Neurocomputing 167 (2015) 434–442

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

New term weighting schemes with combination of multiple classifiers for sentiment analysis

Mohamed Abdel Fattah [a,b,*]

[a] Department of Computer Sciences, CCSE, Taibah University, Saudi Arabia
[b] Department of Electronics Technology, FIE, Helwan University, Cairo, Egypt

ARTICLE INFO

Article history:
Received 5 November 2014
Received in revised form
22 February 2015
Accepted 16 April 2015
Communicated by Y. Chang
Available online 29 April 2015

Keywords:
Sentiment classification
Openion mining
Term weighting schemes

ABSTRACT

The rapid growth of social media on the Web, such as forum discussions, reviews, blogs, micro-blogs, social networks and Twitter has created huge volume of opinionated data in digital forms. Therefore, last decade showed growth of sentiment analysis task to be one of the most active research areas in natural language processing. In this work, the problem of classifying documents based on overall sentiment is investigated. The main goal of this work is to present comprehensive investigation of different proposed new term weighting schemes for sentiment classification. The proposed new term weighting schemes exploit the class space density based on the class distribution in the whole documents set as well as in the class documents set. The proposed approaches provide positive discrimination on frequent and infrequent terms. We have compared our new term weighting schemes with traditional and state of art term weighting schemes. Some of our proposed terms weighting schemes outperform the traditional and state of art term weighting schemes results.

1. Introduction

Recent years have seen rapid growth in review sites and on-line discussion groups (e.g., the New York Times' Books web page) where a crucial characteristic of the posted articles is their opinion or overall sentiment towards the subject matter—for example, whether a product review is negative or positive [1–3]. Sentiment analysis is exploited to analyze people's opinions, evaluations, sentiments, attitudes, appraisals and emotions towards certain entities as services, products, individuals, organizations, events, topics and issues.

Sentiment classification is considered as a text classification task with two classes (positive and negative classes). Therefore, many existing supervised learning approaches can be exploited for this purpose. For instance, Naïve Bayes, Neural Networks, Hidden Markov Models, Gaussian Mixture Models and support vector Models. Many features and learning algorithms were exploited to train and test the supervised learning approaches such as: terms and their frequency, part-of-speech (POS), sentiment words and phrases, rules of opinions, sentiment shifters and syntactic dependency. Unsupervised Learning approaches can be exploited for sentiment classification task as well.

The dominant approaches in sentiment categorization generally follow traditional topical text categorization approaches [4], where a document is regarded as a bag of words (BOW), mapped into a feature vector, and then classified by machine learning techniques [5–11] such as naive Bayes (NB) [12], maximum entropy (ME) [13], or support vector machines (SVM) [9,14]. The effectiveness of machine learning techniques when applied to sentiment classification tasks is evaluated in the research by Pang et al. [15].

Yu and Hatzivassiloglou used a modified log-likelihood score value to determine the negative or positive orientation for each adverb, adjective, verb and noun [16]. Hu and Liu proposed a lexicon-based algorithm for sentiment orientation of a sentence [17]. The approach is based on a sentiment lexicon created using bootstrapping with given negative and positive sentiment word seeds and the antonyms and synonyms relations in WordNet. Kim and Hovy used a similar approach. However, they calculated the sentiment orientation by multiplying the scores of the sentiment words in the sentence [18]. Kim et al. exploited supervised learning methods. Some approaches used a domain specific lexicon with a shallow natural language processing approach to specify the sentiment orientation of a sentence [19–21]. Anthony and Gamon exploited a semi-supervised learning algorithm which is based on Expectation Maximization (EM) using naive Bayes as a

* Corresponding author at: Department of Electronics Technology, FIE, Helwan University, Cairo, Egypt.
E-mail address: mohafi2003@helwan.edu.eg

http://dx.doi.org/10.1016/j.neucom.2015.04.051
0925-2312/

classifier to learn from a small set of labeled sentences and a large set of unlabeled sentences [22]. Ryan et al. proposed a hierarchical sequence learning algorithm similar to conditional random fields (CRF) to learn sentiment [23,24]. Täckström and McDonald proposed an integrated supervised and a partially supervised model to perform multi-level sentiment classification [25]. Hassan, Qazvinian and Radev proposed a method to specify attitudes about participants in online discussions [26]. Davidov, Tsur and Rappoport proposed a supervised learning approach for sentiment classification of Twitter postings [27]. Zhang et al. proposed a sentiment elicitation approach that uses compositional semantic rule algorithm, bag-of-words with rule-based algorithm and numeric sentiment identification algorithm to train machine learning tool for classifying a tweet [28]. Liu et al. have investigated whether a flip polarity (switch negation) is a reasonable way to quantify negation or not [29]. Flip polarity seems to work well in certain cases and fails in others.

Vector Space Model (VSM) is exploited to represent documents for sentiment analysis task. The weight of each term in a document's vector is the key component of the VSM of document representation that measures the importance of the term in a document. In the indexing process, two features are of main concern: statistical term weighting where term weighting is based on discriminative supremacy of a term that appears in a document or a group of documents and semantic term weighting where term weighting is based on a term's meaning [30].

In information retrieval, complex term weighting approaches that are based on learning term weight by optimization are considered [31]. Although the complex term weighting approaches used in information retrieval task are proved to achieve good accuracy, the work of Paltoglou et al. showed that these approaches only provide information about the general distribution of terms without providing any evidence of class preference [32]. In Salton and Buckley's work, three factors have been considered for term weighting; term frequency, inverse document frequency and normalization [33]. Based on three factors, many approaches of term weighting have been presented for information retrieval. tfidf and normalized tfidf is considered as the best document weighting functions for information retrieval and text categorization tasks [34]. Although tfidf score gives positive discrimination to rare terms and is biased against frequent terms, it ignores the category information in text categorization task. Theeramnukkong and Lertnattee conducted different experiments based on various combinations of inter-class standard deviation, standard deviation with tfidf and class standard deviation [35]. Although the average results were bad compared with tfidf, one of their proposed approaches performed better than tfidf. In [36], idf has been replaced with information gain (X^2) statistic. However, it has not shown a consistent superiority over the standard tfidf. In [37], idf has been replaced with some feature parameter functions in text classification task. This work reported that tf.chi is better than tfidf when it is associated with SVM as a classifier.

A variety of feature selection techniques [46], such as information gain [38], chi-square test, and document frequency [39], have been used to reduce the dimension of the vectors. Soucy and Mineau introduced a new term weighting method (ConfWeight) based on statistical confidence intervals [40]. The experimental results of this work showed that ConfWeight outperformed tfidf when it was applied on three document sets. Lan et al. proposed a supervised term weighting approach, tf.rf, to improve the terms' discriminating power for text classification task [41]. The experimental results show that tf.rf gives better performance than some supervised term weighting approaches including tfidf. Luo et al. replaced the idf function with a semantic weight (sw) using WordNet [42]. The experiment's results showed that the proposed tf.sw scheme gives better results than tfidf scheme. Using semantic term weighting, it is possible to address a limited number of terms in a term index. However, it is difficult to provide the appropriate semantic knowledge of a term based on categories with large number of terms in the term index. Therefore, it is more convenient to exploit statistical term weighting approaches to compute the score of a term.

In this work, term frequency inverse document frequency, term frequency inverse class frequency, term weighting based on: mutual information, odds ratio, weighted log likelihood ratio and X^2 statistic are considered as base line approaches. In this paper, we propose new term weighting schemes for sentiment analysis. The proposed new term weighting schemes exploit the class space density based on the class distribution in the whole document set as well as the class documents set. These proposed approaches provide positive discrimination on frequent and infrequent terms. Support vector machine classifier (SVM), probabilistic neural network (PNN), Gaussian mixture model (GMM) have been exploited to investigate the traditional and proposed term weighting schemes effectiveness. Combination of the previously mentioned classifiers using simple voting and Borda count approaches have been exploited as well.

The paper is organized as follows: Section 2 illustrates the proposed term weighting schemes with combination of multiple classifiers, Section 3 describes the experimental results and Section 4 is the conclusions and future works.

2. The proposed term weighting schemes with combination of multiple classifiers

Before we present our proposed term weighting schemes, we investigate some traditional term weighting schemes to exploit them as base line approaches.

2.1. Traditional term weighting schemes

2.1.1. Term frequency inverse document frequency (tfidf)

Given a set of classes $C=\{c_p$ (positive class), c_n (negative class)$\}$ and a set of training documents $D=\{d_1, d_2,\ldots d_N\}$ where each training document d_i is assigned to one class (positive or negative), sentiment classification is a task to use this given information to find one suitable category for a new document. In a vector space model, a document is represented by a vector based on the weight of each term in the document.

Term frequency (tf) and inverse document frequency (idf) in the form of tfidf is applied in most research works to weight a term in a document. The term frequency in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_j. Thus the term frequency is defined as follows:

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

where $n_{i,j}$ is the number of occurrences of the considered term in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j. The inverse document frequency is a measure of the general importance of the term (obtained by dividing the number of all documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$\mathrm{idf}_i = \log \frac{|D|}{|\{d : t_i \in d\}|} \tag{2}$$

With $|D|$=total number of documents in the corpus and $|\{d : t_i \in d\}|$=number of documents where the term t_i appears (that is $n_{i,j} \neq 0$).

then

$$\text{tfidf}_{i,j} = \text{tf}_{i,j} \times \text{idf}_i \tag{3}$$

Normalized form of $\text{tfidf}_{i,j}$ is given as

$$\text{tfidf}^{norm}_{i,j} = \frac{\text{tfidf}_{i,j}}{\max_{i,j} \text{tfidf}_{i,j}} \tag{4}$$

A high weight in tfidf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms.

2.1.2. Term frequency inverse class frequency (tficf)

The term frequency (tf) is calculated as in formula (1). The inverse class frequency (icf) is calculated as in following equation:

$$\text{icf} = \log \frac{|C|}{|\{c : t_i \in c\}|} \tag{5}$$

With $|C|$=total number of classes. $|\{c : t_i \in c\}|$=number of classes where the term t_i appears in.

Then

$$\text{tficf}_{i,j} = \text{tf}_{i,j} \times \text{icf}_i \tag{6}$$

Normalized form of $\text{tficf}_{i,j}$ is given as

$$\text{tficf}^{norm}_{i,j} = \frac{\text{tficf}_{i,j}}{\max_{i,j} \text{tficf}_{i,j}} \tag{7}$$

A high weight in tficf is reached by a high term frequency (in the given class) and a low document frequency of the term in the whole collection of documents of the other class; the weights hence tend to filter out common terms.

2.1.3. Term weighting based on mutual information (MI)

Deng and Church [43,44] presented a term weighting based on mutual information. Given term t_i and document set D^c (set of documents in a certain class; positive or negative), the mutual information between them is defined as

$$\text{MI}(t_i, D^c) = \log \frac{P(t_i, D^c)}{P(t_i) \times P(D^c)} \tag{8}$$

The term weighting based on mutual information may be estimated as follows:

$$\text{MI}(t_i, D^c) \approx \log \frac{\frac{n_c(t_i)}{|D|}}{\frac{n_p(t_i)+n_n(t_i)}{|D|} \times \frac{n_c}{|D|}} = \log \frac{n_c(t_i) \times |D|}{[n_p(t_i)+n_n(t_i)] \times n_c} \tag{9}$$

With $|D|$=total number of documents in the training corpus, $n_c(t_i)$ is the number of documents that contain the term (t_i) and belong to class (c), $n_p(t_i)$ is the number of documents that contain the term (t_i) and belong to positive class, $n_n(t_i)$ is the number of documents that contain the term (t_i) and belong to negative class and n_c is the total number of documents in class (c).

Then the term weighting based on mutual information is given as follows:

$$\text{TW}_{MI}(t_i) = \max\{\text{MI}(t_i, D^p), \text{MI}(t_i, D^n)\} \tag{10}$$

2.1.4. Term weighting based on odds ratio (OR)

Odds ratio is used in information retrieval [43,44,45]. In text classification, occurrence of different words as features is used to classify documents. Given a term (t_i) and a set of positive or negative documents D^c, the odds ratio is given as follows:

$$\text{OR}(t_i, D^c) = \log \frac{P(t_i|D^c)(1-P(t_i|\overline{D^c}))}{(1-P(t_i|D^c))P(t_i|\overline{D^c})} \tag{11}$$

$$\text{OR}(t_i, D^c) \approx \log \frac{n_c(t_i) \times (|D| - n_c - n_{\overline{c}}(t_i))}{(n_c - n_c(t_i)) \times n_{\overline{c}}(t_i)} \tag{12}$$

With $n_{\overline{c}}(t_i)$ is the number of documents that contain the term (t_i) and not belong to class (c).

Then the term weighting based on odds ratio is given as follows:

$$\text{TW}_{OR}(t_i) = \max\{\text{OR}(t_i, D^p), \text{OR}(t_i, D^n)\} \tag{13}$$

2.1.5. Term weighting based on weighted log likelihood ratio (WLLR)

Weighted log likelihood ratio (WLLR) is proved to be effective for text categorization [43,47]. Given a term (t_i) and a set of positive or negative documents D^c, the WLLR is given as follows:

$$\text{WLLR}(t_i, D^c) = P(t_i|D^c) \log \frac{P(t_i|D^c)}{P(t_i|\overline{D^c})} \tag{14}$$

$$\text{WLLR}(t_i, D^c) \approx \frac{n_c(t_i)}{n_c} \log \frac{n_c(t_i) \times (|D| - n_c)}{n_{\overline{c}}(t_i) n_c} \tag{15}$$

Then the term weighting based on weighted log likelihood ratio is given as follows:

$$\text{TW}_{WLLR}(t_i) = \max\{\text{WLLR}(t_i, D^p), \text{WLLR}(t_i, D^n)\} \tag{16}$$

2.1.6. Term weighting based on X^2 statistic (CHI)

CHI; X^2 statistic measures the lack of independence between two random variables (t_i and D^c) [43,48]. The CHI between the term t_i and document set D^c is calculated as follows:

$$\text{CHI}(t_i, D^c) = \frac{|D| \times [(n_c(t_i) \times n_{\overline{c}}(\overline{t_i}) - n_{\overline{c}}(t_i) \times n_c(\overline{t_i})]^2}{[n_c(t_i)+n_c(\overline{t_i})] \times [n_{\overline{c}}(t_i)+n_{\overline{c}}(\overline{t_i})] \times [n_c(t_i)+n_{\overline{c}}(t_i)] \times [n_c(\overline{t_i})+n_{\overline{c}}(\overline{t_i})]} \tag{17}$$

With $n_{\overline{c}}(\overline{t_i})$ is the number of documents that neither contain term (t_i) nor belong to class (c). $n_c(\overline{t_i})$ is the number of documents that belong to class (c) but do not contain term (t_i).

Then the term weighting based on CHI statistic is given as follows:

$$\text{TW}_{CHI}(t_i) = \max\{\text{CHI}(t_i, D^p), \text{CHI}(t_i, D^n)\} \tag{18}$$

2.2. The proposed term weighting schemes

In the traditional tfidf approach that is used in information retrieval, a group of similar documents is linked with a certain term and the idf function provides the lowest score for this term. However, in the classification task, it is not clear that the idf function provides the lowest score for the terms that are similar. The similar documents that are linked with a certain term may be belonging to a certain category. Therefore, it is of some interest to explore the occurrence characteristics of terms in both class space and document space. To create more informative terms, we propose to weight a term based on contribution of document-based approach and class based approach. Then, the proposed term weighting schemes give positive discrimination for frequent and infrequent terms. In the following proposed term weighting schemes, the distribution of a certain term in the documents, classes and a specific class is investigated.

2.2.1. Term weighting based on class density (CD) relative to all class documents (CD_{all_c})

To measure the importance of a certain term in a certain class relative to all class documents, the following formula could be used:

$$CD_{all\_c}(t_i) = \frac{n_c(t_i)}{D(t_i)} \tag{19}$$

With $CD_{all\_c}(t_i)$ is class density of the term (t_i) relative to all class documents that contain this term. $D(t_i)$ is the number of documents that contain term (t_i) in all classes.

Then the term weighting based on the class density of a term is calculated as follows:

$$TW_{CD\_all\_c}(t_i) = \underset{c}{\operatorname{argmax}}\, CD_{all\_c}(t_i) \tag{20}$$

Normalized form of $TW_{CD\_all\_c}(t_i)$ is given as:

$$TW_{CD\_all\_c}(t_i)^{norm} = \frac{TW_{CD\_all\_c}(t_i)}{\max_i TW_{CD\_all\_c}(t_i)} \tag{21}$$

Formula (21) reflects the distribution of a term t_i in a certain class relative to its distribution in the whole documents.

2.2.2. Term weighting based on class density (CD) relative to all documents in the same class (CD_c)

To measure the importance of a certain term in a certain class relative to all documents in this class, the following formula could be used:

$$CD_c(t_i) = \frac{n_c(t_i)}{nc} \tag{22}$$

Then the term weighting based on the class density of a term relative to all documents in that class is calculated as follows:

$$TW_{CD\_c}(t_i) = \underset{c}{\operatorname{argmax}}\, CD\_c(t_i) \tag{23}$$

Normalized form of $TW_{CD\_c}(t_i)$ is given as:

$$TW_{CD\_c}(t_i)^{norm} = \frac{TW_{CD\_c}(t_i)}{\max_i TW_{CD\_c}(t_i)} \tag{24}$$

Formula (24) reflects the distribution of a term t_i in a certain class relative to the whole documents in this class.

2.2.3. Term weighting based on logarithmic ratio between $CD_p(t_i)$ and $CD_n(t_i)$; (log_CD)

The importance of a certain term in a positive class relative to all documents in this class is:

$$CD_p(t_i) = \frac{n_p(t_i)}{n_p} \tag{25}$$

With $n_p(t_i)$ is the number of documents that contain the term (t_i) and belong to positive class, and n_p is the total number of documents in positive class.

The importance of a certain term in a negative class relative to all documents in this class is:

$$CD_n(t_i) = \frac{n_n(t_i)}{n_n} \tag{26}$$

With $n_n(t_i)$ is the number of documents that contain the term (t_i) and belong to negative class, and n_n is the total number of documents in negative class.

Then the term weighting based on the logarithmic ratio between $CD_p(t_i)$ and $CD_n(t_i)$ is calculated as follows:

$$TW_{\log\_CD}(t_i) = |\log \frac{CD_p(t_i)}{CD_n(t_i)}| \tag{27}$$

Normalized form of $TW_{\log\_CD}(t_i)$ is given as:

$$TW_{\log\_CD}(t_i)^{norm} = \frac{TW_{\log\_CD}(t_i)}{\max_i TW_{\log\_CD}(t_i)} \tag{28}$$

Formula (28) reflects the distribution of a term t_i in a certain class relative to its distribution in the other class.

2.2.4. Term weighting based on max weight over all previous term weighting schemes ($max_{all\ schemes}$)

For the previously mentioned traditional term weighting schemes, normalization took place like the proposed term weighting schemes. Max term weight is selected over all previously mentioned term weighting schemes (traditional and proposed term weighting schemes) for this approach. The max term weight is specified as follows:

$$TW_{max}(t_i) = \max_{allschemes} TW(t_i) \tag{29}$$

2.3. Combination of multiple classifiers

Before presenting combination of multiple classifiers, investigation of each classifier is presented.

2.3.1. Support Vector Machine Classifier (SVM)

Support Vector Machine (SVM) as a classification method has often been found to provide excellent classification results [9,8]. The SVM approach tries to find the optimal separating hyperplane between classes.

The Sigmoid kernel function is used to implement SVM. It is given as follows:

$$K(x_i, x_j) = \tan h(\gamma \cdot x_i^T x_j + r) \tag{30}$$

where $\gamma$and r are the kernel parameters. γ is given the value (1) and r is given the value (-100).

2.3.2. Probabilistic Neural Network (PNN)

Probabilistic Neural Networks are a versatile and efficient tool to classify high-dimensional data [6]. Bayes' theorem can be used to perform probabilistic neural network optimal classification as follows:

The probability distribution function (PDF) for a feature vector (X) to be of a certain category (positive class or negative class) is given by

$$f_a(X) = 1/(2\pi)^{p/2}\sigma^p(1/n_a)\sum_{i=1}^{n_a}\exp(-(X-Y_{ai})^\tau(X-Y_{ai})/2\sigma^2) \tag{31}$$

Where

$f_a(X)$=the value of the PDF for class A at point X.
X=test vector to be classified.
i=training vector number.
p=the training vector size.
n_a=number of training vectors in class A.
$Y_{ai}=i^{th}$ training vector for class A.
τ=transpose
σ=the standard deviation of the Gaussian curves used to construct the PDF.

Consider (n_a/n_{total}) to represent the relative number of trials in each category. Therefore $(1/n_a)$term is canceled out as follows:

$$f_a(X) = 1/(2\pi)^{p/2}\sigma^p(1/n_{total})\sum_{i=1}^{n_a}\exp(-(X-Y_{ai})^\tau(X-Y_{ai})/2\sigma^2) \tag{32}$$

Table 1
The sentiment classification accuracy using support vector machine based on (tfidf, tficf, MI, OR, WLLR, CHI, CD_{all_c}, $D_{\_c}$, log_CD, $max_{all\ schemes}$).

Term weighting scheme	tfidf (%)	tficf (%)	MI (%)	OR (%)	WLLR (%)	CHI (%)	CD_{all_c} (%)	$CD_{\_c}$ (%)	log_CD (%)	$max_{all\ schemes}$ (%)
Movie reviews	85.7	83.8%	87.6	88.3	86.2	85.4	91.2	90.1	90.8	89.3
Books	82.8	80.2	84.7	85.2	83.4	82.2	88.3	87.4	88.1	86.8
Dvd	83.7	82.3	85.5	85.9	84.1	83.3	89.2	88.3	88.9	87.5
Electronics	84.9	83.1	86.8	87.3	85.4	84.5	90.6	89.7	90.4	89.1
Kitchen	86.4	85.4	88.1	88.5	86.3	85.5	91.5	90.4	91.1	89.8

Terms common to all classes such as $1/(2\pi)^{p/2}$,σ^p and n_{total} could also be eliminated, leaving the following formula:

$$f_a(X)\alpha \sum_{i=1}^{n_a} \exp(-(X-Y_{ai})^{\tau}(X-Y_{ai})/2\sigma^2) \tag{33}$$

Hence the classifier may be expressed as follows:

For a feature parameter X to belong to a category (r); the following formula could be verified:

$$\sum_i \exp(-(X-Y_{ri})^{\tau}(X-Y_{ri})/2\sigma^2) \geq \sum_i \exp(-(X-Y_{si})^{\tau}(X-Y_{si})/2\sigma^2) \tag{34}$$

With (s) represents the other category.

The expression $(X-Y_{ri})^{\tau}(X-Y_{ri})=(X^{\tau}X)-(2X^{\tau}Y_{ri})+(Y_{ri}{}^{\tau}Y_{ri})=1-(2X^{\tau}Y_{ri})+1=2-(2X^{\tau}Y_{ri})$ allowing formula (34) to be simplified as follows:

$$\sum_i \exp((X^{\tau}Y_{ri}-1)/\sigma^2) \geq \sum_i \exp((X^{\tau}Y_{si}-1)/\sigma^2) \tag{35}$$

2.3.3. Gaussian mixture model (GMM)

The use of Gaussian mixture models as a classification tool is motivated by the interpretation that the Gaussian components represent some general output dependent features and the capability of Gaussian mixtures to model arbitrary densities [6,7,26,30].

The probability density function for a certain class feature vector X is a weighted sum, or mixture, of k class-conditional Gaussian distributions. For a given class model λc, the probability of observing X is given by

$$p(X|\lambda c) = \sum_{k=1}^{K} w_{c,k} N(X; \vec{\mu}_{c,k}, \Sigma_{c,k}) \tag{36}$$

Where $w_{c,k}$, $\vec{\mu}_{c,k}, \Sigma_{c,k}$are the mixture weight, mean, and covariance matrix, respectively, for the i-th component, which has a Gaussian distribution given by

$$N(X; \vec{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\sum|}} \exp(-\frac{1}{2}(X-\vec{\mu})^{\tau}\Sigma^{-1}(X-\vec{\mu})) \tag{37}$$

Where n is the dimension of X. We used Σ as diagonal covariance matrices. Given a set of training vectors of a certain class, an initial set of means is estimated using k-means clustering. The mixture weights, means, and covariances are then iteratively trained using the expectation maximization (EM) algorithm.

Using this approach, we constructed a class-dependent model for each category. After that we used all models for the classification task using the maximum likelihood of each category as follows:

For a given set of class-dependent reference models $(\lambda 1, \lambda 2)$ and one feature vector sequence$X = \{x_1, x_2, \ldots\ldots\ldots, x_n\}$, the minimum error Bays' decision rule is

$$\arg\max_{1 \leq l \leq 2} p(\lambda l|X) = \arg\max_{1 \leq l \leq 2} \frac{p(X|\lambda l)}{p(X)} p(\lambda l) \tag{38}$$

Using formula (38), a feature vector sequence X may be classified as one of the two classes (positive or negative).

2.3.4. Combination of multiple classifiers using simple voting approach

In simple voting approach [49], a test document is classified as positive or negative based on the majority of individual classifier results. If two of the previously mentioned classifiers (SVM, PNN, GMM), classify one document as class A and the third classifier classifies the same document as class B, then the specified classification for the document under consideration is class A.

2.3.5. Combination of multiple classifiers using Borda count approach

In Borda count [49], for every class, addition of the ranks in the n-best lists of each classifier with the first entry in the n-best list is accomplished. That means, the most likely class label, contributing the highest rank number and the last entry having the lowest rank number. Hence, the final output label for a given test pattern X is the class with highest overall rank sum. Therefore, the following formula may be exploited:

$$r_i = \sum_{j=1}^{N} r_i^j \tag{39}$$

With N be the number of classifiers (3), r_i^j is the rank of class i in the n-best list of the j-th classifier. Hence, the test pattern X is assigned the class i with the maximum overall rank count r_i.

3. Experimental results

We have selected two data sets for the experiments; movie reviews [50] and multi-domain sentiment dataset [51]. Movie reviews domain is experimentally convenient because there are large on-line collections of such reviews, and because reviewers usually summarize their overall sentiment with a machine-extractable rating indicator, such as a number of stars; therefore, there is no need to manually-label the data for supervised training or evaluation purposes. The movie reviews data contains 1000 negative and 1000 positive documents. Multi-domain sentiment dataset contains product reviews taken from Amazon.com from many product types (domains). Some domains have hundreds of thousands of review and others much less. In this work, we have used the reviews from four different domains (books, dvd, electronics and kitchen). We have considered 2000 reviews (1000 positive reviews and 1000 negative reviews) from each domain.

We have employed 10-fold cross validation methodology in which the movie reviews dataset and each domain of the multi-domain sentiment dataset is randomly divided into 10-folds. Each turn, one data fold is exploited for testing and the remaining folds are exploited for training. Stop words are removed and light stemmer is applied as preprocessing steps to prepare the data sets. Reviews texts sometimes contain some orthographic mistakes, abbreviations, colloquial expressions, idiomatic expressions or ironic sentences. These bad portions of text could be filtered out (as a preprocessing step) using text summarization. Hence, the methodology of [2] was used to summarize the datasets under consideration. 60% compression rate was used.

Table 2
The sentiment classification accuracy using probabilistic neural network based on (tfidf, tficf, MI, OR, WLLR, CHI, CD_{all_c}, $D_{\_c}$, log_CD, $max_{all\ schemes}$).

Term weighting scheme	tfidf (%)	tficf (%)	MI (%)	OR (%)	WLLR (%)	CHI (%)	CD_{all_c} (%)	$CD_{\_c}$ (%)	log_CD (%)	$max_{all\ schemes}$ (%)
Movie reviews	84.5	82.6	86.8	88.5	86.1	84.9	90.8	90.2	90.3	89.4
Books	81.7	79.6	83.7	85.5	83.3	82.1	88.2	87.7	87.8	86.7
Dvd	81.6	79.4	82.5	84.3	82.1	80.9	86.6	86.1	86.3	85.5
Electronics	83.7	81.5	83.7	85.6	83.4	82.3	88.6	87.9	88.2	87.1
Kitchen	85.2	83.3	85.5	87.3	85.2	84.1	90.9	90.3	90.5	89.2

Table 3
The sentiment classification accuracy using Gaussian mixture model based on (tfidf, tficf, MI, OR, WLLR, CHI, CD_{all_c}, $CD_{\_c}$, log_CD, $max_{all\ schemes}$).

Term weighting scheme	tfidf (%)	tficf (%)	MI (%)	OR (%)	WLLR (%)	CHI (%)	CD_{all_c} (%)	$CD_{\_c}$(%)	log_CD (%)	$max_{all\ schemes}$ (%)
Movie reviews	86.1	84.2	87.8	88.7	85.7	85.8	91.6	90.5	90.7	89.8
Books	83.2	81.3	84.9	85.8	82.7	82.8	88.5	87.4	87.6	86.7
Dvd	84.1	82.1	85.7	86.8	83.5	83.7	89.4	88.2	88.5	87.3
Electronics	85.3	83.4	86.8	87.9	84.8	84.7	90.6	89.3	89.4	88.2
Kitchen	86.8	84.5	88.2	89.3	86.1	86.3	92.1	91.0	91.3	90.4

Table 4
The sentiment classification accuracy using combination of multiple classifiers with simple voting approach based on (tfidf, tficf, MI, OR, WLLR, CHI, CD_{all_c}, $CD_{\_c}$, log_CD, $max_{all\ schemes}$).

Term weighting scheme	tfidf (%)	tficf (%)	MI (%)	OR (%)	WLLR (%)	CHI (%)	CD_{all_c} (%)	$CD_{\_c}$ (%)	log_CD (%)	$max_{all\ schemes}$ (%)
Movie reviews	86.4	84.6	88.4	89.3	85.9	85.6	92.2	91.3	91.2	90.5
Books	83.4	81.6	85.7	86.6	83.1	82.8	89.4	88.5	88.3	87.7
Dvd	84.5	82.7	86.5	87.4	84.1	83.9	90.5	89.6	89.3	88.8
Electronics	85.6	83.8	87.6	88.5	85.0	84.6	91.3	90.5	90.4	89.7
Kitchen	87.1	85.3	89.1	90.2	86.8	86.2	92.7	91.9	91.7	91.1

Table 5
The sentiment classification accuracy using combination of multiple classifiers with Borda count approach based on (tfidf, tficf, MI, OR, WLLR, CHI, CD_{all_c}, $CD_{\_c}$, log_CD, $max_{all\ schemes}$).

Term weighting scheme	tfidf (%)	tficf (%)	MI (%)	OR (%)	WLLR (%)	CHI(%)	CD_{all_c} (%)	$CD_{\_c}$ (%)	log_CD(%)	$max_{all\ schemes}$ (%)
Movie reviews	86.3	84.8	88.8	89.6	86.1	85.9	92.3	91.5	91.2	90.7
Books	83.2	81.7	85.7	86.5	83.1	83.3	89.8	89.1	88.6	88.0
Dvd	84.4	83.0	87.1	88.0	84.6	84.9	91.4	90.7	90.4	89.8
Electronics	85.3	83.8	87.6	88.4	85.1	85.4	91.9	91.1	90.8	90.2
Kitchen	87.2	85.7	89.5	90.3	86.9	87.2	93.7	92.8	92.5	91.9

Here classification accuracy is considered for evaluation. Different measures, such as F measure and precision-recall graphs have been used in the literature. However, since our goal in sentiment classification is to achieve high separation between the two classes on a test set and low misclassification rates, it is thought that accuracy is the best measure of performance. The accuracy is given by the following formula:

$$\text{Accuracy} = \frac{\text{number of true positive} + \text{number of true negative}}{\text{number of true positive} + \text{number of true negative} + \text{number of false positive} + \text{number of false negative}} \quad (40)$$

We have exploited the 10 normalized term weighting schemes; tfidf (formula (4)), tficf (formula 7), MI (formula 10), OR (formula 13), WLLR (formula 16), CHI (formula 18), CD_{all_c} (formula 21), $CD_{\_c}$ (formula 24), log_CD (formula 28), $max_{all\ schemes}$ (formula 29) with SVM (Table 1), PNN (Table 2), GMM (Table 3), simple voting approach (Table 4), Borda count approach (Table 5). Tables 1 to 5 show the results.

It is clear from Tables 1–5 that the proposed term weighting approaches results outperform traditional term weighting schemes using SVM, PNN, GMM and the combination of multiple classifiers. The results of GMM are the best over SVM and PNN. The results of Borda count combination approach are better than that of simple voting combination approach. It is also clear from the tables that the term weighting based on maximum weight over all previous term weighting schemes does not show promising performance compared with the rest proposed term weighting schemes. That might be because of the contributions of some traditional term weighting schemes such as tficf and WLLR have bad effects on the total performance. In the future work we may exploit a certain optimization approach such as genetic algorithm to get the optimum combination of all term weighting schemes. That might improve the total performance. The results associated with the term weighting based on class density (CD) relative to all class documents (CD_{all_c}) are slightly better than other proposed term weighting schemes' results. That reflects the effectiveness of

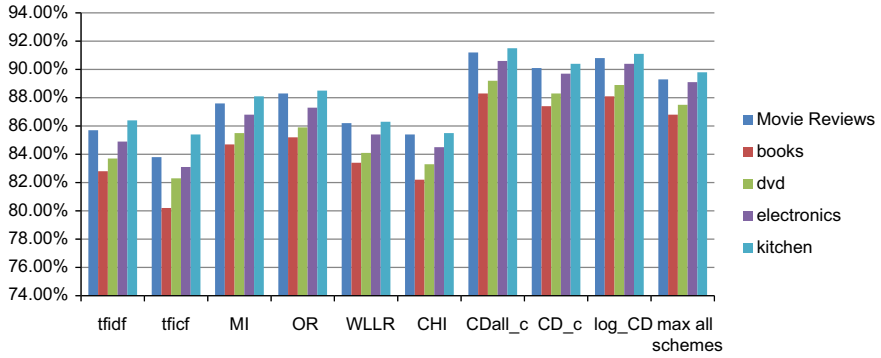

Fig. 1. The sentiment classification accuracy using support vector machine based on all term weighting schemes.

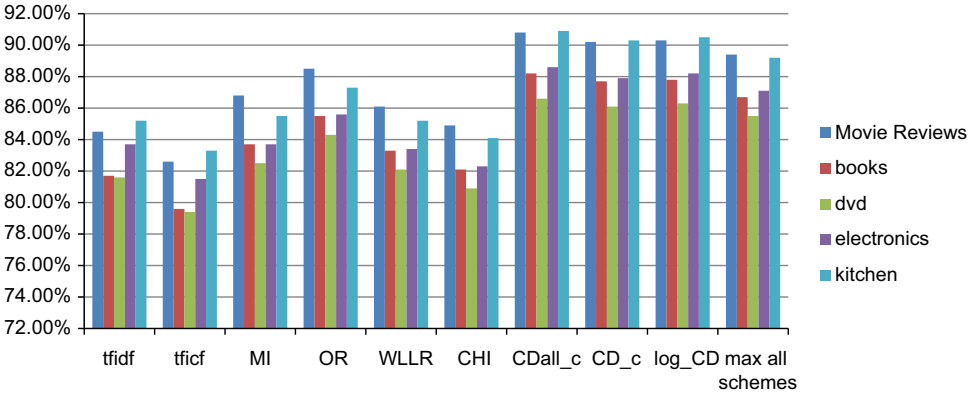

Fig. 2. The sentiment classification accuracy using probabilistic neural network based on all term weighting schemes.

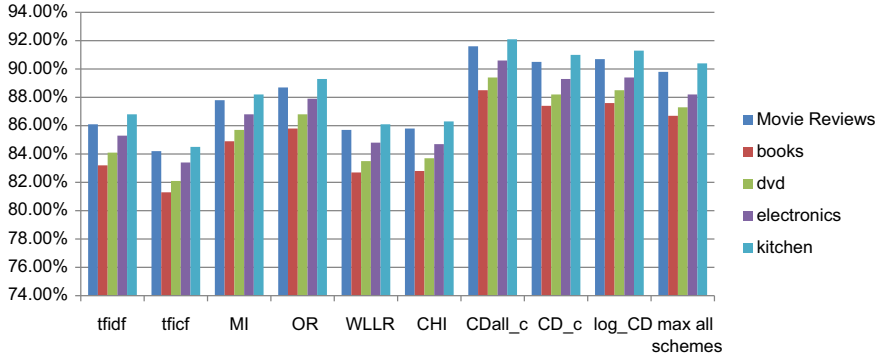

Fig. 3. The sentiment classification accuracy using Gaussian mixture model based on all term weighting schemes.

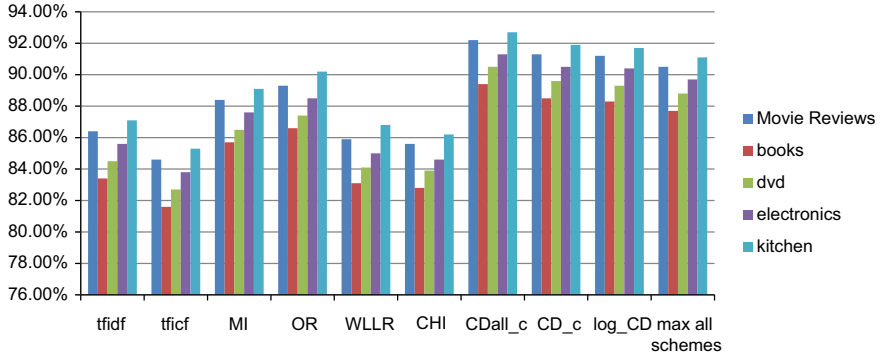

Fig. 4. The sentiment classification accuracy using combination of multiple classifiers with simple voting approach based on all term weighting schemes.

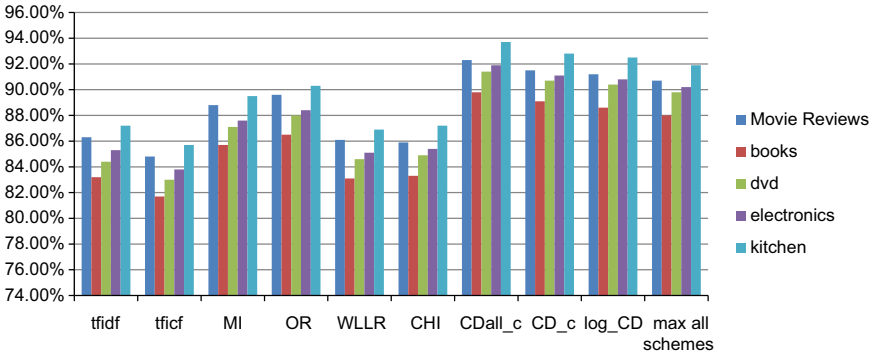

Fig. 5. The sentiment classification accuracy using combination of multiple classifiers with Borda count approach based on all term weighting schemes.

the distribution of a term t_i in a certain class relative to its distribution in the whole documents. Like other approaches, the accuracy depends on the used data set. For instance, "book and dvd" data sets provide slightly lower accuracies.

Figs. 1–5 summarize the sentiment classification accuracies for all approaches with all term weighting schemes for all datasets.

4. Conclusions and future work

In this paper, different approaches for automatic sentiment classification have been investigated based on different proposed new term weighting schemes. We have proposed new term weighting schemes that exploit the class space density based on the class distribution in the whole document set as well as the class documents set. The proposed approaches provided positive discrimination on frequent and infrequent terms. Some of our proposed term weighting schemes outperformed the traditional and state of art term weighting schemes results.

In the future work, some other new term weighting schemes will be created to improve automatic sentiment classification. A certain optimization approach such as genetic algorithm could be exploited to get the optimum combination of all term weighting schemes.

Acknowledgment

This work is supported by the Deanship of Scientific Research, Taibah University, Saudi Arabia.

References

[1] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79–86.
[2] M. Fattah, Sentiment classification using text summarization, in: Proceedings of Arabic Language Technology International Conference (ALTIC), Alexandria, Egypt, 9–10 October 2011, pp. 59–64.
[3] M. Fattah, GMM & FFNN based models for sentiment classification, in: Proceedings of the 1st Taibah University International Conference on Computing and Information Technology, Saudi Arabia, 12–14 March 2012, pp. 45–49.
[4] R. Xia, C. Zong, S. Li, Ensemble of feature sets and classification algorithms for sentiment classification, Inf. Sci. 181 (2011) 1138–1152.
[5] M. Fattah, F. Ren, S. Kuroiwa, Sentence alignment using feed forward neural network, Int. J. Neural Syst. 16 (6) (2006) 423–434.
[6] M. Fattah, F. Ren, S. Kuroiwa, Sentence alignment using P-NNT and GMM, Comput. Speech Lang. 21 (4) (2007) 594–608.
[7] M. Fattah, F. Ren, GA, MR, FFNN, PNN & GMM based models for automatic text summarization, Comput. Speech Lang. 23 (1) (2009) 126–144.
[8] N. Begum, M. Fattah, F. Ren, Automatic text summarization using support vector machine, Int. J. Innov. Comput. Inf. Control 5 (7) (2009) 1987–1996.
[9] M. Fattah, The use of MSVM and HMM for sentence alignment, J. Inf. Process. Syst. 8 (2) (2012) 707–720.
[10] M. Fattah, F. Ren, S.h. Kuroiwa, Effects of phoneme type and frequency on distributed speaker identification and verification, IEICE Trans. Inf. Syst. E89-D (5) (2006) 1712–1719.
[11] M. Fattah, F. Ren, S.h. Kuroiwa, Sentence alignment using hybrid model, in: Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'2005), 2005, pp. 388–392.
[12] D. Lewis, Naive (Bayes) at forty: the independence assumption in information retrieval, Lect. Note Comput. Sci. 1398 (1998) 4–18.
[13] K. Nigam, J. Lafferty, A. McCallum, Using maximum entropy for text classification, in: Proceedings of IJCAI Workshop on Machine Learning for Information Filtering, 1999, pp. 61–67.
[14] M. Joshi, C. Penstein-Rosé, Generalizing dependency features for opinion mining, in: Proceedings of the 47th ACL and the 4th IJCNLP Conference, Association for Computational Linguistics, 2009, pp. 313–316.
[15] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79–86.
[16] Y. Hong and V. Hatzivassiloglou, Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, in: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2003), 2003.
[17] H. Minqing and B. Liu, Mining and summarizing customer reviews, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004). 2004.
[18] Soo-Min Kim and E. Hovy, Determining the sentiment of opinions, in: Proceedings of Interntional Conference on Computational Linguistics (COLING-2004), 2004.
[19] Soo-Min Kim and E. Hovy, Automatic identification of pro and con reasons in online reviews, in: Proceedings of COLING/ACL 2006 Main Conference Poster Sessions (ACL-2006), 2006.
[20] Soo-Min Kim and E. Hovy, Crystal: analyzing predictive opinions on the web, in: Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL-2007), 2007.
[21] A. Syed, M. Aslam, A. Martinez-Enriquez, Lexicon based sentiment analysis of urdu text using sentiunits, advances in artificial intelligence, Lect. Notes Comput. Sci. 6437 (2010) 32–43.
[22] A. Anthony and M. Gamon, Customizing sentiment classifiers to new domains: a case study, in Proceedings of Recent Advances in Natural Language Processing (RANLP-2005), 2005.
[23] M. Ryan, K. Hannan, T. Neylon, Mike Wells,Jeff Reynar, Structured models for fine-to-coarse sentiment analysis, in Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2007), 2007.
[24] L. John, A. McCallum, and F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in Proceedings of International Conference on Machine Learning (ICML-2001), 2001.
[25] Oscar Täckström, R. McDonald, Discovering fine-grained sentiment with latent variable structured prediction models, In Proceedings of the 33rd European ConferenceonInformation Retrieval (ECIR2011), Dublin, Ireland, 2011.
[26] Ahmed Hassan, V. Qazvinian, and D. Radev, What's with the attitude? Identifying sentences with attitude in online discussions, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP-2010), 2010.
[27] Dmitry Davidov, O. Tsur, and A. Rappoport, Enhanced sentiment learning using twitter hashtags and smileys, in: Proceedings of Coling-2010, 2010.
[28] K. Zhang, Y. Cheng, Y. Xie, D. Honbo A. Agrawal, D. Palsetia, K. Lee, W. Liao, and A. Choudhary, SES: sentiment elicitation system for social media data, in: Proceedings of the 11th IEEE International Conference on Data Mining Workshops, 2011.
[29] J. Liu and S. Seneff, Review sentiment scoring via a parse-and-paraphrase paradigm, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 2009, pp. 161–169.
[30] F. Figueiredo, L. Rocha, T. Counto, T. Salles, M.A. Goncalves, Word co-occurrence features for text classification, Inf. Syst. 36 (2001) 843–858.

[31] A.L. Mass, R.E. Daly, P.T. Pham, D. Huang., A.Y. NG, & C. Potts, Learning word vectors for sentiment analysis. in: Proceedings of the 49th annual meeting of the association for computational linguistics (ACL 2011), Portland, Oregon, USA, 2011, pp. 142–150.
[32] G. Paltoglou, M. Thelwall, A study of information retrieval weighting schemes for sentiment analysis, in: Proceedings of ACL 2010, 2010, pp. 1386–1395.
[33] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manag. 24 (5) (1998) 513–523.
[34] M. Fattah, GMM & FFNN based models for sentiment classification, in: Proceedings of the 1st Taibah University International Conference on Computing and Information Technology, Saudi Arabia, 12–14 2012, pp. 45–49.
[35] T. Theeramnukkong, V. Lertnattee, Effect of term distributions on centroid-based text categorization, Inf. Sci. 158 (2004) 89–115.
[36] F. Debole, & F. Sebastiani, Supervised term weighting for automated text categorization, in: Proceeding of the 2003 ACM symposium on applied computing (SAC 2003), NY, USA, 2003, pp. 784–788.
[37] Z. Deng, S. Tang, D. Yang, M. Zhang, L. Li, & K. Xie, A comparative study on feature weight in text categorization, in: Proceedings of the sixth Asia-Pacific web conference (APWeb 2004), Hangzhou, China, 2004, pp. 588–597.
[38] C. Lee, G.G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, Inf. Process. Manag. 42 (1) (2006) 155–165.
[39] Y. Yang, J.O. Pedersen, A Comparative study on feature selection in text categorization, in: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, TN, 1997, pp. 412–420.
[40] P. Soucy. & G.W. Mineau, Beyond tfidf weighting for text categorization in the vector space model, in: Proceedings of the nineteenth international joint conference on artificial intelligence (IJCAI 2005), Edinburgh, Scotland, UK, 2005, pp. 1130–1135.
[41] M. Lan, C.L. Tan, J. Su, Y. Lu, Supervised and traditional term weighting methods for automatic text categorization, IEEE Trans. Pattern Anal. Mach. Intell. 31 (4) (2009) 721–735.
[42] Q. Luo, E. Chen, H. Xiong, A semantic term weighting scheme for text classification, Expert Syst. Appl. 36 (2011) 12708–12716.
[43] Z. Deng, K. Luo, H. Yu, A study of supervised term weighting scheme for sentiment analysis, Expert Syst. Appl. 41 (2014) 3506–3513.
[44] K.W. Church, & P. Hanks, Word association norms, mutual information and lexicography, in: Proceeding of ACL 27, Vancouver, Canada, 1989, pp. 76–83.
[45] D. Mladenic, M. Grobelnik, Feature selection for classification based on text hierarchy, in: Proceeding of conference on automated learning and discovery (CONALD 1998), 1998.
[46] C.J. Van Rijsbergen, D.J. Harper, M.F. Porter, The selection of good search terms, Inf. Process. Manag. 17 (1981) 77–91.
[47] V. Ng, S. Dasgupta, & S. M. Niaz Arifin, Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews, in: Proceedings of the COLING/ACL main conference poster sessions, 2006.
[48] Y. Yang, and J. Pedersen, A comparative study on feature selection in text categorization. In Proceedings of international conference of machine learning (ICML 1997), 1997, pp. 412–420.
[49] M. Van Erp, L.G. Vuurpijl, and L. Schomaker. An overview and comparison of voting methods for pattern recognition, in: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR-8), Niagara-on-the-Lake, Canada, 2002, pp. 195–200.
[50] ⟨http://www.cs.cornell.edu/people/pabo/movie-review-data/⟩.
[51] ⟨http://www.cs.jhu.edu/~mdredze/datasets/sentiment/⟩.

Mohamed Abdel Fattah received the B.Sc. and M.Sc. degrees in Electronics from the Faculty of Engineering, Cairo University, Cairo, Egypt, in 1994 and 2003, respectively, and the Ph.D. degree in information science and intelligent systems from the University of Tokushima, Japan, in 2007. He was awarded a Japan Society of the Promotion of Science (JSPS) postdoctoral fellowship from 2007 to 2009 in Department of Information Science and Intelligent Systems, Tokushima University. He is currently an Associate Professor with FIE, Helwan University, Cairo. His research interests include information retrieval, natural language processing, speech recognition and document processing.