Exploiting RLPI for Sentiment Analysis on Movie Reviews

H K Darshan, Aditya R Shankar, B S Harish, and Keerthi Kumar H M
Sri Jayachamarajendra College of Engineering, Mysore, India
Email: {bharadwajdarshan, adityaravishankar2012, keerthiihm.pace}@gmail.com, bsharish@sjce.ac.in

***Abstract*—The rapid growth in internet usage has made people to share their opinions publicly. Public opinions generally influence the crowd to a great extent. It becomes important to analyze the sentiment expressed as opinion to derive useful conclusions. Sentiment Analysis (SA) on movie reviews deals with summarizing the overall sentiment of the reviews. In literature, many researchers worked on sentiment analysis on IMDb reviews by identifying relevant features and classifying the reviews. In this paper, we show that exploiting Regularized Locality Preserving Indexing (RLPI) as a feature selection method shows better results compared to other feature selection methods like Information Gain, Correlation and Chi Square when tested with classifiers like SVM, KNN and Naive Bayes. RLPI reduced the overall complexity by extracting discriminating features from the input data and improved classification accuracy.**

***Index Terms*—sentiment analysis, reviews, feature reduction, classification**

I. Introduction

Microblogging has become the current trend where people share their opinions on Social networking sites such as Facebook, Twitter, Tumblr, etc., based on their experience. These opinions may be in the form of text, video or images. A lot of service requests and product purchases are done based on the overall opinion of the crowd. Microblogs contain textual information from different domains such as real-time news, political reviews and advertisements [1] each having its respective emotions. Increase in the usage of internet is the main reason for people choosing microblogs over traditional blogging and article writing. Organizations that sell products or offer services can use microblogs on social media to find out the sentiment of end users and consider their feedbacks to improve the quality of their products and the services they offer. Hence, there is a need to analyze the overall sentiment of such data and this process is known as Sentiment Analysis [2].

Sentiment Analysis (SA) will become very effective in defining the overall polarity of product reviews, movie reviews or service feedback which may project wide range of emotions like Anger, Like, Dislike. Positive reviews increase the value of a product or service whereas negative reviews decrease the value due to negative responses. Hence, microblog textual reviews have become significant in defining product and service quality.

Sentiment Analysis in recent days is widely used by E-commerce websites [3] to identify the reviews provided by customers. Manufacturers and service providers use these reviews to improve the quality of their services and products. SA on social media is also used to know the opinion of the people towards a political party [4] and based on this; parties can use new strategies to influence on the public opinion.

SA of microblogs involves processing large amounts of unconventional text data. Bloggers use abbreviations such as “gud” instead of “Good”, “nyc” instead of “Nice” and it becomes a challenging task to handle such abbreviations. Apart from words, bloggers use special characters such as “:)”, “: D”, “:(” which influences the inclination of the overall sentiment. Usually, microblogs are written in various languages and it is difficult to analyze the sentiment of such blogs. Such drawbacks make the SA task much harder. Hence, researchers have proposed many Machine Learning techniques [5], Lexicon based approaches [6] and the combination of both [7] to improve the SA model. Lexicon based approaches capture the textual features like Parts of Speech (POS) tagging and textual ordering for SA and these features often contribute to the efficiency of SA models. However, Lexicon based algorithms fail to capture the high-quality information that is typically derived through the devising of patterns which is possible only by statistical pattern learning algorithms. Thus, Machine Learning algorithms are highly efficient and require preprocessing of data. This preprocessed data should have features that contribute to the overall accuracy of SA.

Mining of microblogs and reviews involves preprocessing that selects significant features. Data sparsity and high dimensionality increases the importance of Feature Engineering [8]. The efficiency of a SA task is dependent on how the textual data is represented for learning and classification. Hence, the number of features considered plays a major role in accurate classification of reviews. Feature extraction maps the input space onto a lower dimensional space, retaining relevant information only. Feature selection identifies the best subset among the available features and reduces the number of dimensions. As the number of features increases, it becomes difficult to handle the data in higher dimension. Hence, data complexity has to be reduced by eliminating

Manuscript received November 11, 2018; revised February 1, 2019.

doi: 10.12720/jait.10.1.14-19

few features that do not influence SA. The features that can be extracted from the data maybe redundant, relevant or irrelevant. It becomes a necessity to use the right feature selection algorithm that not only reduces complexity but also retains features that make significant contributions to classification accuracy. Conventional feature selection methods like Chi- Squared [9], Information Gain [10], Correlation [11] and Mutual Information [12] are some of the most commonly used. However, unconventional methods can be transformed into feature selection methods to improve the efficiency of feature selection.

In this paper, we use Regularized Locality Preserving Indexing (RLPI) [13] as a Feature Selection method which follows the principle of Locality Preserving Indexing (LPI) and selects the discriminating features among the entire feature space. RLPI decomposes the regular LPI problem as a graph embedding problem and in addition as a regularized least squares problem which is more efficient and helps in handling large matrices.

The contents of the paper are divided into five sections. Section 2 provides an overview of the previous works and their shortcomings. Section 3 presents the proposed system. Section 4 shows experimental results and the paper is concluded in section 5.

II. Literature Survey

Reviewers generally present summaries in their reviews. Hence, Sentiment Analysis on movie reviews is a challenging task since. The problem on SA can be seen as a general classification task where the reviews fall under Positive or Negative class. Sentiment Analysis based on Machine Learning approaches for IMDb reviews was proposed by Pang *et al.*, (2002) [14]. Classification of reviews based on the overall sentiment was done using Naïve Bayes, Maximum Entropy and Support Vector Machines (SVM) classifiers. Feature representation was done by using unigram, bigram, position of words and Parts of Speech (POS) tags. A prominent challenge in Sentiment Analysis is handling run-time data. Approaches for sentiment analysis on run-time data was proposed in the works of Chamansingh *et al.*, (2016) [15]. The authors briefed the use of Machine Learning Classifiers like Support Vector Machines (SVM) and Maximum Entropy for Twitter run time data. The work shows the significant reduction in data input and processing can be achieved by maintaining acceptable level of accuracy for run-time data.

Preprocessing the text data has been considered as one of the important methods to improve classification accuracy. Also, feature extraction and selection methods can be exploited to improve the classification accuracy and reduce the overall complexity. The work proposed by Ghosh *et al.*, (2017) [16] highlighted the importance of preprocessing by using camera reviews for text Sentiment Analysis. Data preprocessing was done by tokenizing, stop word removal and stemming. SVM, Maximum Entropy and Naïve Bayes classifiers were used for classification. In this work, SVM achieved best results among all of them. Chakrit *et al.*, (2016) [17] briefed the importance of reducing the dimension of the data during preprocessing. Chi- square feature selection method was used for feature selection and the use of vote ensembled machine learning gave better results compared to existing machine learning approaches. Use of hybrid approaches in selecting a good feature subset was proposed by Yang *et al.*, (2015) in [18]. Sentiment lexicons and unigrams having high information gain were used as feature. The model was trained with six different classifiers of which Naïve Bayes Multinomial (NBM) showed good accuracy.

Next set of literature presents the importance of representing data or reviews. Tripathy *et al.*, (2016) [3] described the of use n-gram representation for the features. Term Document Matrix (TDM) was created by using n-gram representation of words and used different classifiers such as Naïve Bayes, Maximum Entropy, Support Vector Machine (SVM) and Stochastic Gradient Descent to train the model. The model showed best results for SVM by combining unigram, bigram and trigram representation of reviews. Sahu *et al.*, (2016) [19] focused their task on IMDB dataset to classify the polarity of the movie review and performed feature extraction and ranking. Apart from the conventional N-gram word representation, 10 extra features were selected based on the polarity of the words in a review. Some of the non-conventional features used were Positive and Negative Sentiment words coupled with Adjectives, Positive and Negative Sentiment words with repeated letters. Information Gain was used to select features from the N-gram representation of reviews.

Selection of relevant features for higher classification accuracy was proposed by Trivedi *et al.*, (2016) in [20]. The main objective of the work was sentiment analysis of Indian movie reviews. Different feature selection methods like Chi-square, One-R, Gain-Ratio, Info-gain and Relief Attribute gave them good F-measure values and False Positive count. In order to reduce the computation complexity involved during feature extraction, Gao *et al.*, (2015) [21] used Chi-Squared and Pointwise Mutual Information (PMI) for feature selection. Experimentation was done on two different corpuses for SA- microblogs and E-commerce data to evaluate performance and have emphasized the importance of feature selection for good results. Apart from the conventional feature selection methods that exists in the literature, Tuba *et al.*, (2016) [22] proposed a new feature selection method called as Query Expansion Ranking. This method showed higher classification accuracy on 3 different Turkish review datasets when compared to Chi squared and Document Frequency Difference methods. Regularized Locality Preserving Indexing (RLPI) was proposed by Cai *et al.*, (2007) [13] as a modified approach of Locality Preserving Indexing (LPI) for a better representation of text documents. Decomposition of document representation using Eigen vector decomposition and using least square problem to select top vectors to represent document space makes RLPI more efficient and handle large matrices. Similar works are proposed by Harish *et al.*, (2016) [23], where RLPI as a feature selection technique for a large Term Document

Matrix (TDM) was used for text clustering. The results presented were highly encouraging.

Use of robust feature selection methods for sentiment analysis and complexity reduction is an open challenge. Researchers have been experimenting on different combinations of features selection methods. This highlights the importance of feature selection in SA. In this paper, we exploited the advantages of RLPI feature selection method on movie review dataset. Further, the advantage of RLPI is also compared with conventional feature selection methods. The experimental results were highly promising in terms of both reductions in complexity of data representation as well as improve in the classification accuracy.

III. METHEDOLOGY

In our proposed work, we consider the task of Sentiment Analysis as "Binary Classification" problem as each review is mapped to either positive or negative sentiment. The experimentation mainly comprises of following stages: Pre-Processing, Representation, Feature Selection and Classification. Fig. 1 gives the overview of our proposed work.

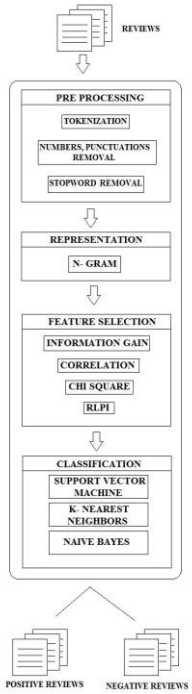

Figure 1. Proposed model

A. Pre-Processing

Data pre-processing is cleaning the data for a better representation and eliminating noise. This segment is crucial for SA and there are multiple stages involved. There are 3 steps in this stage- a) Tokenization b) Removal of numbers and punctuations and c) Stopword elimination.

The dataset has reviews that contains a mixture of numbers, special characters like '!', '?', ';' etc., and textual information. Not all such features contribute to SA and hence, only features that matter are taken into consideration and other features are eliminated. Removal of special characters and numbers is done after tokenization. Tokenization is parsing of textual reviews to split reviews into separate tokens or elements. These tokens maybe numbers, special characters or words from the review. After eliminating numbers and special characters, only words are left out in the dataset. The next step is to remove words that have no significance in influencing the overall sentiment of the reviews. These words are called stop words and the stop word removal of words like "a", "an", "the", "this", "will" etc., will reduce number of words or features that needs to be processed. Preprocessed textual data has to be represented in the form of a Term Document Matrix (TDM) for further processing.

B. Representation

In our work, we have represented the words in the form of a Term Document Matrix (TDM). TDM has words or features are columns and a weighing measure value for their presence in reviews. Words were considered as single features, which mean that each word or a unigram in the preprocessed word set is considered as a feature. Words like 'good', 'acting' and 'direction' are considered separately and this is an n-gram representation for n=1. Experimentation was also done on bigrams (n=2) where a combination of two words in the preprocessed word set is taken into account for weighting. This means that words are taken in pairs as a single feature as- 'good acting', 'good direction' and 'acting direction'. Further, weighting schemes like Term Frequency (tf) and Term Frequency Inverse Document Frequency (tf-idf) were used on n-gram representations. The tf-idf weight for a particular word is the product of tf and idf.

Here, $'C'$ is a set of reviews and $'w'$ is a word which is present in $'C'$ and $'N'$ be the total number of reviews in $'C'$. Then tf-idf can be calculated as follows:

$$tf - idf = tf \times idf \tag{1}$$

where,

$$tf = \frac{\text{Frequency of word in that review}}{\text{Total number of words in that review}},$$

$$idf = \frac{\text{Total number of reviews } (N)}{\text{Number of reviews in set of reviews } (C) \text{ that contain word } w}$$

C. Feature Selection

During this phase, we select the subset of features that has more importance when compared to other features. Different feature selection methods use different statistical measures to calculate the importance of each feature. Feature selection is done to reduce the dimension of the data. In our work, we have used Chi-Square, Information-Gain, Correlation and Regularized Locality Preserving Indexing (RLPI) feature selection methods. Chi-Square feature selection algorithm [24] tests the independence between two events. i.e., Occurrence of feature and Occurrence of class. The features with higher value of chi-squared are considered as the feature which is more correlated with the class. Information Gain

feature selection algorithm [25] considers individual feature and checks if removal of that feature affects the performance. The theory behind this algorithm is removing relevant feature from the feature set will negatively effect on the classification task. Correlation feature selection algorithm [26] calculates the correlation of a feature with respect to class and with respect to other features. The good feature subset contains features having high correlation with its class and uncorrelated with each other.

Use of RLPI [13] as a feature selection algorithm is one of the major contribution of this paper.

RLPI follows the principle of Locality Preserving Indexing (LPI) which is used to extract the most discriminative features by decomposition of eigen vectors. It is more time and space efficient compared to LPI as it avoids eigen decomposition of dense matrices, by transforming the same as graph embedding problem and in addition a regularized least square problem.

D. Classification

In the proposed work, we have used supervised learning algorithms such as: Support Vector Machines (SVM) [27, 15], Naïve Bayes [14, 15], K- Nearest Neighbor (KNN) [28] for classification. SVM finds a hyperplane that can separate two classes of given samples with a maximum separation margin between two classes. SVM performs well for both training and testing data for same distribution. Maxent is a probabilistic classifier which differs from traditional Naïve Bayes classifier by omitting the independence of feature. Maxent classifier maximizes the entropy of the system by calculating the conditional probability of the class label. KNN compares each review in the test data with the training data reviews that are most similar to it. It uses similarity measure such as Euclidean distance to find similarity. It first computes the distance of unknown record with training records. Further it identifies K-nearest neighbor for the unknown record. Finally use class labels of nearest neighbors to determine the class label of unknown record by using weight factor. Algorithm 1 presents the step by step procedure of the proposed method.

Algorithm 1: Proposed Method

Data: Set of IMDb reviews 'C' containing reviews $r_1, r_2, r_3, ..., r_n$. Class label $l \in \{Positive, Negative\}$ corresponding to the review in set 'C'.

Result: Class label 'l_u' for the new review 'r_u'.

Step 1: Preprocess the reviews using tokenization, numbers and punctuations removal, stopword elimination.

Step 2: Preprocessed reviews are represented using n-gram representation and the reviews are represented as matrix (TDM) where each row corresponds to a review and column corresponds to feature.

Step 3: RLPI feature selection technique is applied to the TDM by using equations (2) to (5) to get discriminating features.

Step 4: Features selected in step 3 is used to train the model with classifiers like SVM, Naïve Bayes and KNN.

Step 5: Given an unlabeled review 'r_u' from the test data, assign label 'l_u' to that review.

IV. Experimental Setup

A. Dataset

For our experiment, we used Internet Movie Database (IMDb) dataset. This dataset is publicly available [29]. The dataset contains equal number of positive and negative reviews which are labelled manually based on the ratings given to the movies. Here, 10% of the overall data (50,000 reviews) was used for experimentation. Data subset was chosen randomly out of all the samples and it is sufficient to design a SA model for this subset as it can be extrapolated to the entire dataset.

B. Experimentation and Discussion

Finding the best results takes a lot of experimentation on different combinations of parameters. The parameters that were used during experimentation were – number of features, classification algorithm, feature selection algorithm and the split ratio for train and test data.

A 5-fold cross validation was done on all combinations. That is 1000 samples were randomly picked from the dataset and SA was done on this set of samples. Same procedure was done on 4 other sets of 1000 reviews that were chosen randomly. The results obtained at each validation step were considered and an average of obtained values is tabulated.

The selected features were split into three different ratios for training and testing as per the following criteria. The experimentation was done with 50:50, 60:40 and 70:30 split ratios. We observed that 60:40 split ratio fetched results with the least accuracy irrespective of the classifiers, weighing measures and feature selection methods. However, a lower split ratio of 50:50 for training and testing showed improvements up to 10% increase in accuracy and up to 0.2 increases in precision and recall values. The experiment was tested on 1-gram and 2-gram representations also where the results were better for unigram representations for all combinations of classifiers and feature selection methods. We achieved the best results for a 70:30 split ratio in terms of accuracy and F-measure. The 50:50 and 70:30 split ratio results are almost equal but 70:30 ratio maintains consistency in accuracy for all the classifiers used.

Now that the split ratio of 70:30 was considered best, various combinations for feature selection methods was done.

Feature selection methods that were used in this experiment are Chi-Square, Correlation, Information Gain and RLPI. Each of these algorithms is used to classify the polarity with all the split ratios. However, each feature selection algorithm performed differently i.e., resulted in different number of reduced features. RLPI usage gave good results for a very small feature subset which was around 50-150 features. The feature variation for other feature selection methods were also done for a

range of values starting from 1000 to 8000 with an increment of 1000 features every time. The best result for these features selection methods were obtained for 2000, 5000 and 8000 features and the best set of features for RLPI was 50, 100 and 150 features respectively.

During the experimentation, RLPI was found to reduce complexity, efficiency of classification using all the above-mentioned classifiers. SVM achieved similar accuracy values across all the split ratios. Maxent showed better results for all feature selection methods and across all split ratios. KNN was experimented with 4 different values of K: 3, 5, 7 and 9. The reason for considering odd values only was to prevent equal weight values for both the classes as discussed earlier. KNN showed better results than SVM and maintained a higher average accuracy throughout all the split ratios.

It was observed that all feature selection methods and classifiers performed best for a 70:30 split ratio and only that split ratio was considered for analysis. The experiment was carried on Ubuntu 16.04 Operating System. We used 'R' Language (RStudio Version 3.4.1) for implementation. For text mining tasks we used 'tm' package and for analysis tasks we used 'RTextTools' package.

The experimental setup was used for multiple combinations of split ratios and feature selection methods. The results are presented in the below Table I-Table III. Tables [1-3] show the results of various classifiers with different feature selection methods.

TABLE I. CLASSIFICATION RESULTS USING SVM CLASSIFIER

Feature Selection Method	Number Of Features	Accuracy	Precision	Recall	F-Measure
Chi Squared	2000	**50.000**	**0.250**	**0.500**	**0.333**
	5000	49.933	0.250	0.499	**0.333**
	8000	50.000	0.250	0.500	**0.333**
Information Gain	2000	**50.000**	**0.250**	**0.500**	**0.333**
	5000	50.000	0.250	0.500	**0.333**
	8000	49.667	0.249	0.497	**0.332**
Correlation	2000	**57.532**	**0.375**	**0.576**	**0.454**
	5000	57.260	0.373	0.573	**0.452**
	8000	50.000	0.300	0.450	**0.360**
RLPI	50	70.593	0.733	0.705	**0.719**
	100	67.727	0.694	0.677	**0.685**
	150	**71.653**	**0.737**	**0.716**	**0.726**

TABLE II. CLASSIFICATION RESULTS USING KNN CLASSIFIER

Feature Selection Method	Number Of Features	Accuracy	Precision	Recall	F-Measure
Chi Squared	2000	**58.067**	**0.647**	**0.401**	**0.452**
	5000	53.333	0.647	0.425	0.368
	8000	53.333	0.575	0.363	0.340
Information Gain	2000	**58.067**	**0.649**	**0.395**	**0.450**
	5000	53.267	0.646	0.428	0.368
	8000	53.467	0.576	0.368	0.344
Correlation	2000	**57.133**	**0.755**	**0.492**	**0.426**
	5000	56.400	0.788	0.224	0.287
	8000	55.933	0.707	0.383	0.366
RLPI	50	**58.000**	**0.650**	**0.273**	**0.367**
	100	54.067	0.602	0.200	0.285
	150	55.867	0.629	0.284	0.351

TABLE III. CLASSIFICATION RESULTS USING NAÏVE BAYES CLASSIFIER

Feature Selection Method	Number Of Features	Accuracy	Precision	Recall	F-Measure
Chi Squared	2000	**53.133**	**0.534**	**0.589**	**0.552**
	5000	54.667	0.541	0.663	0.591
	8000	54.733	0.538	0.711	0.608
Information Gain	2000	**53.133**	**0.534**	**0.589**	**0.552**
	5000	54.667	0.541	0.663	0.591
	8000	54.733	0.538	0.711	0.608
Correlation	2000	**60.733**	**0.659**	**0.567**	**0.574**
	5000	56.667	0.567	0.632	0.584
	8000	56.133	0.568	0.655	0.582
RLPI	50	**63.400**	**0.680**	**0.509**	**0.568**
	100	58.467	0.591	0.593	0.584
	150	57.400	0.571	0.624	0.592

It can be seen that all the accuracies see an improvement when RLPI is used as feature selection method. SVM shows the best accuracy of 71.65% using RLPI as a feature selection algorithm. Since, we have used 5-fold validation technique; the above-mentioned results are the average of results of individual partition. Results may increase or decrease depending on the occurrence of words in that partition.

V. CONCLUSION

In this paper, RLPI when used as a feature selection method produces discriminating features which reduces the complexity of data representation by reducing the total number of features. In addition, RLPI shows better results for all three classifiers- SVM, KNN and Naïve Bayes when compared with traditional feature selection methods like- Information Gain, Correlation and Chi Square. Hence RLPI performs extremely well as a dimensionality reduction technique with best classification accuracy.

REFERENCES

[1] Y. Li and Y. Shiu, "A diffusion mechanism for social advertising over microblogs," *Decision Support Systems*, vol. 54, no. 1, pp. 9-22, December 2012.

[2] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. International Conference on Language Resources and Evaluation*, Valletta, Malta, 17-23 May 2010.

[3] K. K. Tseng, R. Y. Lin, and H. Zhou, "Price prediction of e-commerce products through Internet sentiment analysis," *Electron Commer Res.*, 2017.

[4] O. Almatrafi, S. Parack, and B. Chavan, "Application of location-based sentiment analysis using twitter for identifying trends towards Indian general elections 2014," in *Proc. 9th International Conference on Ubiquitous Information Management and Communication*, article no. 41, 2015.

[5] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications*, vol. 57, pp. 117-126, 2016.

[6] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14-46, November 2015.

[7] H. Zhang, W. Gan, and B. Jiang, "Machine learning and lexicon based methods for sentiment classification: A survey," in *Proc. 11th Web Information System and Application Conference*, 2014.

[8] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *Proc. Science and Information Conference*, 2014.

[9] Zareapoor, Masoumeh, and K. R. Seeja, "Feature extraction or feature selection for text classification: A case study on phishing email detection," *International Journal of Information Engineering and Electronic Business*, vol. 7, no. 2, pp. 60-65, Mar. 2015.
[10] L. Jin, W. Gong, W. Fu, and H. Wu, "A text classifier of English movie reviews based on information gain," in *Proc. 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, 2015.
[11] R. Tiwari and M. P. Singh, "Correlation-based attribute selection using genetic algorithm," *International Journal of Computer Applications*, vol. 4, no. 8, August 2010.
[12] S. Cang, "A mutual information based feature selection algorithm," in *Proc. 4th International Conference Biomedical Engineering and Informatics*, 2011.
[13] D. Cai, X. He, W. V. Zhang, and J. Han, "Regularized locality preserving indexing via spectral regression," in *Proc. Sixteenth ACM Conference on Conference on Information and Knowledge Management*, 2007, pp. 741–750.
[14] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proc. Conference on Empirical Methods in Natural Language Processing*, Philadelphia, July 2002, pp. 79-86.
[15] N. Chamansingh and P. Hosein, "Efficient sentiment classification of twitter feeds," in *Proc. IEEE International Conference on Knowledge Engineering and Applications*, 2016.
[16] M. Ghosh and G. Sanyal, "Preprocessing and feature selection approach for efficient sentiment analysis on product reviews," in *Advances in Intelligent Systems and Computing*, S. Satapathy, V. Bhateja, S. Udgata, and P. Pattnaik Eds., Springer, 2017, vol. 515.
[17] C. Pong-Inwong and K. Kaewmak, "Improved sentiment analysis for teaching evaluation using feature selection and voting ensemble learning integration," in *Proc. 2nd IEEE International Conference on Computer and Communications*, 2016.
[18] A. Yang, J. Zhang, L. Pan, and Y. Xiang. "Enhanced twitter sentiment analysis by using feature selection and combination," in *Proc. International Symposium on Security and Privacy in Social Networks and Big Data*, 2015.
[19] T. P. Sahu and S. Ahuja, "Sentiment analysis of movie reviews: A study on feature selection & classification algorithms," in *Proc. International Conference on Microelectronics, Computing and Communications*, 2016.
[20] S. K. Trivedi and A. Tripathi, "Sentiment analysis of Indian movie review with various feature selection techniques," in *Proc. IEEE International Conference on Advances in Computer Applications*, 2016.
[21] K. Gao, S. Su, and J. Wang, "A sentiment analysis hybrid approach for microblogging and e-commerce corpus," in *Proc. 7th International Conference on Modelling, Identification and Control*, Sousse, Tunisia, December 18-20, 2015.
[22] T. Parlar and S. Ayşe Özel, "A new feature selection method for sentiment analysis of Turkish reviews," in *Proc. International Symposium on Innovations in Intelligent Systems and Applications*, 2016.
[23] B. S. Harish, M. B. Revanasiddappa, and S. V. A. Kumar, "A modified support vector clustering method for document ategorization," in *Proc. IEEE International Conference on Knowledge Engineering and Applications*, 2016.
[24] M. S. Mubarok, Adiwijaya, and M. D. Aldhi, "Aspect-based sentiment analysis to review products using Na ̈ve Bayes," *AIP Conference Proceedings*, vol. 1867, id.020060, 2017.
[25] F. K. Ahmad, "Comparative analysis of feature extraction techniques for event detection from news channels' facebook page," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 1-2, pp. 2289-8131.
[26] L. Deng, Y. Hu, J. P. Y. Cheung, and K. D. K. Luk, "A data-driven decision support system for scoliosis prognosis," *IEEE Access*, vol. 5, 2017.
[27] S. Pathak and D. R. Rao, "Adaptive system for handling variety in big text," in *Lecture Notes in Networks and Systems*, Y. C. Hu, S. Tiwari, K. Mishra, and M. Trivedi, Eds., Springer, 2018, vol. 19.
[28] M. K. Raju, S. T. Subrahmanian, and T. Sivakumar, "A comparative survey on different text categorization techniques," *International Journal of Computer Science and Engineering Communications*, vol. 5, no. 3, pp. 1612-1618, 2017.
[29] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. (2011). Learning word vectors for sentiment analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*. [Online]. Available: http://ai.stanford.edu/~amaas/data/sentiment/

H K Darshan is pursuing B.E in Information Science and Engineering from Sri Jayachamarajendra college of Engineering. He is interested in Data Mining, Text Mining and Machine Learning.

Aditya R Shankar is pursuing B.E in Information Science and Engineering from Sri Jayachamarajendra college of Engineering. He is interested in Data Mining, Text Mining, Machine Learning and Cyber Security.

H M Keerthi Kumar received his B.E in Information Science and Engineering and M.Tech in Software Engineering from Visvesvaraya Technological University, India. He is currently pursuing Ph.D degree in Computer Science from University of Mysore, India. His area of research includes Data Mining, Pattern Recognition and Machine Learning.

B S Harish obtained his B.E in Electronics and Communication (2002), M.Tech in Networking and Internet Engineering (2004) from Visvesvaraya Technological University, Belagavi, Karnataka, India. He completed his Ph.D. in Computer Science (2011); thesis entitled "Classification of Large Text Data" from University of Mysore. He is presently working as an Associate Professor in the Department of Information Science & Engineering, JSS Science & Technology University, Mysuru. He was invited as a Visiting Researcher to DIBRIS - Department of Informatics, Bio Engineering, Robotics and System Engineering, University of Genova, Italy from May-July 2016. He delivered various technical talks in National and International Conferences. He has invited as a resource person to deliver various technical talks on Data Mining, Image Processing, Pattern Recognition, Soft Computing. He is also serving and served as a reviewer for National, International Conferences and Journals. He has published more than 50 International reputed peer reviewed journals and conferences proceedings. He successfully executed AICTE-RPS project which was sanctioned by AICTE, Government of India. He also served as a secretary, CSI Mysore chapter. He is a Member of IEEE (93068688), Life Member of CSI (09872), Life Member of Institute of Engineers and Life Member of ISTE. His area of interest includes Machine Learning, Text Mining and Computational Intelligence.