IJITKM 0973-4414 Volume-8, Number-2 Jan-Jun 2015 pp. 61-70 available online at www.csjournalss.com

# Optimising Sentiment Classification using Preprocessing Techniques

Ms. Kranti Vithal Ghag [1], Dr. Ketan Shah [2]
Information Technology Department, MET's SAKEC, Mumbai, India [1]
Information Technology Department, SVKM's NMIMS MPSTME, Mumbai, India [2]
krantiis@rediffmail.com [1], ketanshah@nmims.edu [2]

***Abstract***—Sentiment 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. In this paper we propose various models with selective combinations of preprocessing techniques and Sentiment Classifiers, to optimize Sentiment Classification. Unlike traditional preprocessing technique where punctuation symbols are discarded, we proposed a set of rules to handle words with apostrophe and then remove punctuation symbols. Sentiment Classifiers that were proposed in our previous research articles are based on term weighting techniques. We evaluated Sentiment Classification models by comparing them with state of art techniques using the movie sentence and movie document dataset. Accuracy increased from unprocessed dataset to preprocessed data. Our Classifiers handled stopwords thus had hardly any impact of stopwords removal in preprocessing unlike traditional Sentiment Classifiers. Our classifiers also displayed accuracy better than traditional classifier and another surveyed classifier based on term weighting technique.

***Keywords— Sentiment Classification; Pre-processing; Term Weighting; Term Frequency; Term Presence; Document Vectors***

## 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 removing 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 or classifying its subjectivity [3]. Polarity of a term, sentence, paragraph or document is classified as positive or negative [4].

Sentiment Classification techniques construct sentiment model trained with the help of tagged reviews. As these reviews are collection of domain-wise tagged set, the model constructed served well for specific domains [5].

It was 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 training set to the classifier for Text Mining. 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 and Term Frequency are two popular techniques used for 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 if the term is not present in document. In Term Frequency technique an element in the document vector is a non-negative integer that is set to count of the given term in a document.

For Sentiment Classification the training dataset consists of reviews tagged as positive and negative. All reviews tagged positive are called positively tagged documents whereas all reviews tagged negative are called negatively tagged documents. Every element in the vector represents a term that occurred in some document/s of training set. 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 preprocessing steps such tokenization, removing of punctuations and removal of stopwords on Sentiment Classification were experimented. Unlike traditional preprocessing technique were punctuation symbols are discarded, we proposed a set of rules to handle words with apostrophe to be mapped to correct words.

These preprocessed dataset were inputted to five different classifiers. Three of these were proposed by us in our previous research work. These classifiers are based on traditional term weighting functions where the vectors are processed to identify and sequence index terms. Some of these are techniques 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 and *handling of words with apostrophe* at preprocessing step. 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 its preprocessing techniques are surveyed in section 2. Section 3 focuses on the Sentiment Classification models for Sentiment Classification. 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 Ruger preprocessed 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]. Other than traditional preprocessing techniques Agrawal et *al.* constructed Emoticons and Acronyms dictionary 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. Using the largest training set size, their approach significantly increased the accuracy on movie review data set.

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 Pennock used 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. Preprocessing techniques such as co-allocation substrings and stemming were applied for generalisation of tokens. When pre-processed, N-grams (bi-gram and tri-gram) improved the results as compared to unigram. They also applied smoothing so that non-zero frequencies were available. 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.

Hybrid approaches having combination resources such sentiment lexicons and classifier can be harnessed for difficult tasks such as news article sentiment analysis.

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 as well as term presence [25] [26]. Term presence could be constructed if term frequency vector is available but vice-versa is not possible.

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

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.

$$IDF(w_i) = \log\left(\frac{|D|}{DF(w_i)}\right) \quad (2)$$

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]. Martineau and Finin constructed vectors to classify a term based on term frequency vector as well as term presence vectors. Unlike TFIDF which used single term presence vector, two vectors were separately constructed 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. Optimising Preprocessing

Our model works on the principle of optimized dimensionality reduction. If the number of unique terms generated is too large the term document matrix tends to be very large and a sparse matrix. If this matrix could be optimally reduced, then it becomes dense and contributes to improvement in accuracy. Our model applies various preprocessing techniques such as handling words with apostrophe and removing punctuations and stopwords for dimensionality reduction.

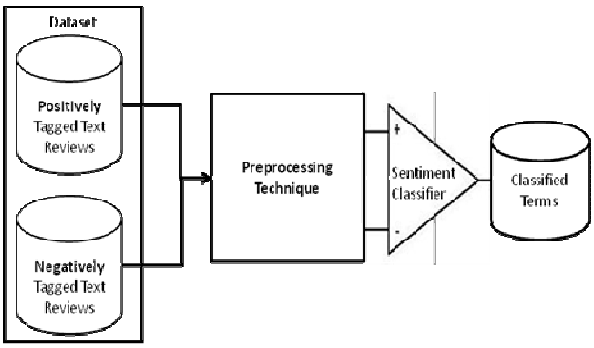



Figure 1 - Generalized Sentiment Classification Model (SCM)

Figure 1 represents a generalized model for Sentiment Classification. It can be divided into three components. The first component is input dataset, the second component is the preprocessing technique and the third is Sentiment Classifier. We proposed various models with selective combinations of preprocessing techniques and Sentiment Classifier to optimize Sentiment Classification.

### A. *Input Dataset*

The dataset inputted to the model is a balanced set of text reviews that are tagged as positive or negative. We

experimented with a movie document dataset. A document dataset is set of text files tagged positive or negative where each file represents sentiments of users.

*B. Preprocessing Technique*

Processing techniques help in dimensionality reduction. As text data are large in size, text classification is expected to benefit due to preprocessing. Preprocessing techniques like handling words with apostrophe, removing punctuations and removing stopwords were applied in selective combinations. The combinations described below were experimentally determined to maximize accuracy.

*1) Dataset 1: No Preprocessing*

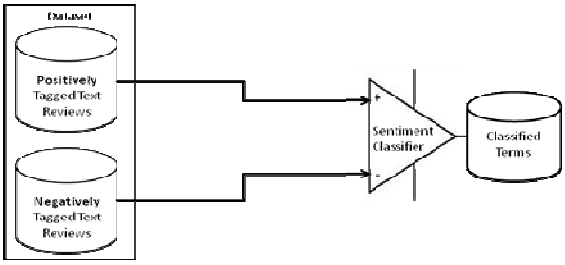



Figure 2 - Sentiment Classification Model (SCM) with Unprocessed Dataset

Initially the dataset was inputted to the classifier without any preprocessing. This dataset was in the format as written by user, without any form of cleaning.

*2) Dataset 3 : Handling Words with Apostrophe and removing Punctuations*

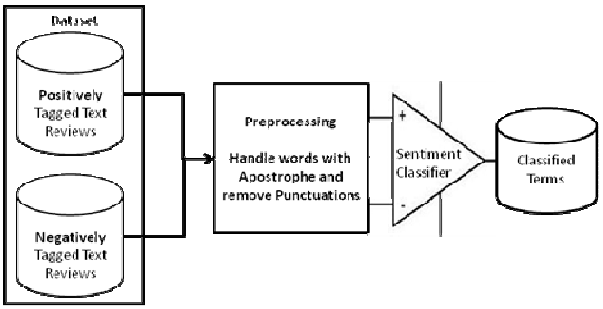



Figure 3 - Sentiment Classification Model with Preprocessing Technique to handle words with an Apostrophe & remove Punctuations

Figure 3 represents a Sentiment Classification model with a Preprocessing Techniques to handle words with Apostrophe as well as remove Punctuation Symbols.

The words with apostrophe such as “isn’t”, “that’s” and “I’m” were an overhead in term-document matrix. Consider a term-document matrix entries for the word “isn’t” in table 1.

Table 1: Example of Term-Document Matrix entries for word “isn’t” before handling apostrophe

| | Terms | | |
|---|---|---|---|
| **Documents** | **Is** | **Not** | **isn’t** |
| **Document 1** | 1 | 3 | - |
| **Document 2** | - | 1 | 1 |
| **Document 3** | 1 | 1 | - |

The term “isn’t” was then replaced with the words “is not”. The resultant term document matrix is represented in table 2

Table 2: Example of Term-Document Matrix entries for word “isn’t” after handling apostrophe

| | Terms | |
|---|---|---|
| **Documents** | **is** | **not** |
| **Document 1** | 1 | 3 |
| **Document 2** | 1 | 2 |
| **Document 3** | 1 | 1 |

It can be observed in table 2 that a dimension in matrix is reduced and the matrix is denser than earlier. Suffices were replaced to handle apostrophe words. These rules are tabulated below.

Table 3: Set of rules for Handling Apostrophe

| No | Rule | Example |
|---|---|---|
| 1 | n’t → _ not | wasn’t → was not |
| 2 | ’s → _ is | that’s → that is |
| 3 | ’re → _ are | you’re → you are |
| 4 | ’ve → _ have | they’ve → they have |
| 5 | ’m → _ am | I’m → I am |
| 6 | ’d → _would | they’d → they would |
| 7 | ’ll → _ will | you’ll → you will |
| 8 | ’em → _them | make’em → make them |
| 9 | in’ → _ ing | fringgin’ → frigging |
| Note: Symbol “_” indicates space | | |

The sequence of rules, listed in table 3 is important.

Along with handling words with apostrophe other punctuation symbols like “!”, “%” and “#” were also removed. Similar to handling apostrophe, removing punctuation also helped in dimensionality reduction. For example terms “Alas” and “Alas!” were different dimensions before the punctuations were handled.

*3) Handling Words with Apostrophe, removing Punctuations and removing Stopwords*

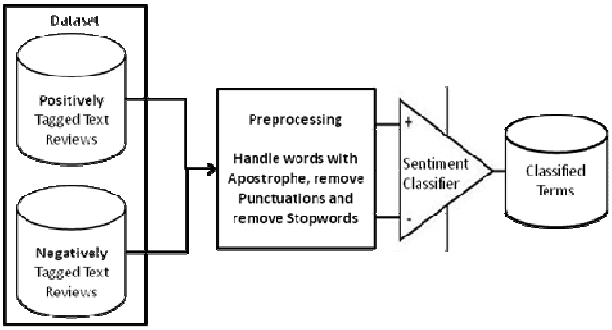



Figure 4 - Sentiment Classification Model with Preprocessing Technique to handle words with an Apostrophe, remove Punctuations & remove Stopwords

Figure 4 represents a Sentiment Classification model with a Preprocessing Techniques to handle words with Apostrophe, remove Punctuations and remove Stopwords. Along with handling words with apostrophe and removing punctuation symbols, stopwords were also removed. Stopwords such as

"the", "and", "is' were removed. Natural Language Toolkit (NLTK) English Corpus Stopwords list was used. This helped further in dimensionality reduction.
Sentiment Classification was accomplished with:

1. Dataset 1 – Unprocessed Dataset.
2. Dataset 2 – Dataset after handling words with apostrophe and removing punctuations.
3. Dataset 3 – Dataset where Stopwords were also removed along with handling apostrophe words and removing punctuation symbols.

At every new dataset 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 as the degree of preprocessing was increased. These Term-Document matrices constructed from different datasets were separately inputted to different Sentiment Classifiers.

### *C. Sentiment Classifier Models*

The preprocessed dataset were provided as input to different Sentiment Classifier. The terms were classified into 3 sentiment classes i.e. positive, negative and neutral.
A term was classified as positive if it was dominant in positively tagged reviews or as negative if it was dominant in negatively tagged reviews, otherwise classified as neutral. Dominancy of a term in reviews was determined by 5 Sentiment Classifiers. Traditional Sentiment Classifier (TSC) [3] & Delta-Term Frequency Inverse Document Frequency (Delta-TFIDF) [4], Average Relative Term Frequency Sentiment Classifier (ARTFSC) [27], Senti-Term Frequency Inverse Document Frequency (Senti-TFIDF) [28] & Relative Term Frequency Sentiment Classifier (RTFSC) [29] were used for Sentiment Classification. The later 3 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.

#### *1) Traditional Sentiment Classifier (TSC) [3]*

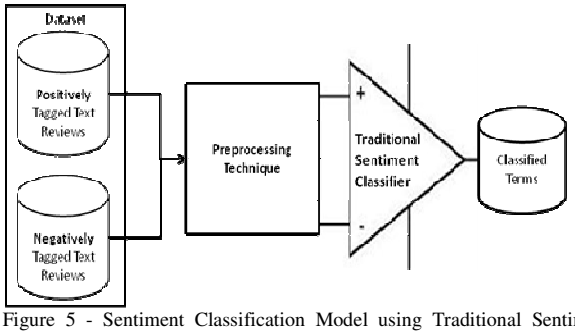



Figure 5 - Sentiment Classification Model using Traditional Sentiment Classifier

Figure 5 represents Sentiment Classification Model using Traditional Sentiment Classifier. Traditional Sentiment Classifier was based on frequency of term in review documents.

$$\text{Polarity} = \begin{cases} 1 & \text{if } P_{ctd} > N_{ctd} \\ 0 & \text{if } P_{ctd} = N_{ctd} \\ -1 & \text{if } P_{ctd} < N_{ctd} \end{cases} \quad (3)$$

where,
$P_{ctd}$ = Frequency of term t in positively tagged documents.
$N_{ctd}$ = Frequency of term t in negatively tagged documents.

Polarity of term was computed in "(3)" 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. Even if a term count varied slightly term was classified as positive or negative. For example if $P_{ctd} = 9$ and $N_{ctd} = 8$, the term was classified as positive.

#### *2) Delta-Term Frequency Inverse Document Frequency (Delta-TFIDF) [4]*

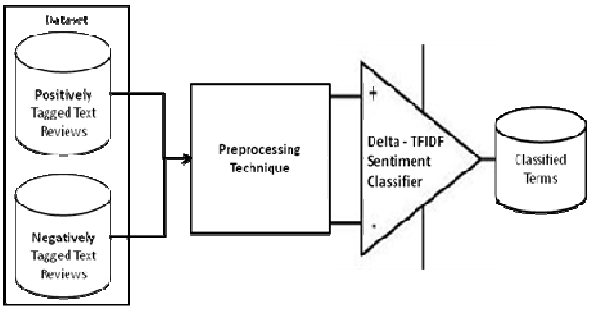



Figure 6 - Sentiment Classification Model using Delta-TFIDF Sentiment Classifier

Figure 6 represents Sentiment Classification Model using Delta-TFIDF Sentiment Classifier. Delta-TFIDF proposed by Martineau Finn was based on term presence count.

$$\text{Polarity} = \begin{cases} 1 & > 0 \\ 0 \text{ if } (P_{ctd} + N_{ctd}) \times \log_2\left(\frac{P_t}{N_t}\right) & = 0 \\ -1 & < 0 \end{cases} \quad (4)$$

where,
$P_{ctd}$ = Frequency of term t in positively tagged documents.
$N_{ctd}$ = Frequency of term t in negatively tagged documents.
$P_t$ = count of positively tagged documents with term t.
$N_t$ = count of negatively tagged documents with term t.

More importance was given in "(4)" to terms that occurred frequently irrespective of its distribution in positively tagged documents or negatively tagged. The later part of the model i.e. log ($P_t/N_t$) contributed to polarity of a term. Term presence count of a term was number of documents that term was present. log($P_t/N_t$) component returned a negative value if a term occurred in more number of positively tagged documents as compared to negatively tagged documents & vice-versa.
If a term was present in equal number of positive and negative document then this component returned zero. Since this value was multiplied with ($P_{ctd}$+$N_{ctd}$), resulting Polarity value was also grounded. These terms were classified as stop words.
It considered overall count of terms in all documents ignoring the frequency distribution of terms across positively and negatively tagged documents. For example if a term was present in more number of negatively tagged

documents as compared to positively tagged document, term was classified as negative. Although the term was present in less number of positively tagged documents, its frequency count in these positively tagged documents may be more which contributed to ($P_{ctd}$+$N_{ctd}$) part. This incorrectly boosted the Polarity value. ($P_{ctd}$ + $N_{ctd}$) being frequency count of terms over all the documents did not correctly relates to second part of the model that dealt with distribution of presence. To calculate polarity of $i^{th}$ term summation of $i^{th}$ element of the vectors was taken in which log ($P_t/N_t$) was common. Sum of ($P_{ctd}$ + $N_{ctd}$) which was always a positive number acted as a boosting factor.
If a term was not present in positively tagged dataset (i.e. $P_t$ = 0) the model returned erroneous results as "log(0)" is invalid number. If a term was not present in negatively tagged dataset (i.e. $N_t$ = 0) the model was affected by divide by zero error

*3) Average Relative Term Frequency Sentiment Classifier (ARTFSC) [27]*

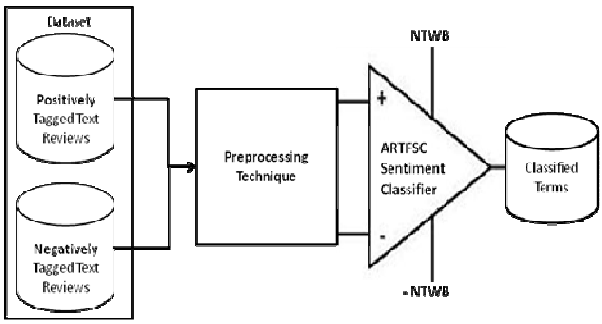


Figure 7 - Sentiment Classification Model using ARTFSC

Figure 7 represents Sentiment Classification Model using ARTFSC Sentiment Classifier. ARTFSC is based on the term frequency count as well as term presence count of term in dataset. It is actually based on average frequency count of term to presence count of term.

$$LDP_t = \log_e\left(\frac{\text{Average count of term t in positively tagged documents}}{\text{Average count of term t in negatively tagged documents}}\right) \quad (5a)$$

$$\text{So } LDP_t = \log_e\left(\frac{\frac{Pctd}{Pt} + 0.001}{\frac{Nctd}{Nt} + 0.001}\right) \quad (5b)$$

$$\text{If } LDP_t \begin{cases} > 0 & \text{Polarity = positive} \\ = 0 & \text{Polarity = neutral} \\ < 0 & \text{Polarity = negative} \end{cases} \quad (5c)$$

where,
$LDP_t$ = Logarithmic differential Polarity.
$P_{ctd}$ = Frequency of term t in positively tagged documents.
$P_t$ = count of positively tagged documents with term t.
$N_{ctd}$ = Frequency of term t in negatively tagged documents.
$N_t$ = count of negatively tagged documents with term t.

Polarity of term was computed in "(5a)", "(5b)" and "(5c)" based on its average frequency count distribution across positively tagged documents and negatively tagged documents. 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

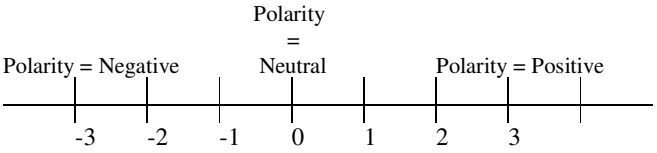


Figure 8 - Sentiment Classification based on $LDP_t$ value.

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 neutral term as shown in figure 8. But if average counts varied slightly also, the term would be classified as positive or negative. To avoid this biased classification, a window was provided as shown in figure 9 for handling neutral words.

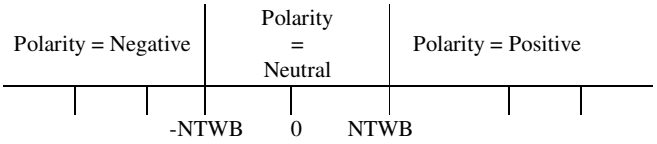


Figure 9 - Sentiment Classification based on $LDP_t$ value with window for neutral words.

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. A window was defined using Neutral Term Window Boundary (NTWB) value.

$$\text{If } LDP_t \begin{cases} > \text{NTWB} & \text{Polarity= positive} \\ = \text{between (-NTWB, NTWB)} & \text{Polarity = neutral} \\ < \text{-NTWB} & \text{Polarity= negative (5d)} \end{cases}$$

where,
$LDP_t$ = Logarithmic differential Polarity.
NTWB = Neutral Term Window Boundary Value.
Accordingly equation "(5c)" was modified to equation "(5d)". That is if the $LDP_t$ value of a term was between –NTWB and NTWB, the term was classified as neutral. If $LDP_t$ value was greater than NTWB then the term was classified as positive. Conversely, if $LDP_t$ value was lesser than −NTWB then the term was classified as negative.
Optimal NTWB value for each Sentiment Classification Model was experimentally determined to maximize accuracy.A term was classified based on its relative average frequency count in positively and negatively tagged documents.

*4) Sentiment Term Frequency Inverse Document Frequency (Senti-TFIDF) [28]*

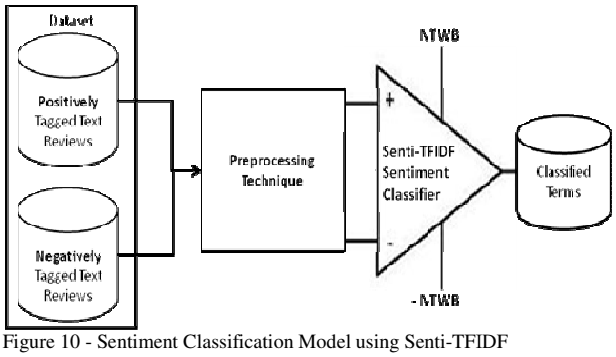



Figure 10 - Sentiment Classification Model using Senti-TFIDF

Figure 10 represents Sentiment Classification Model using Senti-TFIDF Sentiment Classifier. Senti-TFIDF works on the principle 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.

$$LDP_t = \log_e\left(\frac{TFIDF\text{ of term t in positively tagged documents}}{TFIDF\text{ of term t in negatively tagged documents}}\right) \quad (6a)$$

$$LDP_t = \log_e\left(\frac{\left(Pctd \times \log_e\left(\frac{P}{Pt}\right)\right) + 0.001}{\left(Nctd \times \log_e\left(\frac{N}{Nt}\right)\right) + 0.001}\right) \quad (6b)$$

If $LDP_t$ > NTWB — Polarity= positive
= between (-NTWB, NTWB) — Polarity = neutral
< -NTWB — Polarity= negative (6c)

where,
$LDP_t$ = Logarithmic differential Polarity.

$P_{ctd}$ = Frequency of term t in positively tagged documents.
$P_t$ = count of positively tagged documents with term t.
P = Total Number of positively tagged documents.

$N_{ctd}$ = Frequency of term t in negatively tagged documents.
$N_t$ = count of negatively tagged documents with term t.
N = Total Number of negatively tagged documents.

NTWB = Neutral Term Window Boundary Value.

If TFIDF of a term in positively tagged documents was equal to or nearly equal to TFIDF of a term in negatively tagged documents, then the term was classified as neutral using equations "(6a)" and "(6b)". Similar to ARTFSC window was defined using Neutral Term Window Boundary (NTWB) value in equation "(6c)". That is if the $LDP_t$ value of a term was between –NTWB and NTWB, the term was classified as neutral. If $LDP_t$ value was greater than NTWB then the term was classified as positive. Conversely, if $LDP_t$ value was lesser than −NTWB then the term was classified as negative.

*5) Relative Term Frequency Sentiment Classifier (RTFSC)*

Figure 11 represents Sentiment Classification Model using Relative Term Frequency Sentiment Classifier. RTFSC works on the principle 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 the term was assigned positive polarity and vice-versa using equation "(7a)". RTFSC is purely based on the term frequency count of term in dataset.

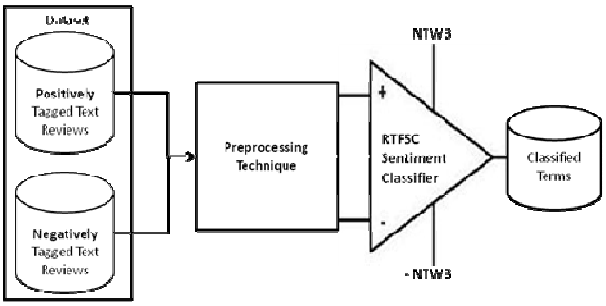



Figure 11 - Sentiment Classification Model using RTFSC

$$LDP_t = \log_e\left(\frac{(Pctd + 0.001)}{(Nctd + 0.001)}\right) \quad (7a)$$

If $LDP_t$ > NTWB — Polarity= positive
= between (-NTWB, NTWB) — Polarity = neutral
< -NTWB — Polarity= negative (7b)

where,
$P_{ctd}$ = Frequency of term t in positively tagged documents.
$N_{ctd}$ = Frequency of term t in negatively tagged documents.

$LDP_t$ = Logarithmic differential Polarity.
NTWB = Neutral Term Window Boundary Value.

If term frequency 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. Similar to ARTFSC and Senti-TFIDF, a window was defined using Neutral Term Window Boundary (NTWB) value. That is if the $LDP_t$ value of a term was between –NTWB and NTWB, the term was classified as neutral. If $LDP_t$ value was greater than NTWB then the term was classified as positive. Conversely, if $LDP_t$ value was lesser than NTWB then the term was classified as negative.The classifier models that were used in experimentation are summarized in table 4.

Table 4: Sentiment Classifier Models

| No | Sentiment Classifier | Classification Criteria for term | Based on |
|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | TSC [3 ] | Traditional Sentiment Classifier | Max ($Pctd, Nctd$ ) | Frequency Count |
| 2 | Delta-TFIDF [4] | Delta TFIDF Sentiment Classifier | $(Pctd + Nctd) \times \log_e\left(\frac{Pt}{Nt}\right)$ | Relative Presence Count |
| 3 | ARTFSC [27] | Average Relative Term Frequency Sentiment Classifier | $\log_e\left(\frac{\frac{Pctd}{Pt} + 0.001}{\frac{Nctd}{Nt} + 0.001}\right)$ | Relative Average Count (ie frequency and presence count) |
| 4 | Senti-TFIDF [28] | Senti-TFIDF Sentiment Classifier | $\log_e\left(\frac{(Pctd \times \log_e(\frac{P}{Pt})) + 0.001}{(Nctd \times \log_e(\frac{N}{Nt})) + 0.001}\right)$ | Relative TFIDF values of terms |
| 5 | RTFSC [29] | Relative Term Frequency Sentiment Classifier | $\log_e\left(\frac{Pctd + 0.001}{Nctd + 0.001}\right)$ | Relative Frequency Count |

where,
$P_{ctd}$ = Frequency of term t in positively tagged documents.
$P_t$ = count of positively tagged documents with term t.
P = Total Number of positively tagged documents.

$N_{ctd}$ = Frequency of term t in negatively tagged documents.
$N_t$ = count of negatively tagged documents with term t.
N = Total Number of negatively tagged documents.

## 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 processed datasets. Various preprocessing techniques such as handling apostrophe, removing punctuations and removing stopwords were applied on the above mentioned unprocessed dataset.

Sentiment Classification was performed on unprocessed and various preprocessed datasets. A list of terms that occurred in the reviews was prepared. A term is entered once in this term list although it may appeared 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 0. These vectors were used to calculate term polarity for the terms in the term list.

Polarity was calculated using Plain Sentiment Classifier (PSC), Delta Term Frequency Inverse Document Frequency (Delta-TFIDF), 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 by our model as positive if total number of positive terms in the reviews were more than negative terms. Similarly a review was classified as negative if total number of negative terms in the reviews were more than positive terms.

If a review was originally tagged as positive & also classified as positive then it contributed to True Positive in confusion matrix.

If a review was originally tagged as negative & also classified as negative then it contributed to True Negative. If a reviews was originally tagged as positive but classified as negative then it contributed to False Negative.

If a reviews was originally tagged as negative but classified as positive then it contributed to False Negative. Below mentioned experiments were performed on 15 Sentiment Classification models. Each of these models had one of the five Sentiment Classifiers (PSC, Delta-TFIDF, ARTFSC, Senti-TFIDF & RTFSC) applied on document dataset of a type. That is unprocessed, words with apostrophe handled & punctuations removed & apostrophe handled with punctuations & stopwords removed.

### A. *Experiment 1*

Experiment 1 was conducted to determine that a term t should be classified as neutral word if $LDP_t$ exactly equals zero or it was within a specified range defined by –NTWB and NTWB values explained in figure 3. For this accuracy was computed using, 10 Fold Cross Validation (10 fold CV). The NTWB range was varied between 0 to 5 at step of 0.5 and simultaneously -NTWB 0 to -5 at step of -0.5.

To calculate accuracy dataset was divided in 10 parts. At every fold this 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. Dataset was divided in 10 parts. At every fold this 10% dataset was used for testing and remaining 90% dataset was used for training the classifier. NTWB and –NTWB values were now set as determined in experiment 1 for terms to be classified as neutral. Confusion matrix was constructed as well as accuracy was calculated at every fold and then averaged to form the accuracy of the model at that value of NTWB

## V. Results and Discussion

Optimal Neutral Term Window Boundary (NTWB) values for all 15 Sentiment Classification models that were experimented were determined. More words were classified as neutral if NTWB value is larger resulting to lesser number of opinionated words and vice-versa. So an optimal value of NTWB was determined for each model.

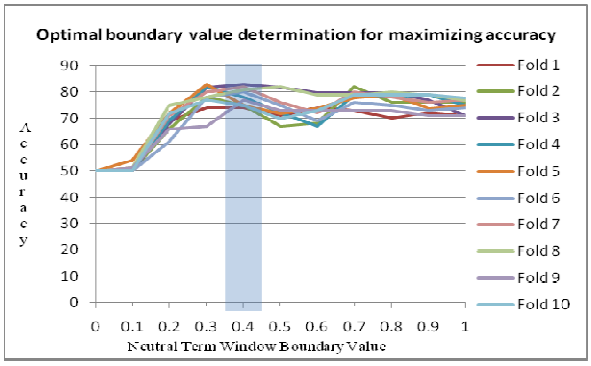
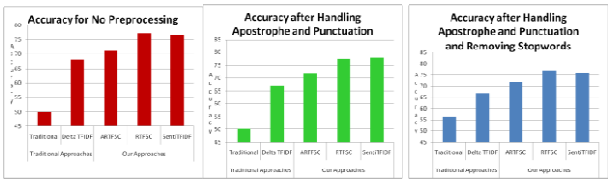



Figure 12 - Optimal boundary value determination for SentiTFIDF classifier on apostrophe handled and punctuations removed dataset for maximizing accuracy.

Figure 12 represents accuracy graph for determining optimal window boundary value for SentiTFIDF classification model. Apostrophe handled and punctuations 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. It can be observed from figure 12 that average accuracy was largest at 0.4 window boundary value. Similarly optimal window boundary values for all 15 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 boundary was 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.

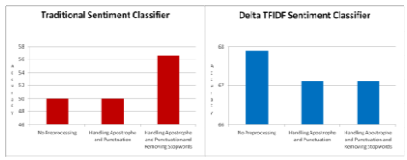
Window boundary values was not applicable (or set to zero) for Traditional Sentiment Classifier (TSC) as it is not based on relative or ratio based mathematical model.



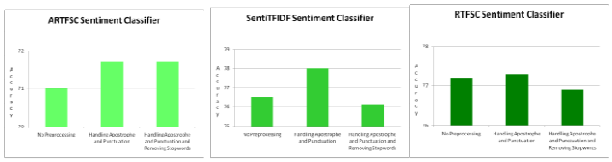

(a) (b) (c)

Figure 13 – Classifier Performance Evaluation.

Classifier performance was evaluated by measuring accuracy of all classifier. Figure 13(a) represents accuracy of all the classifiers for an unprocessed input dataset. Similarly the input dataset in figure 13(b) was apostrophe handled and punctuation removed dataset. Figure 13(c) illustrates the accuracy when apostrophe handled, stopwords and punctuation removed dataset was provided as input. It can be observed that our sentiment classifiers RTFSC, Senti-TFIDF and RTFSC performed much better than the Traditional Sentiment Classification model (TSC) for all datasets. Accuracy of our models is also better than Delta-TFIDF Sentiment Classifier which is also based on term weighting. Of the three models proposed, Relative Term Frequency Sentiment Classifier (RTFSC) has highest accuracy then Sentiment Term Frequency Inverse Document Frequency (Senti-TFIDF) ranked second and Average Relative Term Frequency Sentiment Classifier (ARTFSC) positioned third.




(a) (b)




(c) (d) (e)

Figure 14 – Effect of preprocessing techniques on sentiment classifier

All the graphs in Figure 14 represent accuracy for movie document dataset with preprocessing techniques. Each of the preprocessing technique was evaluated on all sentiment classifiers. For all classifier the accuracy obtained from unprocessed dataset was lowest.

Figure 14(a) shows that Traditional sentiment classifier did not show any improvement for apostrophe handled dataset but accuracy increased when stopwords were removed.

Figure 14(b) shows that Delta-TFIDF, a comparable term weighting technique did not show any improvement for any type of preprocessing.

Figure 14(c), 14(d) and 14(e) represent the performance of our classifier (ARTFSC, Senti-TFIDF and RTFSC) for different preprocessing techniques. Accuracy increases from unprocessed dataset to when apostrophe handled and punctuations cleaned dataset is provided as input. It can be observed that the accuracy goes on increasing when the level of preprocessing is increasing.

When level of preprocessing is still increased to removing stopwords along with removing punctuations and handling apostrophe, accuracy of models either drops or some remain same

Our classifiers provided neutral term window boundary (NTWB) value due to which stopwords were efficiently handled at the time of classification itself. Unlike traditional classifiers separate preprocessing task of stopwords removal is not needed for our classifier.

A comparable term weighting technique Delta-TFIDF also shows a better performance when stopwords are not removed but still its accuracy is lesser than all our classifier.

## VI. CONCLUSION AND FUTURE WORK

Accuracy increases from unprocessed dataset to when preprocessing levels is further increased to handling words with apostrophe and removing punctuation symbols. The

accuracy goes on increasing when the level of preprocessing is increasing.

When level of preprocessing is still increased to removing stopwords along with removing punctuations and handling apostrophe, accuracy of models either drops or some remain same except for Traditional Sentiment Classifier (TSC). Removal of stopwords was needed in traditional classification model. Our models that are, ARTFSC, Senti-TFIDF and RTFSC handle stopwords using NTWB window at the time of classification itself, thus have hardly any impact of stopwords removal in preprocessing.

Our Sentiment Classification Models RTFSC, Senti-TFIDF and RTFSC performed much better than the Traditional Sentiment Classification model (TSC). Accuracy of our models is also better than Delta-TFIDF Sentiment Classification Model. Out of our three models, Sentiment Term Frequency Inverse Document Frequency (Senti-TFIDF) has highest accuracy of 78%. Then Relative Term Frequency Sentiment Classifier (RTFSC) ranked second with accuracy 77.2% and Average Relative Term Frequency Sentiment Classifier (ARTFSC) positioned third with accuracy of 71.7%.

Although the accuracies of surveyed techniques cannot be directly compared as the experimental parameters may vary, the sentiment classification model with apostrophe handled and punctuations removed dataset applied to Sentiment Term Frequency Inverse Document Frequency classifier (Senti-TFIDF) performed better than most existing techniques.

Our classification models are based on appropriate Preprocessing Techniques and Sentiment Classifiers 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.

[30] T. Magadza, A. Mukwazvure, and K.Supreethi, "Exploring Sentiment Classification Techniques in News Articles," *In International Journal of IT & Knowledge Management,* vol. 08, issue no. 01, pp. 55-58, 2014.