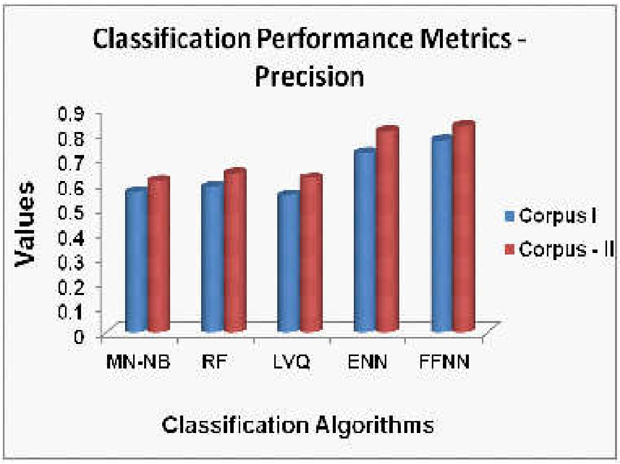

Figure 4 – Precision values of various algorithms obtained by existing and proposed pre-processing methods

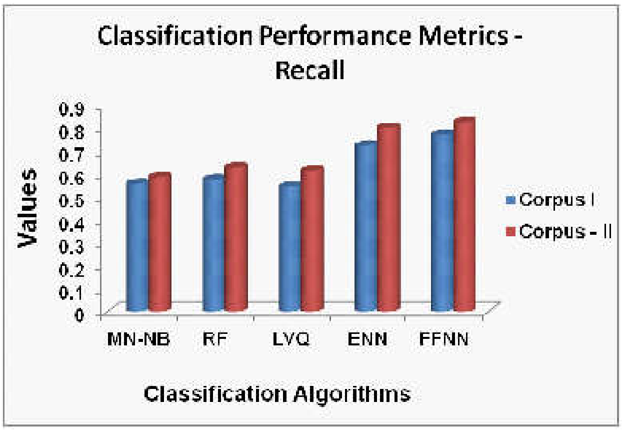

Figure 5 – Recall values of various algorithms obtained by existing and proposed pre-processing methods

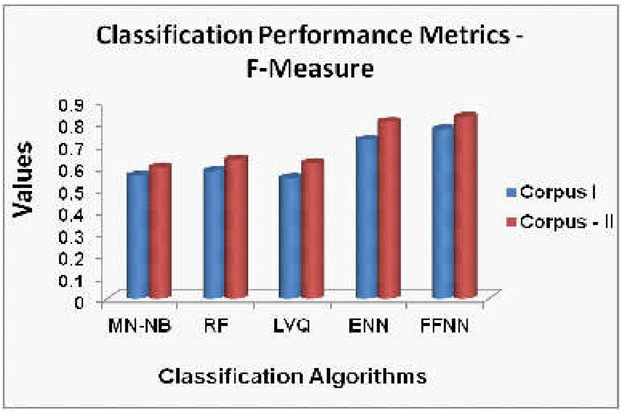

Figure 6 – F-Measure values of various algorithms obtained by existing and proposed pre-processing methods

4. CONCLUSION

In this paper, advanced data cleaning techniques and feature and opinion word extracted methods are proposed based on term frequency which it appears in all the documents. The document term matrix is prepared by removing irrelevant word, which is not useful to mine the opinion. SVD has been used to identify the outlier words and the document term matrix dimension has been reduced to improve the classification accuracy. Based on comparative analysis, Feed Forward Neural Network gives best classification accuracy among all classifiers. The classification accuracy for Neural network algorithm, namely LVQ, ENN, FFNN for Corpus I (existing method) and Corpus II (Proposed preprocessing methods) is listed in Table 5. Average 3.16% of classification accuracy has been increased with the implementation of the proposed preprocessing framework.

5. REFERENCES

[1] K. Gibert, J. Izquierdo, G. Holmes, I. Athanasiadis, J. Comas, M. Sànchezpost, "On the role of pre and post processing in environmental data mining," *International Environmental Modelling and Software Society* (iEMSs), 2008, pp. 1937-1958.

[2] S. Vijayarani, J. Illamathi, P. Nithya, "Preprocessing techniques for text mining – an overview," *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7-16, 2015.

[3] Q. Zhao, H. Wang, Pin Lv., "Joint propagation and refinement for mining opinion words and targets," *Proceedings of the IEEE International Conference on Data Mining Workshop*, 14 Dec. 2014.

[4] T. Jiang, M. Zhong, S. Shumei, S. Luo, "Mining opinion word from customer review," International Journal of Database and Theory and Application, vol. 9, no. 2, pp. 129-136, 2016.

[5] R. Talib, M. K. Hanif, S. Ayesha, F. Fakeeha Fatima, "Text mining: techniques, applications and issues," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, pp. 414-418, Nov. 2016.

[6] C. Silva, B. Ribeiro, "The importance of stop word removal on recall values in text categorization," *Proceedings of the IEEE International Joint Conference on Neural Network*, Porland, USA, 2003.

[7] K. Amarasinghe, M. Manic, R. Hruska, "Optimal stop word selection for text mining in critical infrastructure domain," *Resilience Week (RWS)*, 2015, pp. 1-6.

[8] C. Moral, A. de Antonio, R. Imbert, J. Ramírez, "A survey of stemming algorithms in information retrieval," *Information Research*, vol. 19, no. 1, March 2014.

[9] A. G. Jivani, "A comparative study of stemming algorithms," *International Journal of Computer Technology and Applications*, vol 2, issue 6, pp. 1930-1938, 2011.

[10] J. Singh, V. Gupta, "A systematic review of test stemming techniques," *Artificial Intelligent Review*, vol. 48, issue 2, pp. 157-217, August 2017.

[11] G. Slaton, C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Management*, 1988

[12] K. Ghag, K. Shah, "SentiTFIDF sentiment classification using relative term frequency inverse document frequency," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 2, pp. 36-43, 2014.

[13] L. Havrlant, V. Kreinovich, "A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation)," *International Journal of General Systems*, vol. 46, No. 1, pp. 27-36, 2017.

[14] R. Akemi Sinoara, J. Antunes, S. O. Rezende, "Text mining and semantics: a systematic mapping study," *Journal of the Brazilian Computer Society*, vol. 23, issue 9, pp. 1-20, June 2017.

[15] D. G. Rees, *Essential Statistics*, 4th Edition, Chapman and Hall/CRC, 2001.

[16] K. P. Nguyen, H. Q. Phan, "Feasible settings for the adaptive latent semantic analysis: Hk-LSA model," *Proceedings of the 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA'2017)*, 2017, pp. 219-224.

[17] Nanda S., M. Sukumar Detection and classification of thyroid nodule using Shearlet coefficients and support vector machine, International Journal of Engineering & Technology, Website: www.sciencepubco.com/index.php/IJET doi: 10.14419/ijet.v6i3.7705

[18] M. Stylianidis, E. Galiotou, C. Sgouropoulou, C. Skourlas, "Opinion mining using an LVQ neural network," Proceedings of the 21st Pan-Hellenic Conference on Informatics PCI 2017, Larissa, Greece, September 28 - 30, 2017, Article No. 61.

[19] J. L Elman, "Finding structure in time," *Cognitive Science*, vol. 14, issue 2, pp. 179-211, 1990.

[20] O. Irsoy, C. Cardie, "Opinion mining with deep recurrent neural networks," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 25-29, 2014, pp. 720–728.

[21] A. Zell, Simulation of Neural Networks, 1st ed., Addison-Wesley, 1994, 73 p.(in German).

[22] Jump up "Deep learning in neural networks: An overview," Neural Networks, no. 61, pp. 85–117. 2014. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003.

[23] P. Chaovalit, L. Zhou, "Movie review mining: a comparison between supervised and unsupervised classification approaches," *Proceedings of the 38th IEEE Hawaii International Conference on System Sciences*, 2015, vol. 4, pp. 112-115.

[24] D.M.W Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correction," *Journal of Machine Learning Technologies*, vol. 2, issue 1, pp. 37-63, 2011.

Helen Josephine V. L. *is a Research Scholar at Bharathiar University, Coimbatore, and she is working as an Assistant Professor in the Department of Computer Applications, CMRIT, Bangalore. Her research interest includes Machine Learning, Web mining, Opinion mining, and Sentiment analysis.*

Dr. S. DURAISAMY *is Assistant Professor of Department of Computer Science in Chikkanna Government Arts College. He obtained Ph.D. in Computer Science in 2008. He has produced 12 Ph.D. candidates and guiding many research scholars. He has published more than 80 articles in national and international journals. His area of interest includes Software Engineering, Software Testing and Data Mining.*