Comparative Analysis of Effect of Stopwords Removal on Sentiment Classification

Ms. Kranti Vithal Ghag[1]
Information Technology Department
MET's SAKEC, Mumbai University
Mumbai, India
krantiis@rediffmail.com

Dr. Ketan Shah[2]
Information Technology Department
SVKM's NMIMS MPSTME
Mumbai, India
ketanshah@nmims.edu

***Abstract*—Classification refers to the computational techniques for classifying whether the sentiments of text are positive or negative. Sentiment Classification being a specialized domain of text mining is expected to benefit after preprocessing such as removing stopwords. Stopwords are frequently occurring words that hardly carry any information and orientation. In this paper the effect of stopwords removal on various sentiment classification models was analyzed. Sentiment Classification models were evaluated using the movie document dataset. Accuracy increased from unprocessed dataset to stopwords removed dataset for Traditional Sentiment Classifiers. Our classifiers had hardly any impact of stopwords removal which indicates that they handled stopwords at the time of classification itself. Our classifiers also displayed accuracy better than traditional classifier and another surveyed classifier based on term weighting technique.**

Keywords—Sentiment Classification; Pre-processing; Stopwords Removal; Term Weighting; Term Frequency; Term Presence

I. INTRODUCTION

The web which is massively increasing resource of information has changed from read only to read write. Organizations now provide opportunity to the user to express their views on the products, decisions and news that are released [1]. Users can express their emotions as well can comment on the earlier user sentiments. Understanding consumer opinion for a product as well as for competitor's products is important for an organization to take crucial decisions. Large amount of sentiment data is generated by various users for products and services. Automatically processing this sentiment data needs to be handled systematically. Sentiment Classification involves preprocessing, extracting, understanding, classifying and presenting the emotions and opinions expressed by the users.

Data preprocessing is done to eliminate the incomplete, noisy and inconsistent data [2]. Preprocessing helps in maximizing classifier performance. Preprocessing for text classification involves tasks like tokenization, removing punctuation, removing special characters and stopwords. Sentiment Classification being a specialized domain of text mining is expected to benefit after preprocessing.

Sentiment Classification generally involves classifying the polarity of a piece of text [4] or classifying its subjectivity [3]. Sentiment Classification techniques construct sentiment model trained for domain specific tagged reviews [5]. It was also noted in our survey article that most of the research in Sentiment Analysis is focused on supervised learning techniques such as Naive-Bayes, Maximum-Entropy and Support Vector Machine (SVM) [6]. It was also marked that SVM was popularly used technique for Sentiment Classification. Supervised learning techniques entirely depend on the availability and the quality of tagged dataset. Generally a set of documents is used as a training set to the classifier. These documents are represented as vectors. Every term in the document is an element in the vector in SVM approach for text mining. Term Presence & Term Frequency are popular techniques used in Text Mining when representing documents as vectors [7]. In Term Presence technique an element can take a binary value. This element is set to one if the term is present in document otherwise set to zero. In Term Frequency technique an element in the document vector is the count of the given term in a document.

For Sentiment Classification the training dataset consists of reviews tagged positive and negative. Reviews tagged positive are called positively tagged documents and reviews tagged negative as negatively tagged documents. Every element in the vector represents a term that occurred in some document/s. Each element of vector has two counts associated with it. One count is number of times of occurrence of that term (element) in positively tagged documents and other is number of times of occurrences in negatively tagged documents.

A. *Contribution*

Our model is based on preprocessing the input text to improve Sentiment Classification. Effects of Removal of Stopwords on Sentiment Classification were experimented. The dataset with stopwords (original) was input to four classifiers and the stopwords were removed and the dataset was provided to same four classifiers. One of these classifiers was Traditional Sentiment Classifier and three were proposed by us in our previous research work [27] [28] [29]. These classifiers are based on term weighting functions and are adapted for sentiment classification [4] [8]. These methods are on combinations of frequency count and presence count distribution of term. Although our approach is based on traditional techniques of Text Mining, we examine whether addressing Sentiment Classification as special case of Text Mining can improve classification accuracy. Accordingly we have attempted to adapt the model for Sentiment Classification, considering the similarities and differences with Text Mining techniques. A term was classified as positive if its dominance in positively tagged documents was more than negatively tagged documents and vice versa. This can be calculated using document vectors. The i^{th} element of each vector that was constructed from positively tagged documents contributed to positivity of i^{th} term and similarly i^{th} element of each vector that was constructed from negatively tagged documents contributed to negativity of the same term.

Our approach differs significantly from traditional approaches on the basis of usage pattern of term presence and term count vectors. Our classifiers focus on proportional frequency count distribution and proportional presence count distribution whereas traditional approaches such as delta TFIDF and other term weighting techniques rely on combination of overall frequency count of term and proportional presence count distribution.

The rest of the paper is organized as follows. Sentiment Classification and preprocessing techniques are analyzed in section 2. Section 3 focuses on the Sentiment Classification models. Experimental setup is discussed in section 4. Results are presented in section 5. Concluding remarks and future scope are put forth in section 6.

II. Prior Work

Lin, Everson and Rugerpreprocessed reviews to extract words and noise such as punctuations, numbers, and non-alphabet characters were removed [9]. Stemming was applied so that the related terms fall in same clusters, thus reducing the vocabulary classes. MPQA and appraisal lexicons were merged stemmed and cleaned to form a new lexicon which was used to classify the document irrespective of the domain.

Haddi, liu and Shi observed enhanced classifier performance when preprocessing techniques such as White space removal, Stopwords removal, Negation handling and Stemming were applied [10]. They also applied feature selection using chi-square method for dimensionality reduction. R. Duwairi and M. El-Orfali also observed increase in classifier performance when preprocessing tasks such as Stemming and Feature correlation were applied [11]. Hemalatha, Varma and Govardhan applied preprocessing on data extracted from twitter to remove URLs, Special characters and Questions to enhance performance [2]. Emoticons and Acronyms dictionary was constructed for preprocessing [12].

Pang, Lee and Vaithyanathan laid the foundation of harnessing supervised machine learning techniques for Sentiment Classification. They are also the pioneers for extracting, transforming and making available the popular movie review dataset. Naive Bayes, maximum entropy classification, and support vector machines algorithms were applied on unigrams and bigrams features and their weights, extracted from this movie dataset [13]. They concluded that sentiment analysis problem needs to be handled in a more sophisticated way as compared to traditional text categorization techniques. SVM classifier applied on unigrams produced best results unlike information retrieval where bigrams generate remarkable accuracy as compared to unigrams.

Mullen and Collier used SVMs and expanded the feature set for representing documents with favorability measures from a variety of diverse sources [14]. They introduced features based on Osgood's Theory of Semantic Differentiation, using Word-Net to derive the values of potency, activity and evaluative of adjectives [15] and Turney's semantic orientation [16]. Their results showed that using a hybrid SVM classifier that uses as features the distance of documents from the separating hyper plane, with all the above features produces the best results.

Zaidan, Eisner, and Piatko introduced "annotator rationales", i.e. words or phrases that explain the polarity of the document according to human annotators [17]. By deleting rationale text spans from the original documents they created several contrast documents and constrained the SVM classifier to classify them less confidently than the originals. Prabowo and Thelwall [18] proposed a hybrid classification process by combining in sequence several ruled-based classifiers with a SVM classifier. The former were based on the General Inquirer lexicon by lin, Wilson, Wiebe and Hauptmann. [19] and the MontyLingua part-of-speech tagger by Liu [20] and co-occurrence statistics of words with a set of predefined reference words. Their experiments showed that combining multiple classifiers can result in better effectiveness than any individual classifier, especially when sufficient training data isn't available. Bruce and Wiebe made an effort to manually tag sentences as subjective or objective by different judges and the resultant confusion matrix was analyzed [21]. 14 articles were randomly chosen and every non-compound sentence was tagged. Also a tag was attached to conjunct of every compound sentence. Authors then attempted to identify if pattern exists in agreement or disagreement between human judges. Authors observed that manual tagging suffered due drawback of biased nature of human beings during tagging phase. Dave, Lawrence and Pennockused a self tagged corpus of sentiments [22] available on major websites such as Amazon and Cnet as training set. Naïve Bayes classifier was trained and refined using the above corpus. The classifier was then tested on other portion of self-tagged corpus. The sentences were parsed to check semantic correctness and then tokenized. Pre-processing techniques such as co- allocation substrings and stemming were applied for generalisation of tokens. Pre-processed, N-grams (bi-gram and tri-gram) improved the results as compared to unigram. Score were then assigned to features.

Zhang constructed computational model that explored reviews linguistics properties to judge its usefulness [23]. Support Vector Regression (SVR) algorithm was used for classification. In contrast to major studies which filter out subjective information in any review or are not considered important, Zhang claimed that the quality of review was reasonably good if it was a good combination of subjective and objective information. Yu, Liu and Huang attempted to identify hidden sentiment factors in the reviews [24]. Bag of words approach was used for sentiment identification in the review. Along with sentiment identification, product sales prediction methods were also proposed. TFIDF is a popular statistical technique to index the term as per their importance. TFIDF is based on documents and term vectors that represent term frequency & presence [25] [26].

$$d^{(i)} = TF(w_i, d).IDF(w_i) \tag{1}$$

Where,

$d^{(i)}$ = TFIDF of term w_i in document d.

TF (w_i,d) = Term Frequency of term w_i in document d.

w_i = i^{th} term.

d = document.

IDF(w_i) = Inverse Document Frequency of term w_i.

TFIDF of term w_i in document d can be computed using "(1)". Term frequency TF(w_i,d) is count of a term w_i in document d. Larger value of a Term Frequency indicates its prominence in a given document. Terms present in too many documents were suppressed as these tend to be stop words. This suppression was handled by the second component IDF.

$$IDF(w_i) = \log [\ |D| / DF(w_i)\ ] \tag{2}$$

Where,

IDF(w_i) = Inverse Document Frequency of term w_i.

w_i = i^{th} term.

$|D|$ = the total count of documents.

DF (w_i) = count of documents that contain term w_i.

If a term is present in all the documents then numerator equals denominator in "(2)". As a result of this IDF (w_i) = log 1 which is zero. But if term occurred in relatively less number of document then DF $(w_i) < |D|$. As a result IDF (w_i) = log (>1) which is a positive integer. Term presence vector was used for calculation of IDF.

TFIDF 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 [4]. Unlike TFIDF which used single term presence vector, Martineau and Finin constructed two vectors for presence in positively tagged documents and negatively tagged documents.

In connection with the occurrences of rare words, different variations of TFIDF scores of words, indicating the difference in occurrences of words in different classes (positive or negative reviews), have been suggested by Paltoglou and Thelwall [8]. They surveyed many term weighting techniques as well proposed "smart" and "BM25" term weighting techniques for sentiment classification.

III. Sentiment Classification Models for Analysing the Effect of Stopword Removal.

A. *Sentiment Classification Algorithm*

Input**:**

Dp = {Dp1,Dp2,…DpM} Set of positively tagged Documents

Dn = {Dn1,Dn2,…DnM} Set of negatively tagged Documents

Output**:**

Tpos = {T1, T2,… TA} Terms classified as positive.

Tneg = {T1, T2,… TB} Terms classified as negative.

Tneu = {T1, T2,… TC} Terms classified as neutral.

B. *Algorithm*:

- Construct unique termlist T = {T1, T2,…Tt}

  Such that Ti € {Dp U Dn}
- Tpos = Tneg = Tneu = ϕ
- For every term Ti in T

  If dominancy of Ti in Dp > Dn

  Tpos = Tpos U Ti

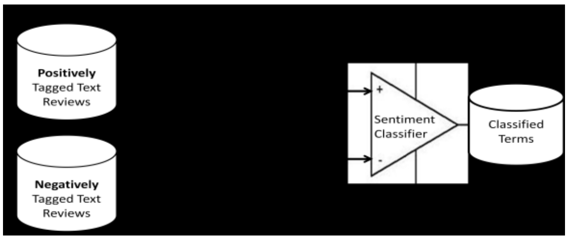
  If dominancy of Ti in Dn > Dp

  Tneg = Tneg U Ti

  Else

  Tneu = Tneu U Ti

Input dataset to the algorithm was initially the original unprocessed dataset. If the number of unique terms generated is too large then term document matrix tends to be very large and sparse. If this matrix could be optimally reduced, then it would become dense and contribute to improvement in accuracy. Sentiment Classification was then performed after removing Stopwords from the unprocessed dataset. Classification task is expected to benefits due to Stopwords removal as it would contribute to Dimensionality Reduction.



C. *Sentiment Classification Model before Stopwords Removal*

Fig. 1. Sentiment Classification Model with Unprocessed Dataset.

Figure 1 represents a model for Sentiment Classification where the unprocessed balanced dataset was input to the classifier. Dataset is a set of text review files, tagged positive or negative. This dataset was in the format as written by user. Sentiment Classification task was accomplished using various Sentiment Classifiers described later in this section.

D. Sentiment Classification Model after Stopwords Removal

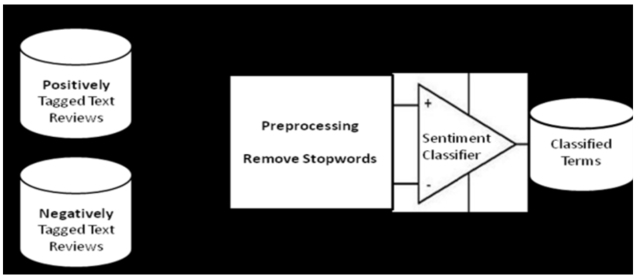


Fig. 2. Sentiment Classification Model with Stopwords removed Dataset.

Figure 2 represents a Sentiment Classification model with a Preprocessing Techniques to Remove Stopwords. Stopwords such as “the”, “and”, “is’ were removed. Natural Language Toolkit (NLTK) English Corpus Stopwords list was used.

Sentiment Classification was accomplished with:

- Dataset 1 – Unprocessed Dataset.
- Dataset 2 – Dataset after Stopwords Removed.

In dataset 2 an attempt was made to reduce the number of unique terms generated in the Term-Document matrix to incorporate Dimensionality Reduction. The resultant Term-Document Matrix was denser. These Term-Document matrices constructed from different datasets were separately inputted to different Sentiment Classifiers.

E. Sentiment Classifier Models

The unprocessed and preprocessed dataset were provided as input to different Sentiment Classifier. The terms were classified into three sentiment classes i.e. positive, negative and neutral. A term was classified as positive if it was dominant in positively tagged reviews, negative if dominant in negatively tagged reviewsotherwise classified as neutral. Dominancy of a term in reviews was determined by four Sentiment Classifiers Models, Traditional Sentiment Classifier (TSC) [3], Average Relative Term Frequency Sentiment Classifier (ARTFSC) [27], Senti-Term Frequency Inverse Document Frequency (Senti-TFIDF) [28] and Relative Term Frequency Sentiment Classifier (RTFSC) [29]. The later three were proposed by us in our previous research articles. The classifier models varied in ways of determining the dominancy of terms in positively tagged and negatively tagged reviews.

In Traditional Sentiment Classifier (TSC), polarity of term was computed based on its frequency count distribution across positively tagged documents and negatively tagged documents. A term was classified as positive if it was present more number of times in positively tagged documents as compared to negatively tagged documents and vice-versa. If a term count varied slightly term was classified as positive or negative. For example if term count in positively tagged documents was = 9 and count of same term in negatively tagged documents was = 8, the term was classified as positive.

Average Relative Term Frequency Sentiment Classifier (ARTFSC) was based on the term frequency count and count of term. It is actually based on average frequency count of term, to presence count of term. A term was classified as positive if its average frequency in positively tagged documents was larger than its average frequency in negatively tagged documents and vice-versa. If average frequency count of a term in positively tagged documents was equal to its average frequency count in negatively tagged document, then the term would be classified as Senti-stop-word. But if average counts varied slightly, the term would be classified as positive or negative. To avoid this biased classification, a window was provided for handling neutral words. If average frequency count of a term in positively tagged documents was equal to or nearly equal to average frequency count of a term in negatively tagged documents, the term was classified as neutral. Optimal values for window boundaries were experimentally determined to maximize accuracy. A term was classified based on its relative average frequency count in positively and negatively tagged documents.

Sentiment Term Frequency Inverse Document Frequency (Senti-TFIDF) worked on the principle of logarithmic proportion of Term Frequency Inverse Document Frequency (TFIDF) of a term across positively tagged documents and negatively tagged documents. If the TFIDF of a term in positively tagged documents was larger than TFIDF of same term in negatively tagged documents the term is assigned positive polarity and vice-versa. TFIDF and thus Senti-TFIDF is based on the term frequency count as well as term presence count of term in dataset. Similar to ARTFSC window was defined for handling Senti-stop-words.

Relative Term Frequency Sentiment Classifier (RTFSC) worked on the principle of logarithmic proportion of Term Frequency of a term across positively tagged documents and negatively tagged documents. If the term frequency of a term in positively tagged documents was larger than term frequency of same term in negatively tagged documents then the term was assigned positive polarity and vice-versa. RTFSC is purely based on the term frequency count of term in dataset. Similar to ARTFSC and Senti-TFIDF window was defined for handling Senti-stop-words.

IV. Experiments Conducted

Pang and Lee’s Movie Document Dataset was used in experiments. Movie document dataset contains 1000 positively tagged text documents and 1000 negatively tagged text documents. Each text document is a review of a user. These review text files size varied from 1 to 15kb. Words per document varied from 17 to 2678.

Initially the experiments were performed on unprocessed reviews i.e. above mentioned dataset. Then same experiments were performed on datasets where Stopwords were removed. A list of terms that occurred in the reviews was prepared. A term is entered only once in this term list although it may appear may times in reviews. A vector was constructed for every review. Every i^{th} element in this vector was count of i^{th} term in this review. If a term in term list was not present in the reviews the count associated with that term was set to zero. These vectors were used to calculate term polarity for the terms in the term list.

Polarity was calculated using Traditional Sentiment Classifier (TSC), Average Relative Term Frequency

Sentiment Classifier (ARTFSC), Sentiment Term Frequency Inverse Document Frequency (Senti-TFIDF) and Relative Term Frequency Sentiment Classifier (RTFSC) models described in section 3. A term was classified either as positive or negative or neutral.A review was classified positive by our model if total number of positive terms in the reviews were more than negative terms otherwise negative.

If a review was tagged as positive and also classified as positive then it contributed to True Positive in confusion matrix. Similarly tagged as negative and classified as negative, contributed to True Negative. If a reviews was tagged as positive but classified as negative then it contributed to False Negative. Similarly tagged as negative but classified as positive, contributed to False Positive. Below mentioned 2 experiments were performed on 08 Sentiment Classification models. Each of these models had one of the four Sentiment Classifiers (PSC, ARTFSC, Senti-TFIDF & RTFSC) applied on unprocessed and Stopwords removed dataset.

A. Experiment 1

Experiment 1 was conducted to determine if a term should be classified as Senti-stop-word. If dominance of a term in positively tagged documents and negatively tagged documents was exactly equal or varied slightly then the term was tagged as Senti-stop-word. A window was defined to tag a term as Senti-stop-word. The boundary values of this window were experimentally determined to maximize accuracy. For this accuracy was computed using, 10 Fold Cross Validation (10 fold CV). The window boundary range was varied between 0 to 5 at step of 0.5 and simultaneously 0 to -5 at step of -0.5.To calculate accuracy dataset was divided in 10 parts. At every fold each 10% dataset was used for testing and remaining 90% dataset was used for training the classifier. Confusion matrix & accuracy was calculated at every fold and then averaged to form the accuracy of the model.

B. Experiment 2

10 Fold Cross Validation (10 fold CV) technique [26] was used to calculate accuracy. Optimal window boundary values, determined in experiment 1 were set. Confusion matrix was constructed as well as accuracy was calculated at every fold and averaged to form the accuracy of the model.

RESULTS AND DISCUSSION

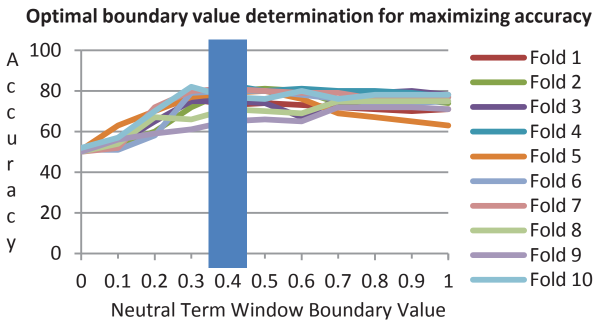


Fig. 3. Optimal boundary value determination for SentiTFIDF classifier on stopwords removed dataset for maximizing accuracy.

Figure 3 represents accuracy graph for determining optimal window boundary value for SentiTFIDF classification model. Stopwords removed dataset was provided to classifier. Accuracy was computed using 10 Fold CV and by varying window boundary value. Accuracy of each fold is represented in different color. Average accuracy at all folds for SentiTFIDF classifier on stopwords removed dataset sentiment classification model was largest at 0.4 window boundary value. Similarly optimal window boundary values for all Sentiment Classification models were experimentally determined as mentioned in experiment 1 to maximize accuracy.More words were classified as neutral if window boundary value was larger resulting to lesser number of opinionated words. Conversely if window boundary value was set to a smaller value stopwords would not be appropriately identified. So an optimal value of window boundarywas determined for each model. The window boundary value was varied over a range for each model as mentioned in experiment 1 and the value that yielded largest accuracy was set as the optimal window boundary values for that specific model. Window boundary values was not applicable (or set to zero) for Traditional Sentiment Classifier (TSC) as it is not based on relative or ratio based mathematical model.

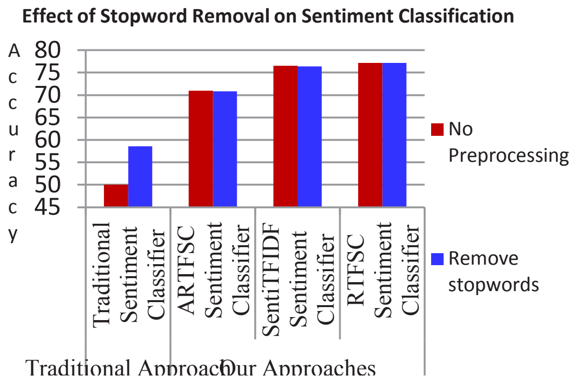


Fig. 4. Accuracy Graph for Movie Document Dataset.

Figure 4 represents accuracy of different classifier on unprocessed dataset and stopwords removed dataset. Accuracy increased from unprocessed dataset to stopwords removed dataset only for Traditional Sentiment Classification Algorithms. Our Sentiment Classification Models ARTFSC, Senti-TFIDF and RTFSC showed nearly equal performance for unprocessed dataset that is the dataset with stopwords as well as for stopwords removed dataset. The mathematical model of our algorithms was designed to handle stopwords.
Overall accuracy of our models is also better than Traditional Sentiment Classification Model. Of the three models proposed, Relative Term Frequency Sentiment Classifier (RTFSC) has highest accuracy. Sentiment Term Frequency Inverse Document Frequency (Senti-TFIDF) ranked second. Average Relative Term Frequency Sentiment Classifier (ARTFSC) positioned third.

V. CONCLUSION AND FUTURE WORK

Traditional Sentiment Classifier showed an improvement in accuracy from 50% to 58.6% when stopwords were removed, whereas our approaches ARTFSC, SentiTFIDF and RTFSC performed nearly same when stopwords were removed. This indicates that our approaches efficiently handle stopwords at the time of classification using the NTWB neutral term window boundary value. A separate preprocessing task of stopwords removal is not needed for our algorithms.

Accuracy of ARTFSC, SentiTFIDF and RTFSC was 71%, 76.5% and 77.2% respectively which is much better than Traditional Sentiment Classifier for which the maximum noted accuracy was 58.6. Accuracy of our models is also better than a comparable term weighting sentiment classification model, Delta-TFIDF with accuracy equal to 67.9%.

ARTFSC, SentiTFIDF and RTFSC also showcased accuracy better than a comparable term weighting technique Delta-TFIDF with accuracy equal to 67.9%

Out of our three models, RTFSC has highest accuracy of 77.2%. Then Senti-TFIDF ranked second with accuracy 76.4% and ARTFSC positioned third with accuracy of 70.8%.

Although the accuracies of surveyed techniques cannot be directly compared as the experimental parameters may vary, Relative Term Frequency Sentiment Classifier (RTFSC) performs better than most existing techniques.

Our sentiment classifiers are based on term frequency and presence distribution. In future we aim to incorporate concept adaptability to Sentiment Classification Models.

References

[1] H. Chen, "Business and market intelligence 2.0,"In IEEE Intelligent Systems, vol. 25, issue no. 01, pp. 68–71, 2010.

[2] I. Hemalatha, Dr. G Varma, and Dr. A Govardhan, "Preprocessing the informal text for efficient sentiment analysis," In International Journal of Energing Trends & Technology in Computer Science, vol. 01, issue no. 02, pp. 58-61, 2012.

[3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and Trends in Information Retrieval, vol. 02, issue no. 01–02, pp. 1–135, 2008.

[4] J. Martineau and T. Finin, "Delta TFIDF: An improved feature space for sentiment analysis," In Proc. of 3rd Int'l AAAI Conf. on Weblogs and Social Media, pp. 258-261, 2009.

[5] R. Xia and C. Zong, "A POS-based ensemble model for cross-domain sentiment classification," In Proc. of 5th Int'l Joint Conf. on Natural Language Processing, pp. 614–622, 2011.

[6] K. Ghag and K. Shah, "Comparative analysis of the techniques for sentiment analysis," In Proc. of Int'l Conf. on Advances in Technology and Engineering, pp. 1-7, 2013.

[7] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," Intelligent Systems, IEEE , vol. 28, issue no. 02, pp. 15-21, 2013.

[8] G. Paltoglou and M. Thelwall, "A study of information retrieval weighting schemes for sentiment analysis," In Proc. of 48th Annual Meeting of the Association for Computational Linguistics, pp. 1386-1395, 2010.

[9] Lin, Y. He, R. Everson, and S. Ruger, "Weakly supervised Joint-Sentiment-Topic detection from Text," In Knowledge and Data Engineering, IEEE Transactions, vol. 24, issue no. 06, pp. 1134-1145, 2012.

[10] E Haddi, X. liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," In Proc. of 1st Int'l Conf. on Information Technology and Quantitative Management, pp. 26-32, 2013.

[11] R. Duwairi and M. El-Orfali, "A study of the effects of preprocessing strategies on sentiment analysis for Arabic text," In International Journal of Information Science, vol. 40, issue no. 04, pp. 501-513, 2014.

[12] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of Twitter data," In Proc.of the Workshop on Language in Social Media, pp. 30-38, 2011.

[13] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," In Proc. of Conf. on Empirical Methods in Natural Language Processing, pp. 79-86, 2002.

[14] T. Mullen and N. Collier, "Sentiment Analysis using Support Vector Machines with diverse information sources," In Proc. of Conf. on Empirical Methods in Natural Language Processing, pp. 412–418, 2004.

[15] C. Osgood, George J. Suci, and Percy H. Tannenbaum, "The measurement of meaning," University of Illinois Press Urbana, 2nd edition, 1967.

[16] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," In Proc. of the 40th Annual Meeting on Association for Computational Linguistics ACL, pp. 417–424, 2002.

[17] O.F. Zaidan, J. Eisner, and C.D. Piatko, "Using annotator rationales to improve machine learning for text categorization," In Proc. of Conf. of North American Chapter of the Association for Computational Linguistics, pp. 260–267, 2007.

[18] R. Prabowo and Mike Thelwall, "Sentiment analysis: A combined approach," In Journal of Informetrics, vol. 03, issue no. 02, pp. 143–157, 2009.

[19] W. Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann, "Which side are you on? identifying perspectives at the document and sentence levels," In Proc. of the Conf. on Natural Language Learning, pp 109-116, 2006.

[20] H. Liu., "MontyLingua: An end-to-end natural language processor with common sense". Technical report,MIT, 2004.

[21] R. Bruce and J. M. Wiebe, "Recognizing subjectivity: A case study in manual tagging," In Natural Language Engineering, ACM, vol. 05, issue no. 02, pp. 187-205, 1999.

[22] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," In Proc. of 12th Int'l Conf. on World Wide Web, pp. 519 - 528, 2003.

[23] Z. Zhang, "Weighing Stars: Aggregating online product reviews for intelligent e-commerce applications," In Intelligent Systems, IEEE, vol. 23, issue no. 05, pp. 42-49, 2008.

[24] X. Yu, Y. Liu, X. Huang, and A. An, "Mining online reviews for predicting sales performance: A case study in the movie domain," In Knowledge and Data Engineering, IEEE Transactions, vol.24, issue no. 04, pp. 720-734, 2012.

[25] G. Salton and M. McGill, "Introduction to modern information retrieval," McGraw-Hill, pp. 105-107 & 205, 1983.

[26] J. Han and M. Kamber, "Data mining concepts and techniques," Morgan Kaufmann Publishers, 02nd edition, pp. 364-365, 2006.

[27] K. Ghag and K. Shah, "ARTFSC – Average relative term frequency sentiment classification," In International Journal of Computers and Technology, (IJCT), vol. 12 issue no. 06, pp. 3591-3601, 2014.

[28] K. Ghag and K. Shah, "SentiTFIDF – Sentiment classification using relative Term Frequency Inverse Document Frequency," In International Journal of Advanced Computer Science and Applications, vol. 05 issue no. 02, pp.36-43, 2014.

[29] K. Ghag and K. Shah, "RTFSC – Relative term frequency sentiment classification," In Proc. of Int'l Conf. on. Recent Trends in Computer and Electronic Engineering, pp. 52-55, 2014.