Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Performance analysis of new word weighting procedures for opinion mining

G. R. BRINDHA[†‡], P. SWAMINATHAN, B. SANTHI

(*School of Computing, SASTRA University, Thanjavur 613401, India*)

[†]E-mail: brindha.gr@ict.sastra.edu

Received Aug. 30, 2015; Revision accepted Feb. 16, 2016; Crosschecked Oct. 17, 2016

Abstract: The proliferation of forums and blogs leads to challenges and opportunities for processing large amounts of information. The information shared on various topics often contains opinionated words which are qualitative in nature. These qualitative words need statistical computations to convert them into useful quantitative data. This data should be processed properly since it expresses opinions. Each of these opinion bearing words differs based on the significant meaning it conveys. To process the linguistic meaning of words into data and to enhance opinion mining analysis, we propose a novel weighting scheme, referred to as inferred word weighting (IWW). IWW is computed based on the significance of the word in the document (SWD) and the significance of the word in the expression (SWE) to enhance their performance. The proposed weighting methods give an analytic view and provide appropriate weights to the words compared to existing methods. In addition to the new weighting methods, another type of checking is done on the performance of text classification by including stop-words. Generally, stop-words are removed in text processing. When this new concept of including stop-words is applied to the proposed and existing weighting methods, two facts are observed: (1) Classification performance is enhanced; (2) The outcome difference between inclusion and exclusion of stop-words is smaller in the proposed methods, and larger in existing methods. The inferences provided by these observations are discussed. Experimental results of the benchmark data sets show the potential enhancement in terms of classification accuracy.

Key words: Inferred word weight, Opinion mining, Supervised classification, Support vector machine (SVM), Machine learning
http://dx.doi.org/10.1631/FITEE.1500283 **CLC number:** TP391

1 Introduction

The exponential growth of social networks has rapidly increased web content (Barnes and Bohringer, 2011). The enormous amount of web data should be used effectively avoiding the waste of the available storage space. Motivated by the voluminous data, researchers are studying, investigating, and mining the web content. For mining, new methods are used by web users and companies to understand customers and enhance their products or services. Interesting studies concerning text analysis using different approaches are the basic motivation for text mining (Church and Hanks, 1989; Geng and Hamilton, 2006; Armstrong *et al*., 2009). These studies about opinion mining follow the text mining processes along with their own specific analysis (Boiy *et al*., 2007; Tsutsumi *et al*., 2007; Boiy and Moens, 2009; Paltoglou and Thelwall, 2010; Saif *et al*., 2012). In text classification, the process starts with preprocessing. The following process is to train the classifier and then the testing process proceeds. Recent works focused on the procedure of term selection, since each term has its own value in conveying its opinions (Das and Chen, 2001; Debole and Sebastiani, 2003). The value of a term depends on its contribution to the review document and the significance of the meaning it conveys. Hence, in between the two stages of term selection and classifier training, the term ‘weighting

‡ Corresponding author
ORCID: G. R. BRINDHA, http://orcid.org/0000-0001-5911-8327

scheme' should be included (Das and Chen, 2001). To obtain a more efficient classification during text categorization, feature weighting plays an important role (Debole and Sebastiani, 2003).

The calculation of the term weight in information retrieval has been a focal point of much research, since this is an aspect of vital importance related to the relevance of the document to which the term belongs (Debole and Sebastiani, 2003; Geng and Hamilton, 2006; Armstrong *et al*., 2009; He *et al*., 2011; Esparza *et al*., 2012). Word association is emphasized and practiced for many applications (Li *et al*., 2009). The popularly known term frequency-inverse document frequency (TF-IDF) indicates that rare words are no less important than frequently used words, that multiple appearances of a word in a document are no less important than a single appearance, and that the same number of words in a lengthy document is no more important than that in a short document (Mladenić and Grobelnik, 1998; Maas *et al*., 2011). Apart from TF-IDF, some researchers applied probability based statistics to determine the weighted relevancy for a term of a document such as BM25 (best match25 or OkapiBM25) (Manning *et al*., 2008). Traditional information theory and statistics such as chi square test, gain ratio, and information gain are used to weight the terms during the training phase, noted in few studies (Das and Chen, 2001; Debole and Sebastiani, 2003; Li *et al*., 2009). A survey lists the probability based measures and their properties to mine the usefulness of information. It gives details about subjective significance measures along with elimination of uninteresting patterns, semantic measures, and measures for summaries (Geng and Hamilton, 2006).

Support vector machine (SVM), the optimized classifier of machine learning, has been used by many researchers (Das and Chen, 2001; Pang *et al*., 2002; Debole and Sebastiani, 2003; Pang and Lee, 2004; Blitzer *et al*., 2007; Zaidan *et al*., 2007; Paltoglou and Thelwall, 2010). Machine learning techniques automatically categorize the text efficiently through programs by labeling texts for a domain (particular application area) with defined class sets (positive and negative review sets). A model for a class set is built by learning the characteristics of the training set. Then the capacity of categorization is checked by assigning it to a test set and verifying the percentage of decisions by the classifier for the data set. Generally, TF-IDF and its variants are used in existing studies to weigh the words based on their frequencies. For the feature selection process, statistical methods are used to compute the information gain. These computations are used to for a single and robust weighting combination. Since the weight of each word is inferred from two angles (such as frequency and information gain) using statistical methods, the proposed combination is called the 'inferred word weighting' method (IWW).

The structure of the opinion classifier has two modules (Fig. 1):

The first module calculates the weights for the words using the following computations:

1. Words in document representations contain:

(1) Significance of a word in a document (SWD based on frequency: term frequency-SWD(2), and normalized term frequency-SWD(3));

(2) Significance of a word in expression (SWE based on information gain: pointwise mutual information (PMI), odds ratio (OR), frequency and odds

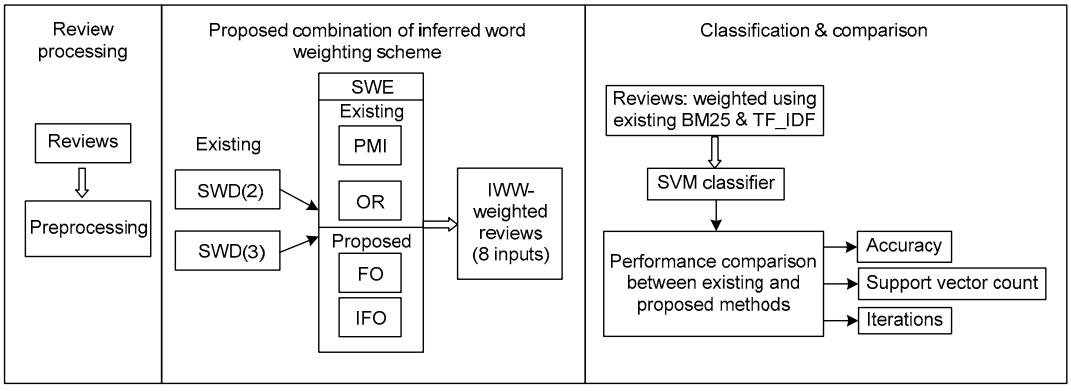

Fig. 1 Workflow of the proposed weighting and classification

(FO), and improved frequency and odds (IFO)).

2. The second module classifies reviews as either positive class or negative class. The classification is the process of building model by training and testing document representation using the classifier.

When TF-IDF and its variants are used for term weighting, they are not stable when there is a change in the stop-word list, and dramatic alteration occurs in term weights (Manning *et al*., 2008). Some of previous studies prepared their data sets by removing stop-words for different reasons (Nigam *et al*., 2000; Pang and Lee, 2004; Maas *et al*., 2011; Saif *et al*., 2012). However, none of them analyzed stop-word inclusion in terms of frequency based weighting, to enhance the classification performance. Since the proposed methods are frequency balancing methods, they are motivated to do the weighting by including stop-words. The proposed IWW methods weight and handle both feature sets (bag-of-words including or excluding stop-words) very well and provide improved accuracy when stop-words are included in the feature set. Existing methods such as TF-IDF and BM25 also check through classification for their performance when the stop-words are included in the bag-of-words. From these verifications, new perspectives on the proposed methods are obtained, which are useful for further studies. Eight combinations of weight computations and two existing methods are applied on data sets to obtain the weighted word matrix. Individually, each weighted matrix (including or excluding stop-words, totally 20 matrices) becomes the input for SVM classification.

2 Proposed weighting system

The inferred word weighting system weights each word through two types of computations in phase I, SWD and SWE. SWD is computed through a term frequency function and SWE is computed based on statistical values of terms in positive and negative documents.

2.1 Foundation

The positive review document set is represented by C^1 and the negative one by C^2. Let $T=\{t_1, t_2, \ldots, t_m\}$ be the term set, which contains distinctive words in C^1 and C^2. The set T is called the bag-of-words and its corresponding weight set will be $d_k=\{w_{1k}, w_{2k}, \ldots, w_{mk}\}$. The proposed weighting system IWW_{jk} is described as

$$\mathrm{IWW}_{jk}=\mathrm{SWD}(t_j,d_k)\cdot\mathrm{SWE}(t_j), \tag{1}$$

where SWD denotes the significance of term t_j in document d_k and $\mathrm{SWE}(t_j)$ the significance of term t_j in document d_k in expressing sentiment. These two parts are used to compute the relevancy of each term in the reviews based on the probability, given as below:

$P(C^i)$: the probability that a document fits in class C^i;

$P(t_j)$: the probability that a term t_j occurs in a class;

$P(t_j, C^i)$: the combined probability that a document contains term t_j and occurs in a class;

$P(t_j|C^i)$: given the constraint that a document belongs to class C^i, the probability that a term t_j fits in the document;

$P(t_j|\rightharpoonup C)$: given the constraint that a document does not belong to class C^i, the probability that a term t_j fits in the document.

To compute the above probabilities, the following statistical notations are used:

ct_{jk}: the occurrence count of term t_j in weight set d_k;

p^i_j: the document count of occurrences of term t_j fitting in class C^i;

q^i_j: the document count of occurrences of term t_j not fitting in class C^i;

n_i: the number of document fits in class C^i.

Using the above statistical notations, the probabilities are computed as follows:

$$P(C_i)\approx\frac{n_i}{n_1+n_2},\ P(t_j)\approx\frac{p^1_j+p^2_j}{n_1+n_2},\ P(t_j,C^i)\approx\frac{p^i_j}{n_1+n_2},$$

$$P(t_j\mid C^i)\approx\frac{p^i_j}{n_i},\ P(t_j\mid\rightharpoonup C)\approx\frac{q^i_j}{n_1+n_2-n_i}.$$

2.1.1 SWD(t_j, d_k) computation

The terms in a document convey information about the content or the domain to which the document belongs. Based on this thought, term frequency calculation is employed in many analyses of text processing. The significance of the term in the

document is computed in two different ways (Manning *et al*., 2008). In our study, SWD includes the term frequency (TF) and normalized term frequency (NTF), are denoted by Eqs. (2) and (3). In further sections, TF is denoted as SWD(2) and NTF as SWD(3).

TF is defined as follows:

$$\mathrm{SWD}(t_j, d_k) = \mathrm{ct}_{jk}. \tag{2}$$

The above-mentioned binary computation fixes SWD(t_j, d_k) based on the frequency of word t_j in document d_k.

NTF provides normalized computation for the occurrences of a term, which is proportional to its document size:

$$\mathrm{SWD}(t_j, d_k) = \gamma + \frac{\gamma \cdot \mathrm{ct}_{jk}}{\max_i \mathrm{ct}_{ik}}, \tag{3}$$

where γ is given a value between 0 and 1, normally set to 0.4, although earlier studies fix the value at 0.5. The term γ is used for smoothing which dampens the part of the second term and also is observed as a scaling down of t by the largest t value in document k (Manning *et al*., 2008). The concept is to evade a large variation in NTF from small alterations in ct_{jk}. The numerator term in Eq. (3) is given the value of the maximum frequency of the ith term, when compared to all other frequencies of terms in document k. Generally, the term frequencies are more in longer reviews because of the repetition of the same words. For example, when a review k is copied twice and a new review k' created, although it is more relevant to k, the use of Eq. (2) would be given a weight as high as k. Now by including the smoothing term, Eq. (3) mitigates this variation. Hence, instead of using other normalization methods, the proposed SWD includes NTF.

2.1.2 SWE(t_j) computation

The computation of the frequency (such as ITF) will not justify the weight calculation for review documents, as only the relevancy of the terms of the document can provide the perfect weight. In many studies, SWD(t_j) methods are used to select features, since these methods provide good variations in the weight computation based on the significance of the terms. So, the proposed study provides a weight to the terms combining SWD and SWE methods.

PMI is used in various text mining studies to select relevant sentences and features (Simmons *et al*., 2011; Sheikh and Conlon, 2012). In some studies PMI was used as mutual information (MI) (Li *et al*., 2009), but the correct term is as given in Eq. (4) (Xu *et al*., 2007). PMI can be used for checking the true relationship between the term and the document in which the term exists. If the joint probability is higher than the individual probability, then the term and the document have a true relationship. If the joint probability and individual probability are more or less equal, then there is no interesting relationship between them (Li *et al*., 2009). For the term t_j in a document that belongs to class C^i, PMI is

$$\mathrm{PMI}(t_j, C^i) = \log \frac{P(t_j, C^i)}{P(t_j) \cdot P(C^i)}. \tag{4}$$

PMI(t_j, C^i) is estimated as

$$\begin{aligned}\mathrm{PMI}(t_j, C^i) &= \log \frac{p_j^i / (n_1 + n_2)}{(p_j^1 + p_j^2)/(n_1 + n_2) \cdot n_i / (n_1 + n_2)} \\ &= \log \frac{p_j^i (n_1 + n_2)}{(p_j^1 + p_j^2) n_i}.\end{aligned} \tag{5}$$

Now SWE(t_j) through PMI is

$$\mathrm{SWE}(t_j) = \max\{\mathrm{PMI}(t_j, C^1), \mathrm{PMI}(t_j, C^2)\}. \tag{6}$$

In general, OR is used to compute the ratio between the odds of the relevant and irrelevant documents, finalized by taking the log. The final value after taking the log to the ratio will be (1) zero, if the word has equal odds of occurring in the relevant class and irrelevant class, and (2) positive, if the term occurs in the relevant document. In the proposed SWE, OR computes the ratio between the odds of the term occurring in the positive class and that in the negative class. Then the logs to both values are computed. Finally the larger one of these two values is fixed as the weight of the term.

The association between term t_j and class C^i is defined as follows (Sebastiani, 2002; Manning *et al*., 2008):

$$\mathrm{OR}(t_j, C^i) = \log \frac{P(t_j|C^i)(1-P(t_j|\rightarrow C^i))}{(1-P(t_j|\rightarrow C^i))P(t_j|\rightarrow C^i)}. \quad (7)$$

$\mathrm{OR}(t_j, C^i)$ is estimated as

$$\mathrm{OR}(t_j, C^i) \approx \log \frac{p_j^i(n_1+n_2-n_i-q_j^i)}{(n_k-p_j^i)q_j^i}. \quad (8)$$

Now $\mathrm{SWE}(t_j)$ through OR is

$$\mathrm{SWE}(t_j) = \max\{\mathrm{OR}(t_j, C^1), \mathrm{OR}(t_j, C^2)\}. \quad (9)$$

Frequency and odds (FO): FO is derived from OR. Eq. (7) contains the product of two terms in the numerator. These are (1) the probability of the term occurring in a document that fits in class 1 (denoted as P), and (2) the probability of the term occurring in a document that does not belong to class 1 (denoted as $1-P$). The denominator includes the same estimate for class 2. From these parts FO takes only the constraint probability of term occurrence in class 1 in the numerator and the constraint probability of term occurrence in class 2. This probability for relative occurrence of a term provides the maximum gain estimate. When this simple estimate is used as it is, then the probability of a high-frequency term will be high and the probability of a low-frequency term gets 0 as the relative occurrence estimate. In other words, if the numerator of this ratio is less than the denominator, then the value will be between 0 and 1. Hence, to obtain a good estimate, the product of the constraint probability is included in FO. Hence, FO estimates a high value for a term with respect to class 1 if it occurs frequently in class 1 and infrequently in class 2.

The FO for term t_j in class C^i is identified as

$$\mathrm{FO}(t_j, C^i) = P(t_j|C^i)\log \frac{P(t_j|C^i)}{P(t_j|\rightarrow C)}. \quad (10)$$

$\mathrm{FO}(t_j, C^i)$ is estimated as

$$\mathrm{FO}(t_j, C^i) \approx \frac{p_j^i}{n_i}\log \frac{p_j^i(n_1+n_2-n_i)}{q_j^i n_i}. \quad (11)$$

Now $\mathrm{SWE}(t_j)$ through FO is

$$\mathrm{SWE}(t_j) = \max\{\mathrm{FO}(t_j, C^1), \mathrm{FO}(t_j, C^2)\}. \quad (12)$$

The above computation relates the frequency and the log of proportion. Hence, FO is biased towards the frequency value, which is motivated to improve the frequency and odds computation.

Improved frequency and odds (IFO): Since FO estimation cannot achieve the expected results (as discussed in the result analysis section), the improved version of FO is introduced. Even though the product of the constraint occurrence makes the model a meaningful one, some sort of smoothing is still needed. This means that the high-frequency estimate should be lowered and the low-frequency estimate be increased by estimating the frequency and odds in terms of α, which is given a value between 0 and 1. In our study, α is used to distribute the features uniformly (note that data distribution differs according to the feature set). Then the probability is updated based on α. Based on the outcome, the decision is made whether the belief is weak or strong. This outcome differs with respect to the size of the bag-of-words (which will be discussed in the result analysis section), and thus α is given a range of values.

The balanced frequency and odds between term t_j and class C^i is defined as follows:

$$\mathrm{IFO}(t_j, C^i) = P(t_j|C^i)^{\alpha}\log\left(\frac{P(t_j|C^i)}{P(t_j|\rightarrow C)}\right)^{1-\alpha}. \quad (13)$$

$\mathrm{IFO}(t_j, C^i)$ is estimated as

$$\mathrm{IFO}(t_j, C^i) \approx \left(\frac{p_j^i}{n_i}\right)^{\alpha}\log\left(\frac{p_j^i(n_1+n_2-n_i)}{q_j^i n_i}\right)^{1-\alpha}. \quad (14)$$

Now $\mathrm{SWE}(t_j)$ through IFO is

$$\mathrm{SWE}(t_j) = \max\{\mathrm{IFO}(t_j, C^1), \mathrm{IFO}(t_j, C^2)\}. \quad (15)$$

Therefore, α refines the weight between the frequency and the odds. After checking α with the range of values (10-fold cross validation for each α), finally 0.1 is fixed for further computation. The ground for fixing the α value and the impact of fixing α=0.1 on the weight matrix of classifier are discussed in the result analysis section.

2.2 Novelty of the proposed work

In existing studies TF-IDF is used for weighting terms in documents that belong to a corpus. For this background discussion, the documents are termed 'reviews', which are either positive or negative. The SWD(#) contains the variants of TF, the popular term weighting computation, which is a part of the TF-IDF weighting technique. The review based term weight, TF, counts the number of times a term used in a review, and the corpus based term weight, IDF, represents the count of reviews from the whole data set that contains a term. The frequency of the term alone cannot provide a perfect weight, since a more frequent term does not need to not be weighted with a high value. So, to balance the frequency based weight, TF variants include different normalized computations. Hence, to demonstrate the effect of these two computations combined with SWE during classification, both the plain frequency and normalized frequency are included in SWD(#).

The novelty of our proposal which replaces IDF with SWE(t_j) also includes statistical computations, such as PMI and OR (existing) and FO and IFO (novel). The reasons behind the proposal of using SWE are discussed below.

The inverse document frequency of a rare word is high, whereas that of a frequent word is low. For example, consider a corpus which includes 806 791 documents. Table 1 provides sample words, the number of documents in which the word occurs, the IDF value, and the weight of the same words computed by the PMI method (calculation is given in Section 2.2.2). Table 1 shows that the IDF weight is inversely proportional to the document frequency. At the same time, it linearly increases with the document frequency. Note that the DF for the word 'Good' is higher than that for words 'Mobile' and 'Warranty', but its PMI and IDF weights decrease. Therefore, IDF computation is justified in this angle. IDF represents the weight of a word based on the number of documents in the corpus as a whole; however, SWE(t_j) computes the weight separately for a positive class and a negative class and takes the larger weight (note the weight difference for the word 'Notepad'). The concept behind PMI (SWE(t_j)) computation is that each word in the document provides an information gain to the class it belongs to, which is different from class to class.

Table 1 Sample word weight by IDF and PMI

Word	Document frequency (DF)	Inverse document frequency (IDF)*	PMI
Mobile	18 165	1.65	0.22
Notepad	6723	2.80	0.45
Warranty	19 241	1.62	0.21
Good	25 235	1.50	0.18

* IDF=log(DF/n), with n being the number of documents in the corpus. Here n=806 791. PMI: pointwise mutual information

IDF computation is not appropriate for review classification. Weighting methods such as TF-IDF and BM25 perform well in studies on topic identification, since frequency is the base used in such areas. The reasons and the motivation for selecting these methods are discussed as part of the starting phase in the result analysis section.

Another novelty lies in selection of NTF as SWD(#), while in most studies TF is used. Actually, when there are changes in the bag-of-words (stop-words included), it is hard to tune NTF computation. However, IWW combines SWE(t_j) with NTF for tuning and obtaining a precise weight. This tuning is possible for SWE(t_j) because the logarithm based probability treats the stop-words (including the bag-of-words) differently from those with a skewed distribution. Though stop-words are not representatives for the positive or negative class, they help in precise classification for large data sets. The reason is that the increase in dimension allows the classifier to identify the reviews with more gaps between them. Therefore, the analysis of performance verifications of weight methods (stop-words excluded) extends to the analysis of stop-word exclusions. Only a few researchers (Nigam *et al*., 2000; Pang and Lee, 2004; Maas *et al*., 2011; Saif *et al*., 2012) considered text processing by including stop-words; however, they did not analyze the effects of stop-words using different weighting methods or determine the effects of the methods, as done in this study.

3 Implementation

The weighted terms are fed to a classifier which develops a model and classifies the unlabeled data set. Table 2 describes the popularly known data sets used in this proposal for opinion mining. Previously, the

Cornell movie reviews (Pang *et al*., 2002), Amazon product reviews (Blitzer *et al*., 2007), and Stanford movie reviews (Maas *et al*., 2011) were introduced and now many studies (Debole and Sebastiani, 2003; Paltoglou and Thelwall, 2010; Esparza *et al*., 2012) used these popularly known data sets for opinion mining analysis.

Table 2 Data set description

Corpus	Corpus size	Training	Testing
Cornell movie review data set[1]			
Positive	1000	500	500
Negative	1000	500	500
Multi-domain[2]			
Positive	4000/domain; 4 domains; total: 16 000	2000	2000
Negative		2000	2000
Stanford large movie review[3]			
Positive	25 000	12 500	12 500
Negative	25 000	12 500	12 500

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[2] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[3] http://ai.stanford.edu/~amaas/data/sentiment/

3.1 Preprocessing

Before applying IWW on the data sets, preprocessing is required to have a fine tuned weighted matrix. During this preprocessing stage, the data set is processed into clean words (Manning *et al*., 2008). Clean word processing includes fine tuning such as special character removal, case folding (upper case to lower case), and tokenization (converting the sentences into words). The bag-of-words (unique word set) is created from the corpus, which is the basic assumption of text mining. The bag-of-words with feature set size 43 769 is retrieved from the whole data set.

The grammatical sentence structure of the reviews in the document is not considered, and the words are processed without affecting the results (Lee *et al*., 2010).

3.2 Word weighting process

The next step is to compute the weight for each feature IWW_{jk}, as explained in Section 2.1, using Eq. (1) based on the combination of frequency and expressions of opinions. This weighting process is computed in the combination of two word frequency functions (SWD(#)) each with four statistical functions SWE(t_j).

3.3 Classification

The weighted values are given as input to SVM for binary classification. SVM provides optimized classification results with the support of various kernels based on data set usages. Researchers are using some of the popular tools such as LIBSVM (Debole and Sebastiani, 2003), LIBLINEAR (Paltoglou and Thelwall, 2010), SVMlight (Pang *et al*., 2002; Ng *et al*., 2006), and SVM Torch (Das and Chen, 2001). Other than these popular tools, many other tools are also using LIBSVM as a base for classification. The difference between LIBSVM and LIBLINEAR is that the latter does not use kernels and it is primarily designed for linear data set usage. In our study, to check the performances with different kernels, LIBSVM is used. The results discussed here are the outcome from using linear kernels. Other kernels such as RBF, sigmoid, and polynomial provide less accuracy than a linear kernel.

Ten-fold cross validation is used to evaluate the performance of the classifier. In 10-fold cross validation, the whole data set is randomly divided into 10 equal subsets. From these sets, nine subsets are used for training to obtain the model. The remaining single set is used to test the model (validation). In the same way, each of the subsets is treated as a testing set when the remaining sets are used for training. The outcome of the folds is averaged to obtain the final estimate.

To check the influence of the proposed statistical method on classification, existing methods such as TF-IDF and BM25 are applied on the corpuses for weight computation. BM25 has been widely used by researchers because of its good performance and the high flexibility of deriving versions of the basic OkapiBM25 (Salton and Buckley, 1998; Pang and Lee, 2004). The influences of the combination SWD(#)*SWE(t_j) during the classification are discussed and compared with the BM25 and TF-IDF outcomes in the next section.

4 Result analysis

This section provides evaluation results and comparative analysis with existing weighting

methods. In Section 4.1 we analyze the classification results of the corpus that excludes stop-words and is weighted with the proposed methods. In Section 4.2 we analyze the classification performance of the corpus that includes stop-words and is weighted with the proposed methods. In Section 4.3 the results of the proposed methods are compared with those of existing studies. The facts behind the varying results for the same data sets are discussed.

Though the TF-IDF weighting method has been used broadly, for large data sets BM25 provides better performance (Esparza *et al*., 2012). Therefore, for performance comparison of the proposed methods, TF-IDF and BM25 are used to weight the corpuses and their effects on the classifier are obtained. Before starting the phases, the performance of the IFO method with a varying α is discussed since the IFO results are analyzed in the following phases.

In the IFO function, a parameter α is introduced to balance the frequency and odds. α is a learned parameter, validated through a 10-fold cross validation to obtain the best value. Each execution provides an α value from 0 to 1. The average of α is 0.1, with which the highest accuracy is achieved. The learning process for each domain provides a different α value according to the feature set size. For a feature set size around 1000, α should be set to 0.5 for better performance. If the feature set size is around 25 000, then α=0.01 provides the best accuracy. The data sets used in this study are weighted with α=0.1, since the feature set size is above 35 000. So, α should be fixed based on the feature set size to obtain better performance.

4.1 Phase I: stop-word exclusion performance analysis

For any data set, it is not always true that high-frequency features are good features and convey important information to the document; instead, they may deteriorate the classifier performance. Hence, stop-words such as 'a', 'an', and 'the' are removed (Tong, 2001; Debole and Sebastiani, 2003). Stop-word removal is applied to all the three data sets and their classification results are discussed.

The accuracies of the Cornell movie reviews based on four SWE(t_j), each with two SWD(#), are compared with those of TF-IDF and BM25 (Fig. 2). Note that in the following discussions about different cases, the formulas are indicated based on the equation number; e.g., SWD(2) indicates Eq. (2) of the first SWD(#).

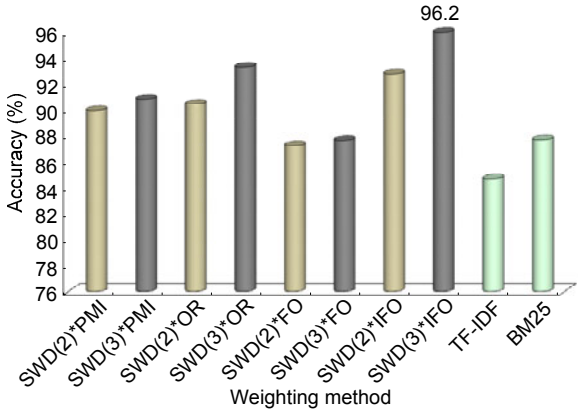

Fig. 2 Weight function performance comparison for Cornell movie reviews

Case 1: The proposed function IFO (96.2) performs better than PMI (90.8) and OR (93.3). IFO and OR provide better accuracy since IFO is the derived version of OR.

Case 2: The FO performances with SWD(2) and SWD(3) are notably lower when compared with those of the other four SWE(t_j). The values of SWD(3)*FO and SWD(2)*FO are approximately equal. The frequencies of each word and its odds, i.e., the occurrences in positive and negative class documents, are given equal weights. This result leads to proper tuning of the weight factor for frequency and its odds.

The above two cases infer that both SWE(t_j) and SWD(#) have significant influence on the final values. Individually, for FO computation, SWD has a very low influence on the result, but its accuracy is still less than those of the rest of the functions.

Case 3: SWD(3)*IFO provides higher accuracy (96.2%) than SWD(2)*IFO (92.8%) with a difference of 3.4%. SWD(3)*OR improves the accuracy (93.3%) when compared to SWD(2)*OR (90.4%) with a difference of 2.9%. This notable variation gives the inference that SWD(#) has more influence when computed with OR and IFO. To be specific, IFO computation performs much better when compared to OR.

Case 4: The classification accuracies of SWD(2)*PMI and SWD(2)*FO are almost equal, but different when used with SWD(3). Their individual computations with two SWD(#) also differ in

negligible accuracy, which provides the inference that the effect of SWD(#) on the result is nullified.

Case 5: Both of the existing methods provide less accuracy than the proposed methods except for SWD(#)*FO. The weight computation of FO cannot achieve better results because of its unbalanced frequency and odds.

Thus, for the Cornell movie reviews, SWD(3) performs better for all weighted functions compared to SWD(2).

To analyze the supremacy of the weighted functions, the data sets are to be selected from entirely different domains. Hence, the Amazon product reviews are obtained for performance analysis (Fig. 3).

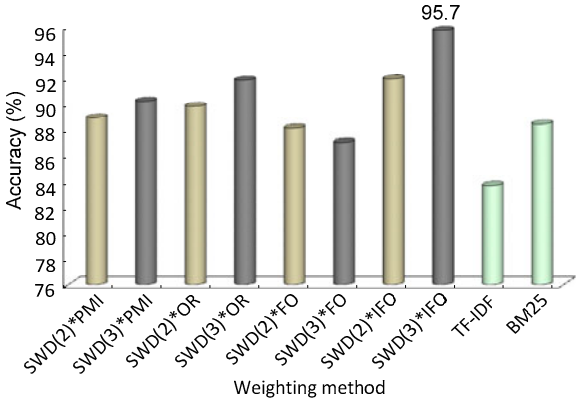

Fig. 3 Weight function performance comparison for Amazon products

Case 6: SWD(3) performs well with all SWE(t_j) (except FO) for Amazon products (Fig. 3). However, there are notable differences with Cornell movie review results. The reasons for the variations are given below:

1. SWD(3) is a normalized word frequency function.

2. Data size is greater when compared to the Cornell movie set.

3. Different domains (i.e., different distributions) have more crisp reviews.

Case 7: The accuracy of SWD(2)*FO for Amazon products is more (difference is 0.3) when compared to that of the Cornell movie set, whereas the accuracy of SWD(3)*FO is less (difference is 0.4). This leads to the inference that SWD(3) computation for FO is controlled to some extent by both the domain and the data set size.

Case 8: PMI, FO, and IFO computations provide similar patterns of results in the following manner. All these three SWE(#) computed with SWD(2) provide less accuracy than with SWD(3) for both domains.

Case 9: TF-IDF fails to provide accuracy on par with those of the proposed methods. In contrast, BM25 performs better than SWD(#)*FO and gives less accuracy than the other proposed methods.

The large Stanford movie set is considered for further analysis since the robustness of the proposed methods needs to be proved. Fig. 4 depicts the outcome of the weight methods on the Stanford movie review.

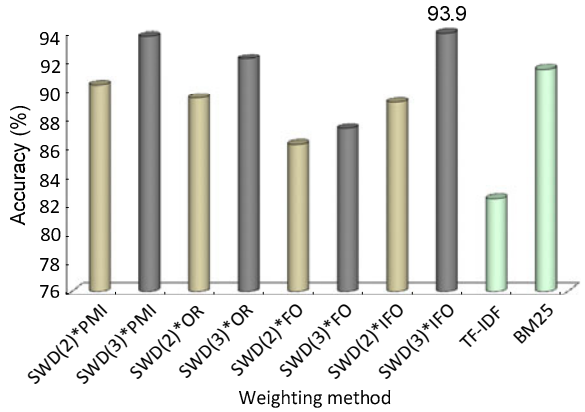

Fig. 4 Weight function performance comparison for the Stanford movie review

Case 10: PMI(93.7%) and IFO(93.9%) computations (with SWD(3)) for the Stanford movie review provide less accuracy when compared with Amazon products and more accuracy when compared with Cornell movie reviews. Thus, the size of the data set and the domain distribution differences both influence the accuracy.

Case 11: Less accuracy of OR(92.1%) in phase II and case 3 infers that OR performs with a short range to some extent (data set size) irrespective of the domain. It is difficult to balance the odds computation for the huge volume of the data set, since the frequency count for each term is high in a large data set. This result also participates as a motivating factor to the proposed IFO.

Case 12: PMI(93.7%) and IFO(93.9%) perform better than BM25(91.4%). IFO and PMI performances are almost equal, reflecting a negligible difference.

Case 13: SWD(#)*FO provides less accuracy than the other proposed methods, showing its non-availability as a balancing factor.

Case 14: The accuracy that TF-IDF provides is less than the lowest accuracy of one of the proposed methods. However, BM25 performs better than SWD(2)*SWE(t_j) and also SWD(3)*FO. The large volume of the data set improves its performance (Esparza *et al*., 2012).

The data set size has a significant influence on the results. In a data set with 4000 reviews, SWD(3)*OR outperforms BM25 and SWD(3)*PMI, whereas in a data set with 25 000 reviews, SWD(3)* PMI performs better than BM25 and SWD(3)*OR.

4.2 Phase II: stop-word inclusion performance analysis

Since frequency based balancing methods can handle stop-words including corpus without any complications and provide significant influence in weighting and classification phases, the corpuses are processed again after the inclusion of stop-words to the bag-of-words. The feature set size (bag-of-words) is 44 000 including stop-words and 43 648 excluding stop-words. The effects of the proposed and existing methods including stop-words are checked through classification, in terms of accuracy (Table 3). The corpus numbers mentioned in Table 3 refer to the corpus names mentioned in Table 2.

Case 15: When each of the accuracies obtained in phase II is compared with the corresponding accuracy in phase I, a surprising fact is noted. The accuracy is improved in phase II and the accuracy differences of all the methods are less than 1.

Case 16: TF-IDF and BM25 can provide more accurate results than in phase I. The fluctuation in terms of bag-of-words size has affected these methods and thus the accuracy is further improved.

From cases 15 and 16, the following facts become clear:

1. The inclusion of stop-words increases the dimension, which enables better categorization. However, performance is based on weighting methods.

2. The increase of 352 words to the bag-of-words does not have much effect when the proposed methods are used for classification, due to their frequency balancing computation.

3. Higher accuracy is achieved by TF-IDF and BM25 during classification. This shows that these methods cannot balance the variations in the bag-of-words. This fact is also proved theoretically (Manning *et al*., 2008).

4. For processing reviews, removal of stop-words is not necessary, and when the removal is avoided, it leads to greater improvement in classification.

5. In the future, the corpuses of opinion mining do not need to be preprocessed by removing the stop-words, if frequency based weighting methods are used. At the same time, they can use the methods proposed in this study without worrying about the variations of stop-words.

4.3 Performance comparison with earlier studies

The results should be verified further by comparison with those of existing studies (Table 4). One of the proposed weighting techniques (SWD(3)*IFO) provides the top accuracy of 97.4% (Cornell movie set), which is better than the accuracy provided by Esparza *et al*. (2012). All of the proposed methods, except SWD(#)*FO, do their best using the Stanford movie review when compared to the existing study which provides 88.89% accuracy (Maas *et al.*, 2011).

Although in some studies the Cornell movie review set with different sizes was used, most studies took 1000 for both training and testing (Table 4). These studies classified the same data set using the popular machine learning algorithms such as SVM, Naive Bayes (NB), *K*-nearest neighbor (KNN), and maximum entropy (ME). SWD(3)*OR and SWD(3)* IFO outperformed existing studies. However, some studies provided better performance than SWD(3)* PMI and SWD(3)*FO.

Table 3 Accuracy comparison of the proposed method with existing methods (including stop-words)

Corpus	SWE(t_j)	Accuracy (%)					
		PMI	OR	FO	IFO	TF-IDF	BM-25
1	SWD(2)	90.2	91.3	88.1	93.4	89.1	90.7
	SWD(3)	91.7	93.9	88.2	97.4		
2	SWD(2)	89.5	90.3	88.8	92.5	87.3	90.9
	SWD(3)	90.9	92.5	87.7	96.3		
3	SWD(2)	90.8	90.2	86.8	89.8	84.6	91.8
	SWD(3)	94.5	92.9	88.1	94.5		

Table 4 Existing studies and their performances on the Cornell movie review sets

Method	Corpus size		Classifier	Accuracy (%)
	Training	Testing		
Existing methods				
Pang *et al*. (2002)	752	1301	SVM	82.90
Salvetti *et al.* (2004)	1500	500	NB	79.50
Pang and Lee (2004)	1000	1000	SVM	86.15
Gabrilovich and Markovitch (2004)	752	1301	SVM	85.40
			KNN	82.70
Boiy *et al*. (2007)	1000	1000	SVM	86.35
			NB	83.95
Tsutsumi *et al.* (2007)	1000	1000	SVM	82.20
			ME	80.50
Zaidan *et al*. (2007)	1000	1000	SVM	92.20
Andreevskaia and Bergler (2008)	5331	5331	NB	81.10
Boiy and Moens (2009)	1000	1000	SVM	85.45
			ME	84.80
Saif *et al*. (2012)	1000	1000	NB	86.30
Proposed methods				
SWD(3)*OR	1000	1000	SVM	93.30
SWD(3)*IFO	1000	1000	SVM	96.20
SWD(3)*FO	1000	1000	SVM	87.60
SWD(3)*PMI	1000	1000	SVM	90.80

SVM: support vector machine; KNN: *K*-nearest neighbor; NB: Naive Bayes; ME: maximum entropy

4.4 Variations of data sets and the classification performance

The following ground facts are set with different weighting combinations, giving a different outcome for the same corpus and same classifier:

Preprocessing: Preprocesses such as stop-word removal, lemmatization and stemming, and data set normalization influence the final outcome. Lemmatization and stemming are done in the proposed study and normalization is taken care of by using weighting methods. The way in which the stop-words change the performances is further discussed.

Tools & parameter settings of the classifier: Most of performance influential factors are based on the tools used or the algorithm implementation. Even similar tools for the same corpus provide different results because of their different parameter settings. The results of our study are obtained using LIBSVM by setting the linear kernel and L2 normalization; parameters C and γ are not set.

Size of corpus: A successful training is always possible with a large number of training data. The proposed study checks the variation of accuracy by linearly increasing the size of the training set. Table 5 shows the improvement achieved.

Table 5 Accuracy improvement for the training sets with different numbers of reviews

Weight method	Accuracy (%)		
	100	250	500
SWD(3)*OR	86.27	89.43	90.87
SWD(3)*IFO	91.30	92.48	94.82
TF-IDF	69.60	73.90	75.82

Training and testing sets: An equal number of positive and negative sets should be used to obtain good classification. This is because an unequal number of training and testing sets will lead to positively or negatively skewed results.

Furthermore, SWD(3)*IFO weighted Cornell movie reviews are classified by replacing the linear kernel with polynomial, RBF, and sigmoid kernels. Their average outcomes from 10 cross validations, including accuracy, the number of iterations, and the number of support vectors, are shown in Table 6.

Table 6 Average performance of classifications for different kernels

Kernel	Accuracy (%)	Number of iterations	Number of support vectors
Polynomial	93.9	2623.3	1576.6
RBF	95.8	1598.1	1112.1
Sigmoid	96.1	1072.5	571.8
Linear	96.2	1305.8	678.3

The linear kernel provides better accuracy compared to the others, and the sigmoid kernel is close to the polynomial and RBF kernels. Based on kernel selection, the margin of classifier width differs and thus the accuracy differs. When the accuracy is focused, the linear kernel can be employed. When also considering the number of iterations and the number of support vectors, select the sigmoid kernel. The sigmoid kernel provides less accuracy than the linear kernel with a difference of 0.1, which will become less, depending on the number of support vectors and the number of iterations. From Table 6, it is fairly visible that the number of support vectors is directly propositional to the number of iterations.

5 Conclusions

In this study we propose a word weighting scheme based on inference through statistical functions. The word weighting combinations are applied to three popularly known and widely used review data sets. The results show that the proposed schemes, especially SWD(3)*IFO, outperform the widely used review data sets based on the inferred weighting and also produce the best accuracy of 97.4% (highest in the Cornell movie reviews). The proposed methods are aimed to reveal that the inferred weighting schemes are better than the schemes that do not consider the association of words and their expressed polarity. In addition to the improvement of classification accuracy, our study reveals that the widely used stop-word removal process is not necessary for opinion mining classification.

The existing inverse document frequency weights the words based on the total documents and the number of documents in which the word occurs. In contrast, in the proposed SWE(t_j) computation, this is done on both positive and negative classes. The opinion conveyed by a word based on both classes is computed separately, and the maximum weighted (significant) value is taken as the SWE(t_j) value of that word. This meaningful computation greatly increases its accuracy.

Readers may be curious about the stop-word inclusion process, which needs to be analyzed further. There are more statistical functions available for mining the data and these can be used in various combinations to obtain the proper word weight.

References

Andreevskaia, A., Bergler, S., 2008. When specialists and generalists work together: overcoming domain dependence in sentiment tagging. Proc. ACL-08, p.290-298.

Armstrong, T.G., Moffat, A., Webber, W., *et al*., 2009. Improvements that don't add up: ad-hoc retrieval results since 1998. Proc. 18th ACM Conf. on Information and Knowledge Management, p.601-610. http://dx.doi.org/10.1145/1645953.1646031

Barnes, S.J., Bohringer, M., 2011. Modeling use continuance behavior in micro blogging services: the case of Twitter. *J. Comput. Inform. Syst.*, **51**(4):1-10.

Blitzer, J., Dredze, M., Pereira, F., 2007. Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. Proc. 45th Annual Meeting of the Association of Computational Linguistics, p.440-447.

Boiy, E., Moens, M.F., 2009. A machine learning approach to sentiment analysis in multilingual web texts. *Inform. Retr.*, **12**(5):526-558. http://dx.doi.org/10.1007/s10791-008-9070-z

Boiy, E., Hens, P., Deschacht, K., *et al*., 2007. Automatic sentiment analysis in on-line text. Proc. 11th Int. Conf. on Electronic Publishing, p.349-360.

Church, K.W., Hanks, P., 1989. Word association norms, mutual information and lexicography. Proc. 27th Annual Meeting on Association for Computational Linguistics, p.76-83. http://dx.doi.org/10.3115/981623.981633

Das, S., Chen, M., 2001. Yahoo! for Amazon: extracting market sentiment from stock message boards. *Manag. Sci.*, **53**(9):1375-1388. http://dx.doi.org/10.1287/mnsc.1070.0704

Debole, F., Sebastiani, F., 2003. Supervised term weighting for automated text categorization. Proc. ACM Symp. on Applied Computing, p.784-788. http://dx.doi.org/10.1145/952532.952688

Esparza, S.G., O'Mahony, M.P., Smyth, B., 2012. Mining the real-time web: a novel approach to product recommendation. *Knowl.-Based Syst.*, **29**(3):3-11. http://dx.doi.org/10.1016/j.knosys.2011.07.007

Gabrilovich, E., Markovitch, S., 2004. Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5. Proc.

21st Int. Conf. on Machine Learning, p.41-50. http://dx.doi.org/10.1145/1015330.1015388

Geng, L., Hamilton, H.J., 2006. Interestingness measures for data mining: a survey. *ACM Comput. Surv.*, **38**(3), Article 9. http://dx.doi.org/10.1145/1132960.1132963

He, B., Huang, J.X.J., Zhou, X., 2011. Modeling term proximity for probabilistic information retrieval models. *Inform. Sci.*, **181**(14):3017-3031. http://dx.doi.org/10.1016/j.ins.2011.03.007

Lee, S., Song, J., Kim, Y., 2010. An empirical comparison of four text mining methods. *J. Comput. Inform. Syst.*, **51**(1):1-10.

Li, S., Xia, R., Zong, C., *et al*., 2009. A framework of feature selection methods for text categorization. Proc. Joint Conf. 47th Annual Meeting of the ACL and Proc. 4th Int. Joint Conf. on Natural Language of the AFNLP, p.692-700.

Maas, A.L., Daly, R.E., Pham, P.T., *et al*., 2011. Learning word vectors for sentiment analysis. Proc. 49th Annual Meeting of the Association for Computational Linguistics, p.142-150.

Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK.

Mladenić, D., Grobelnik, M., 1998. Feature selection for classification based on text hierarchy. Proc. Int. Conf. on Automated Learning and Discovery.

Ng, V., Dasgupta, S., Arifin, S.M.N., 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. Proc. Int. Conf. on COLING/ACL, p.611-618.

Nigam, K., McCallum, A.K., Thrun, S., *et al*., 2000. Text classification from labeled and unlabeled documents using EM. *Mach. Learn.*, **39**(2-3):103-134. http://dx.doi.org/10.1023/A:1007692713085

Paltoglou, G., Thelwall, M., 2010. A study of information retrieval weighting schemes for sentiment analysis. Proc. 48th Annual Meeting of the Association for Computational Linguistics, p.1386-1395. http://dx.doi.org/10.3115/1218955.1218990

Pang, B., Lee, L., 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. Proc. 42nd Annual Meeting of the Association for Computational Linguistics, p.271-278. http://dx.doi.org/10.3115/1218955.1218990

Pang, B., Lee, L., Vaithyanathan, S., 2002. Thumbs up?: sentiment classification using machine learning techniques. Proc. Conf. on Empirical Methods in Natural Language Processing, p.79-86. http://dx.doi.org/10.3115/1118693.1118704

Saif, H., He, Y., Alani, H., 2012. Alleviating data sparsity for Twitter sentiment analysis. CEUR Workshop Proc., p.2-9.

Salton, G., Buckley, C., 1998. Term-weighting approaches in automatic text retrieval. *Inform. Process. Manag.*, **24**(5):513-523. http://dx.doi.org/10.1016/0306-4573(88)90021-0

Salvetti, F., Lewis, S., Reichenbach, C., 2004. Impact of lexical filtering on overall opinion polarity identification. Proc. AAAI Spring Symp. on Exploring Attitude and Affect in Text: Theories and Applications.

Sebastiani, F., 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, **34**(1):1-47. http://dx.doi.org/10.1145/505282.505283

Sheikh, M., Conlon, S., 2012. A rule-based system to extract financial information. *J. Comput. Inform. Syst.*, **52**(4):10-19. http://dx.doi.org/10.1080/08874417.2012.11645572

Simmons, L., Conlon, S., Mukhopadhyay, S., *et al*., 2011. A computer aided content analysis of online reviews. *J. Comput. Inform. Syst.*, **52**(1):43-55. http://dx.doi.org/10.1080/08874417.2011.11645521

Tong, R.M., 2001. An operational system for detecting and tracking opinions in on-line discussion. Working Notes of the ACM SIGIR Workshop on Operational Text Classification, p.1-6.

Tsutsumi, K., Shimada, K.K., Endo, T., 2007. Movie review classification based on a multiple classifier. Proc. Annual Meetings of the 21st Pacific Asia Conf. on Language, Information and Computation, p.481-488.

Xu, Y., Jones, G.J., Li, J., *et al*., 2007. A study on mutual information-based feature selection for text categorization. *J. Comput. Inform. Syst.*, **3**(3):1007-1012. http://dx.doi.org/10.1016/j.eswa.2008.07.062

Zaidan, O., Eisner, J., Piatko, C.D., 2007. Using "annotator rationales" to improve machine learning for text categorization. Proc. HLT-NAACL, p.260-267.