J Intell Inf Syst (2018) 50:549–572
DOI 10.1007/s10844-017-0482-3

An automatic classification of text documents based on correlative association of words

Deepak Agnihotri[1] · Kesari Verma[1] ·
Priyanka Tripathi[2]

Received: 7 April 2016 / Revised: 4 August 2017 / Accepted: 4 August 2017 /
Published online: 14 August 2017

Abstract Training speed of the classifier without degrading its predictive capability is an important concern in text classification. Feature selection plays a key role in this context. It selects a subset of most informative words (terms) from the set of all words. The correlative association of words towards the classes increases an incertitude for the words to represent a class. The representative words of a class are either of positive or negative nature. The standard feature selection methods, viz. Mutual Information (MI), Information Gain (IG), Discriminating Feature Selection (DFS) and Chi Square (CHI), do not consider positive and negative nature of the words that affects the performance of the classifiers. To address this issue, this paper presents a novel feature selection method named Correlative Association Score (CAS). It combines the strength, mutual information, and strong association of the words to determine their positive and negative nature for a class. CAS selects a few (k) informative words from the set of all words (m). These informative words generate a set of N-grams of length 1-3. Finally, the standard Apriori algorithm ensembles the power of CAS and CHI to select the top most, b informative N-grams, where b is a number set by an empirical evaluation. Multinomial Naive Bayes (MNB) and Linear Support Vector Machine (LSVM) classifiers evaluate the performance of the selected N-Grams. Four standard text data sets, viz. Webkb, 20Newsgroup, Ohsumed10, and Ohsumed23 are used for experimental analysis. Two standard performance measures named Macro_F1 and Micro_F1 show a significant improvement in the results using proposed CAS method.

✉ Deepak Agnihotri
agnihotrideepak@hotmail.com; dagnihotri.phd2012.mca@nitrr.ac.in

Kesari Verma
kverma.mca@nitrr.ac.in

Priyanka Tripathi
ptripathi@nitttrbpl.ac.in

[1] Department of Computer Applications, National Institute of Technology Raipur, CG, India

[2] Department of Computer Engineering and Applications, National Institute of Technical Teachers Training and Research Bhopal, MP, India

Keywords Feature selection · Text classification · Term frequency · Text analysis · Text mining

1 Introduction

Tremendous growth of text data due to heavy use of electronic devices and internet technologies, necessitates efficient techniques or tools (like Text Mining) that automatically arrange text documents into known classes[1,2] (Joachims 1996). In a multi-class environment of text classification, the classifier algorithm predicts the class label of new documents based on the training of the model. In the real world, its applications are sentiment classification, spam filtering, classification of Pubmed articles, organizing web contents into topical hierarchies, and news filtering, etc. In this paper, the text documents are classified by two standard classifiers, Linear Support Vector Machines (LSVM) (Joachims 1996) and Multinomial Naive Bayes (MNB) (Manning et al. 2008; Sebastiani 2002; Joachims 1998). MNB is known as the fastest probabilistic generative learning model. However, its accuracy is relatively modest. LSVM is based on graceful foundations of statistical learning theory. The training time of SVM is higher than MNB but the classification results of SVM are more accurate. The strength of the classifier depends upon the contents of documents. The contents are grammatical sequences of the words organized in the form of sentences. The word (named term) is the smallest constituent of the text contents and plays a vital role in text classification.

Text classification process uses following steps to select the most relevant words as features: (1) feature extraction from the corpus (i.e. generation of tokens from text contents), (2) elimination of less informative words (i.e. stop words, punctuation marks, numbers, links, white spaces, etc.), and (3) lemmatization or stemming of the words (Agnihotri et al. 2017). Finally, the resultant words build a vocabulary of the corpus that helps in the construction of a vector space. The frequency of each word present in the documents is represented as a vector (Yang and Pedersen 1997). Term Frequency-Inverse Document frequency (TF-IDF) based scaling technique is used to normalize the frequency of the words (Mladenic and Grobelnik 1999). The collection of word vectors in a matrix form is called a vector space. In this vector space, each individual word constitutes one dimension. For a typical document collection, there may be millions of words. Thus, text classification process requires a much larger dimension to fit in a limited memory space that makes this process cumbersome (Kevin and Moshe 2013). Feature selection techniques eliminate the less informative features and selects a reduced subset of features for training. It increases the performance as well as the speed of the classifier and considered as an important step in the classification process (Lamirel et al. 2015; Joachims 1998).

The representative words are either of positive or negative nature due to their correlative association with many classes. The correlative association creates an uncertainty to determine the most representative words. An uneven distribution of terms, that are present in the documents belonging to different classes, has given birth to the concept of correlative association. The word which is present in the j^{th} class (say C_j) while absent in all other classes (say $\bar{C}_j$) is of a positive nature for the j^{th} class C_j. Whereas, if a word is absent in the j^{th} class C_j but present in all other classes $\bar{C}_j$ is of a negative nature for the j^{th} class C_j. The words of negative nature are also important to find out the class labels of the documents.

[1] http://www.isical.ac.in/∼acmsc/TMW2014/TMW2014.html

[2] http://www.isical.ac.in/∼scc/DInK%2710/studymaterial/textmining.eps

The presence of the negative nature words in a document, assures that this document does not belong to the class for which the word is negative (Uysal and Kursat 2016). The common words are distributed equally to all the classes, whereas the rare words belong in most of the documents of a specific class. The sparse words occur less frequently in the documents of a class, and its presence or absence is not important to decide the class label of the documents (Agnihotri et al. 2016).

The standard methods do not consider positive and negative nature of the words while determining their importance. It degrades the performance of the classifiers. To address this issue, this paper presents an information theory based new feature selection method named Correlative Association Score (CAS). It combines the strength, mutual information, and strong association of the words to determine the positive and negative nature of words. CAS selects a few (k) informative words from the set of all words (m). These informative words generate a set of N-grams of length 1–3. Finally, the standard Apriori algorithm ensembles the power of CAS and CHI to select the top most, b informative N-grams, where b is a number set by an empirical evaluation. To evaluate the performance of selected top most b N-grams, the MNB and LSVM classifier models classify four standard text data sets, viz. Webkb, 20Newsgroup, Ohsumed10, and Ohsumed23. A significant improvement in the performance of the classifiers that is based on two standard measures named Macro_F1 and Micro_F1, prove the effectiveness of the proposed CAS method. The processing flow of the proposed work is shown by the Fig. 1 (steps (1)–(13)), the main contribution of the paper concerns steps (2)–(8).

The rest of the paper comprises of six sections which are as follows: The preliminary concepts and related works are discussed in Sections 2 and 3 respectively. Section 4 describes the proposed work. Section 5 explains the experimental setup, datasets and performance evaluation metrics. Section 6 illustrates results and discussion. The paper concludes in Section 7.

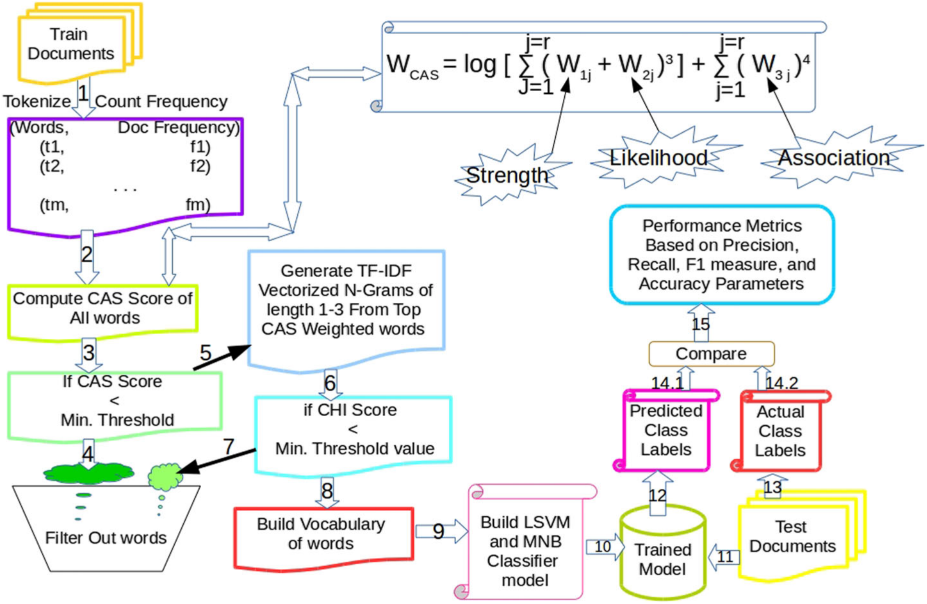
Train Documents
1
Tokenize
Count Frequency
(Words, Doc Frequency)
(t1, f1)
(t2, f2)
. . .
(tm, fm)
$W_{CAS} = \log [\sum_{J=1}^{j=r} (W_{1j} + W_{2j})^3] + \sum_{j=1}^{j=r} (W_{3j})^4$
Strength
Likelihood
Association
2
Compute CAS Score of All words
3
If CAS Score < Min. Threshold
4
Filter Out words
5
Generate TF-IDF Vectorized N-Grams of length 1-3 From Top CAS Weighted words
6
if CHI Score < Min. Threshold value
7
8
Build Vocabulary of words
9
Build LSVM and MNB Classifier model
10
Trained Model
11
Test Documents
12
13
Predicted Class Labels
Actual Class Labels
14.1
14.2
Compare
15
Performance Metrics Based on Precision, Recall, F1 measure, and Accuracy Parameters

Fig. 1 Processing flow of the proposed work

2 Preliminary concept

The preliminary concepts related to this paper, i.e. word representation, normalization, feature selection, and text document classification are described in this section.

2.1 Word representation

The representation of the words in the form of vectors is the base to determine the computational informativeness of the words and plays a vital role in an automatic classification of the text documents. The most common models to represent the words as vectors are the Bag Of Words (BOW) and N-grams Language (NGL). In BOW model, the frequency of each word in the documents of the corpus represents a vector. In this model, the order of word occurrence is not important. The N-grams are the combination of 2–4 words that co-occurred together in the documents. In the NGL model, the set of N-grams represents a vector space. Consider two documents $D1$ and $D2$, where $D1$ represents "viral disease" category and $D2$ "Bacterial disease". The contents of $D1$ and $D2$ are as follows:

1. D1: "Viral diseases are extremely widespread infections caused by viruses, a type of microorganism. There are many types of viruses that cause a wide variety of viral diseases. The most common type of viral disease is the common cold, which is caused by a viral infection of the upper respiratory tract (nose and throat)".[3]
2. D2: "Bacterial diseases include any type of illness caused by bacteria. Bacterial diseases occur when pathogenic bacteria get into the body and begin to reproduce and crowd out healthy bacteria, or to grow in tissues that are normally sterile. Bacterial diseases are contagious and can result in many serious or life-threatening complications, such as blood poisoning (bacteremia), kidney failure, and toxic shock syndrome".[4]

In documents $D1$ and $D2$, the frequency of word "disease" is higher than other words (e.g. "Bacterial" or "Viral") and looks more informative using BOW. In NGL model, the combination of the words "Bacterial", "Viral", and "disease" as "Bacterial disease" and "Viral disease" looks more informative than an individual representation of words as "disease", "Bacterial" or "Viral". Thus, the order of word occurrence is maintained in the NGL model and improves the quality of word representation. In this paper, the BOW is used at the initial level to represent the words using CAS, while NGL at the second level of filtration using CHI.

2.2 Normalization

Normalization is a technique of scaling the data in a fixed range. The authors (Dewang and Singh 2017; Agnihotri et al. 2014), and Sebastiani (2002) addressed problems like keyword spamming, scaling up frequent words and scaling down rare words. The problem of keyword spamming occurred when a word appears repeatedly in a document with the purpose of improving its ranking on the Information Retrieval system or even to create a bias towards longer documents. The word frequency in a document of a vector space is usually normalized using the Term Frequency-Inverse Document Frequency (TF-IDF) method to

[3] https://www.healthgrades.com/conditions/viral-diseases

[4] https://www.healthgrades.com/conditions/bacterial-diseases

overcome this problem (see (1)) (Agnihotri et al. 2014; Manning et al. 2008; Sebastiani 2002).

$$W_{i,j} = tf_{i,j} * \log \frac{N_d}{df_i} \tag{1}$$

Where $W_{i,j}$ = weight for word t_i in document d_j, N_d = Total number of documents in the corpus, $tf_{i,j}$ = frequency of word t_i in document d_j, df_i = document frequency of i^{th} word in the corpus.

2.3 Feature selection

Feature selection improves the performance and accelerate the training speed of the classifiers. It reduces a huge feature space into a smaller subset. Let us define, p as the total number of words in the corpus, and n as the total number of documents. Subsequently, the text contents of the entire corpus D (as discussed in Section 2.4) is extracted as tokens p and kept in a set t. Let $t = [t_1, t_2, ..., t_p]$, where $p > 0$ and each word contains some information to discriminate the class label of the documents. The selection of words $t_q \in t$ that contain the maximum information to discriminate a class label which helps in correct classification of the documents is known as feature selection (Agnihotri et al. 2016).

2.4 Text documents classification

In text document classification, the documents set ($D = [d_1, d_2, ..., d_j]$) of a r number of classes $C = [C_1, C_2, ..., C_r]$ is divided into two subsets, i.e. training (D_{train}) and test (D_{test}). The objective of the classification is to build a model based on the known class labels of training set documents which have the capability to predict the class labels of test documents with maximum accuracy (Manning et al. 2008; Sebastiani 2002; Joachims 1998).

3 Related works

In literature, many researchers have contributed significantly in the area of feature selection. The core contribution of this paper is compared with four state-of-the-art methods, viz. MI, IG, DFS, and CHI. The brief description of these methods is given in this section.

Mutual information (MI) concept (Manning et al. 2008; Joachims 1998; Yang and Pedersen 1997) is carried out from information theory to measure the dependencies between random variables and used to measure the information contained by a word $t_i \in t$. If the feature word t_i possesses higher mutual information with the class C_j, it contains more information about the class C_j. The MI computes the dependence between the word t_i and the class C_j using (2) and the MI weight of the word t_i is computed using (3). The preliminary notations used in this study are defined in Table 1.

$$MI(t_i, C_j) = \log(\frac{p(t_i, C_j)}{p(t_i) \times p(C_j)}) \approx \log \frac{a \times N}{(a+c) \times (a+b)} \tag{2}$$

$$MI(t_i) = \max_{j=1}^{j=r} MI(t_i, C_j) \tag{3}$$

The Information Gain (IG) is a measure of reduction in entropy for words when they are separated into different classes. The IG score of a word t_i given in (4) is the contribution of

Table 1 Preliminary notations

Notations	Value	Meaning
a	$count(t_i, C_j)$	count of word t_i in the documents of class C_j
b	$count(t_i, \bar{C}_j)$	count of word t_i in the documents of other classes $\bar{C}_j$
c	$count(\bar{t}_i, C_j)$	count of other words $\bar{t}_i$ in the documents of class C_j
d	$count(\bar{t}_i, \bar{C}_j)$	count of other words $\bar{t}_i$ in the documents of other classes $\bar{C}_j$
N	$(a + b + c + d)$	total number of words in all r numbers of classes
df	$df(t_i, C_j)$	document frequency of word t_i in class C_j
$maxf$	$max(t_i, C_j)$	maximum frequency of word t_i in class C_j
$avgf$	$mean(t_i, C_j)$	average frequency of word t_i in class C_j
$p(t_i)$	$(a + b)/N$	The probability of word t_i
$p(\bar{t}_i)$	$(c + d)/N$	The probability of other words $\bar{t}_i$
$p(C_j)$	$(a + c)/N$	The probability of class C_j
$p(\bar{C}_j)$	$(b + d)/N$	The probability of other classes $\bar{C}_j$
$p(t_i, C_j)$	a/N	The probability of word t_i for being in class C_j
$p(t_i, \bar{C}_j)$	b/N	The probability of other words $\bar{t}_i$ for being in class C_j
$p(\bar{t}_i, C_j)$	c/N	The probability of word t_i for being in other classes $\bar{C}_j$
$p(\bar{t}_i, \bar{C}_j)$	d/N	The probability of other words $\bar{t}_i$ for being in other classes $\bar{C}_j$
$p(t_i\|C_j)$	$a/(a + c)$	The probability of word t_i when class C_j is present
$p(\bar{t}_i\|C_j)$	$c/(a + c)$	The probability of word t_i when other classes $\bar{C}_j$ are present
$p(t_i\|\bar{C}_j)$	$b/(b + d)$	The probability of other words $\bar{t}_i$ when class C_j is present
$p(\bar{t}_i\|\bar{C}_j)$	$d/(b + d)$	The probability of other words $\bar{t}_i$ when other classes $\bar{C}_j$ are present
$p(C_j\|t_i)$	$a/(a + b)$	The probability of class C_j when word t_i is present
$p(\bar{C}_j\|t_i)$	$b/(a + b)$	The probability of other classes $\bar{C}_j$ when word t_i is present
$p(C_j\|\bar{t}_i)$	$c/(c + d)$	The probability of class C_j when other words $\bar{t}_i$ are present
$p(\bar{C}_j\|\bar{t}_i)$	$d/(c + d)$	The probability of other classes $\bar{C}_j$ when other words $\bar{t}_i$ are present

word t_i in class C_j (Uysal and Gunal 2012; Forman 2003; Yang and Pedersen 1997; Lewis and Ringuette 1994).

$$\begin{aligned} IG(t_i) \; &= \; p(t_i) \times \sum_{j=1}^{j=r} p(C_j|t_i) \times \log p(C_j|t_i) \\ &\quad + p(\bar{t}_i) \times \sum_{j=1}^{j=r} p(C_j|\bar{t}_i) \times \log p(C_j|\bar{t}_i) - \sum_{j=1}^{j=r} p(C_j) \times \log p(C_j) \end{aligned} \tag{4}$$

Uysal and Gunal (2012) defined the Distinguishing Feature Selector (DFS) method to compute the weight of a word t_i for a class C_j shown in (5).

$$DFS(t_i) = \sum_{j=1}^{j=r} \frac{p(C_j|t_i)}{p(\bar{t}_i|C_j) + p(t_i|\bar{C}_j) + 1} \tag{5}$$

Mathematically, Chi square testing is used to determine the independence of the word t_i and class C_j during the feature selection. If CHI (t_i, C_j) = 0, the word t_i and class C_j are independent, thus the word t_i does not contain category information. Otherwise, if the value of the CHI (t_i, C_j) is higher, the word t_i contains more information to represent the class

C_j. The contribution of word t_i in class C_j (see (6)) is used to compute the contribution of word t_i using the CHI method (Manning et al. 2008; Yang and Pedersen 1997).

$$CHI(t_i) = \sum_{j=1}^{j=r} p(C_j) \times \frac{N \times (a \times d - b \times c)^2}{(a+c) \times (a+b) \times (c+d) \times (b+d)} \tag{6}$$

4 Proposed work

Substantial works were carried out in the area of feature selection to improve the prediction capability of the classifiers. The standard methods, viz. Mutual Information (MI), Information Gain (IG), Discriminating Feature Selection (DFS) and Chi Square (CHI), did not consider positive and negative nature of the words that affects the performance of the classifiers. To address this issue, a new feature selection method named Correlative Association Score (CAS) is proposed. CAS combines the strength, mutual information, and strong association of the words to determine the positive and negative nature of the words for the class. The weight assignment process of the CAS is shown by the Algorithm 1 and its summary is as follows:

Algorithm 1 Algorithm for computation of correlative association score (CAS) of terms

Declaration:

1. Input is a set D of documents of each class $C_k \in C$. $D = [d_1, d_2, ..., d_n]$, where $n > 0$. $C = [C_1, C_2, ..., C_j]$ where, $0 < j <= r$.
2. The output of the algorithm is a set of most informative features $t[k] \subset t[m] \subset t[p]$.

Procedure:

```
$D = D_{train} + D_{test}$, Where $D_{train}$ is the training set corpus, and $D_{test}$ is the test corpus.
$t[k]$=FEATURES($D_{train}, th1$)
function wordScore$t_i, tf_{ij}$                                ▷ Computing CAS Score of each word
    $CAS(t_i) = \log\left(\sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3\right) + \sum_{j=1}^{j=r}(W_{3j}^4)$
    return $(CAS(t_i))$
function Preprocessing$D$                                       ▷ Preprocessing of documents in the corpus
    $T = [t_1, t_2, ..., t_p] \leftarrow Tokenizer(D)$              ▷ Tokenization
    $T$= stopWordsRemoval($T$)                                  ▷ Stop words removal
    $T$= punctuationMarksRemoval($T$)                           ▷ Punctuation marks removal
    $T \leftarrow [t_1, t_2, ..., t_m]$ = whiteSpaceRemoval($T$)   ▷ Where $m < p$   ▷ White Space Removal
    return $(T)$
function Features$D_{train}, th1$                               ▷ Selection of most informative words
    $t$=PREPROCESSING($D_{train}$)
    $\sum_{i=1}^{i=m} \sum_{j=1}^{j=r} tf_{ij} \leftarrow n(t_i|C_j)$   ▷ $tf_{ij}$ is the occurring frequency of $i^{th}$ word $t_i$ in $j^{th}$ class
    $ts[t_i]$=WORDSCORE($t_i, tf_{ij}$)
    $FS[m] \leftarrow Sort(t_i, ts[t_i])$                       ▷ Sorting in descending order
    $FS[k] \leftarrow Select(FS[m], th1)$                       ▷ Where $k < m$
    return $(FS)$
```

1. In this study, the NGL model is used to represent the words as a set, i.e. $NG[b]$ of b N-Grams of length 1–3. The set $NG[b]$ has been generated by using the **join** and **prune** steps of the Apriori algorithm.

   (a) *The join step:* Suppose $L_{k-1} = \{t_1, t_2, .., t_m\}$ is the set of uni-grams, $L_k = \{t_1t_2, .., t_{m-1}t_m\}$ is the set of bi-grams, i.e. (t_{m-1}, t_m) where $t_{m-1}, t_m \in L_{k-1}$. Similarly, $L_{k+1} = \{t_1t_2t_3, .., t_{m-2}t_{m-1}t_m\}$ is the set of tri-grams. Finally, the set $NG[g]$ is generated by taking the union of $L_{k+1} \bigcup L_k \bigcup L_{k-1}$.

   (b) *The prune step:* This step eliminates less informative words, initially from set L_k, and subsequently from set $NG[g]$, by using a threshold value (i.e. determined empirically).

2. The CAS extracts a set of m most discriminating words L_{k-1} from the set of all words using a threshold value. It computes weight W_{CAS} of each word t_i as follows:

   (a) The CAS computes a unique weight of each word on the basis of three criteria, first criterion computes weight W_{1j} to measure the strength of i^{th} word t_i for j^{th} class C_j (i.e. $W_1(t_i, C_j)$), second criterion computes weight W_{2j} to measure the likelihood of class C_j when the word t_i is present (i.e. $W_2(C_j|t_i)$), and third criterion computes weight W_{3j} of the word t_i to measure the association of t_i with class C_j (i.e. $W_3(t_i, C_j)$). The resultant weight (W_{CAS}) of the word t_i is computed as,

$$\mathbf{W}_{CAS} = \log\left(\sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3\right) + \sum_{j=1}^{j=r}\left(W_{3j}^4\right) \tag{7}$$

   Where,

$$W_1(t_i, C_j) = \frac{maxf(t_i, C_j)}{\epsilon + avgf(t_i, C_j)} + \log_2\left[\epsilon + \frac{\epsilon + (p(t_i|C_j) \times (1 - p(t_i|\bar{C_j})}{\epsilon + (p(t_i|\bar{C_j}) \times (1 - p(t_i|C_j)}\right] \tag{8}$$

$$W_2(C_j|t_i) = a \times \log\frac{p(t_i, C_j)}{p(t_i) \times p(C_j)} + b \times \log\frac{p(t_i, \bar{C_j})}{p(t_i) \times p(\bar{C_j})} \tag{9}$$

$$W_3(t_i, C_j) = \left|\frac{a}{a + c + df_{[t_i, j]}} - \frac{c}{a + c + df_{[t_i, j]}}\right| \tag{10}$$

   (b) Sort the words in descending order based on the CAS Weight (W_{CAS}).

   (c) Select the top t of CAS weighted words from set L_k based on a threshold value t.

   (d) Generate the set of N-grams, i.e. $NG[g]$ of length 1-3 from the top t of CAS weighted words. Normalize the frequencies of N-grams using TF-IDF weight (Agnihotri et al. 2014; Sebastiani 2002; Mladenic and Grobelnik 1999) (see (1)).

2. Finally, CHI method is applied to select the top most, b discriminating N-grams $NG[b] \subset NG[g]$.

The time complexity of the Algorithm 1 is $\mathcal{O}(p \times n \times r)$, where n is the total number of documents, r is the total number of classes, p is the total number of words, m number of words obtained after removal of stop words, punctuation marks and white spaces, k of CAS weighted words are selected as the most informative words based on a threshold value.

Table 2 Example dataset

Words ↓	C1			C2				C3					
	Documents			Documents				Documents					
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
toad	0	0	0	3	1	1	1	0	0	0	0	0	0
cat	1	0	1	1	0	0	0	1	0	0	0	0	1
shark	0	0	0	0	0	0	0	2	1	1	0	1	0
cow	1	0	0	0	0	0	1	0	0	0	1	0	0
rays	0	0	0	0	0	0	0	1	3	1	0	4	2
ostrich	1	1	2	0	0	0	0	1	2	0	0	1	1
emu	0	0	0	1	0	1	1	1	0	1	1	1	0
mouse	1	1	1	2	1	0	1	0	0	0	0	0	0
turtle	3	2	1	0	0	0	0	0	0	0	0	0	0

4.1 Explanation of CAS

The example data shown in the Table 2 explains the process of weight assignment by CAS. In this dataset, there are three categories of documents (i.e. $C1$, $C2$, and $C3$) with eight unique words. The confusion matrix shown in the Table 3 represents the values of a, b, c, and d (see Table 1) of the words. An analysis of words, such as their document, maximum, and average frequencies are shown in Table 4. The properties of eight unique words of Table 2 are as follows:

Table 3 Confusion matrix for words as per preliminary notations shown in Table 1

Words	C_1	$\bar{C}_1$	C_2	$\bar{C}_2$	C_3	$\bar{C}_3$
$toad = 1$	$a1 = 0$	$b1 = 6$	$a2 = 6$	$b2 = 0$	$a3 = 0$	$b3 = 6$
$toad = 0$	$c1 = 16$	$d1 = 29$	$c2 = 0$	$d2 = 45$	$c3 = 29$	$d3 = 16$
$cat = 1$	$a1 = 2$	$b1 = 3$	$a2 = 1$	$b2 = 4$	$a3 = 2$	$b3 = 3$
$cat = 0$	$c1 = 4$	$d1 = 28$	$c2 = 9$	$d2 = 23$	$c3 = 19$	$d3 = 13$
$shark = 1$	$a1 = 0$	$b1 = 5$	$a2 = 0$	$b2 = 5$	$a3 = 5$	$b3 = 0$
$shark = 0$	$c1 = 16$	$d1 = 26$	$c2 = 19$	$d2 = 23$	$c3 = 7$	$d3 = 35$
$cow = 1$	$c1 = 1$	$d1 = 2$	$c2 = 1$	$d2 = 2$	$c3 = 1$	$d3 = 2$
$cow = 0$	$c1 = 9$	$d1 = 40$	$c2 = 14$	$d2 = 35$	$c3 = 26$	$d3 = 23$
$rays = 1$	$a1 = 0$	$b1 = 11$	$a2 = 0$	$b2 = 11$	$a3 = 11$	$b3 = 0$
$rays = 0$	$c1 = 16$	$d1 = 22$	$c2 = 19$	$d2 = 19$	$c3 = 3$	$d3 = 35$
$ostrich = 1$	$a1 = 4$	$b1 = 5$	$a2 = 0$	$b2 = 9$	$a3 = 5$	$b3 = 4$
$ostrich = 0$	$c1 = 0$	$d1 = 25$	$c2 = 19$	$d2 = 6$	$c3 = 6$	$d3 = 19$
$emu = 1$	$a1 = 0$	$b1 = 12$	$a2 = 7$	$b2 = 5$	$a3 = 5$	$b3 = 7$
$emu = 0$	$c1 = 16$	$d1 = 12$	$c2 = 2$	$d2 = 26$	$c3 = 10$	$d3 = 18$
$mouse = 1$	$a1 = 3$	$b1 = 4$	$a2 = 4$	$b2 = 3$	$a3 = 0$	$b3 = 7$
$mouse = 0$	$c1 = 0$	$d1 = 31$	$c2 = 2$	$d2 = 29$	$c3 = 29$	$d3 = 2$
$turtle = 1$	$a1 = 6$	$b1 = 0$	$a2 = 0$	$b2 = 6$	$a3 = 0$	$b3 = 6$
$turtle = 0$	$c1 = 0$	$d1 = 48$	$c2 = 19$	$d2 = 29$	$c3 = 29$	$d3 = 19$

Table 4 Analysis of word's: document, maximum, and average frequencies

Words	C_1			C_2			C_3		
	df	maxf	avgf	df	maxf	avgf	df	maxf	avgf
toad	0	0	0	4	3	1.5	0	0	0
cat	2	1	0.67	1	1	0.25	2	1	0.33
shark	0	0	0	0	0	0	4	2	0.83
cow	1	1	0.33	1	1	0.25	1	1	0.17
rays	0	0	0	0	0	0	5	4	1.83
ostrich	3	2	1.33	0	0	0	4	2	0.83
emu	0	0	0	3	4	1.75	4	2	0.83
mouse	3	1	1	3	2	1	0	0	0
turtle	3	3	2	0	0	0	0	0	0

1. The word "turtle" is of a positive nature for class C1, whereas "emu" is negative.
2. The word "toad" is of a positive nature for class C2, whereas "ostrich" is negative.
3. The words "shark" and "rays" are of a positive nature for class C3, whereas "mouse" is negative.
4. The words "cat" and "cow' are present in all three classes, i.e. C1, C2, and C3 and named common words.

Table 5 describes mathematical computations of each ensemble part of CAS method. Table 6 presents an explanation for the use of ensemble parts in CAS method. The weight assigned by the CAS method are compared with MI, IG, DFS, and CHI methods (see Table 7). The CAS has assigned a much higher weight to the words of a positive nature (i.e. "rays, turtle, toad, and shark"), no matter the words are used most frequently or not. A medium weight to the negative words (i.e. "mouse, ostrich, and emu"), and a lower weight to the common (i.e. "cat, and cow") and sparse words. Each ensemble part of CAS has its own value to decide an importance of the words. The ensemble parts of CAS are as follows:

Strength of a word to represent a class (W_1) The strength of the word t_i depends on its occurrence in a class C_j compared to other classes $\bar{C}_j$. It is the sum of two ratios. The first is a ratio of maximum occurrence (say, $maxf(t_i, C_j)$) of t_i with its average occurrence (say, $avgf(t_i, C_j)$) in class C_j (see (8)). Second, is the ratio similar to the odds ratio method (Rehman et al. 2015; Forman 2003). The second ratio of (8) assigns a highest positive value to the rare words of positive nature, whereas the negative values for the common words of negative nature for a class. Its resultant sum with the first ratio has balanced the negative value with positive value.

Likelihood of a word for a class (W_2) It is an improvement of the standard MI (Forman 2003) method, where each logarithmic quantity is multiplied by $p(t_i, C_j)$ and $p(t_i, \bar{C}_j)$ (see Table 1). In W_2, each logarithmic quantity is multiplied with the total occurrence of a word t_i in the documents of a class C_j and other classes $\bar{C}_j$ (see (9)). The likelihood weight (W_{2j}) assigns a very high weight to the rare words of a positive nature and a medium weight to the common words of a negative nature for the class C_j; e.g. "*toad*" (12.77) and "*emu*" (6.12) (see Table 5).

Ensemble of W_{1j} with W_{2j} as $\log \sum_{j=1}^{j=r} (W_{1j} + W_{2j})^3$ An ensemble of W_1 with W_2 increases weight of the rare words with positive nature and common words of a negative

Table 5 Analysis of ensemble parts of CAS

Words	$\frac{maxf}{avgf}$			$\log\left(\frac{ad}{bc}\right)$			$Likelihood$			$Association$		
	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
toad	0	1.5	0	−0.68	6.29	−0.69	2.25	12.77	5.03	1	0.6	1
cat	0.86	1.3	1.2	1.61	0.13	−0.04	0.79	0.07	0.28	0.25	0.73	0.74
shark	0	0	1.5	−0.68	−0.68	5.86	2.08	2.58	6.8	1	1	0.125
cow	1.2	1.3	1.5	0.99	0.56	−0.05	0.16	0.01	0.21	0.73	0.81	0.89
rays	0	0	1.71	−0.69	−0.69	6.65	4.34	5.39	13.76	1	1	0.42
ostrich	1.1	0	1.5	5.31	−0.69	1.48	2.97	7.35	1.02	0.57	1	0.07
emu	0	1.8	1.5	−0.69	2.9	0.58	6.12	3.55	0.04	1	0.42	0.26
mouse	0.7	1.3	0	5.23	2.91	−0.69	3.11	3.1	10.04	0.5	0.22	1
turtle	1.2	0	0	6.36	−0.69	−0.69	13.11	2.6	4.61	0.7	1	1

nature. Further, It decreases the weight of most common (i.e. present in all the classes) and sparse words (see Table 6). Table 6 presents an analysis of the ensemble parts of CAS. In this table we can observed that if W_1 is multiplied with W_2 in spite of addition, the resultant weight of the words is zero or negative for some classes in which the occurrence of these words is lesser to other classes. Whereas, it causes a very high increase in the positive values of the words for the classes in which they are most frequent. Therefore, the resultant weight, i.e. $W_1 \times W_2$ does not fit for the current scenario, but definitely it may be a choice with some other approaches. Hence, $W_1 + W_2$ is the best choice for the proposed method. The 3^{rd} power of $W_{1j} + W_{2j}$ is chosen empirically (see Fig. 2). Figure 2 shows that the characteristics of 1^{st}, 2^{nd}, 3^{rd}, and 4^{th} power are similar, but the goodness of fit of the classification model is better for 3^{rd} power rather than 1^{st} or 2^{nd}. Whereas, for the 4^{th} power the classification model over-fits the words of training documents and consequently degrades the performance. One way to understand over-fitting intuitively is that a model may use some relevant words of the data (signal) and some irrelevant words (noise). An over-fitted model picks up the noise or random fluctuations in the training data, which increases its performance in case of known noise (training data) and decreases its performance in the case of novel noise (test data). Thus, over-fitting negatively impacts the performance of the model on new data. The aim is to keep the cube value of $W_{1j} + W_{2j}$

Table 6 Explanation of CAS method

Words	$\log\left(\sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3\right)$: (Rank)	$\log\left(\sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3\right) + \sum_{j=1}^{j=r}\left(W_{3j}^4\right)$: (Rank)
toad	9.08 : (3)	11.21 : (3)
cat	3.72 : (8)	4.3 : (9)
shark	7.95 : (4)	9.95 : (4)
cow	3.2 : (9)	4.55 : (8)
rays	9.3 : (1)	11.35 : (1)
ostrich	7.07 : (6)	8.18 : (6)
emu	6.59 : (7)	7.62 : (7)
mouse	7.57 : (5)	8.64 : (5)
turtle	9.09 : (2)	11.29 : (2)

Table 7 Comparison of scores on Example Dataset

Words↓	MI (rank)	IG (rank)	DFS (rank)	CHI (rank)	CAS (rank)
toad	0.25 (3)	0.36 (1)	1 (1)	21.02 (2)	11.21 (3)
cat	0.14 (8)	0.06 (7)	0.68 (8)	1.07 (8)	4.3 (9)
shark	0.35 (2)	0.12 (6)	0.78 (5)	7.67 (7)	9.95 (4)
cow	0.1 (9)	0.03 (8)	0.77 (6)	0.29 (9)	4.55 (8)
rays	0.36 (1)	0.31 (3)	0.9 (3)	17.11 (3)	11.35 (1)
ostrich	0.17 (7)	0.26 (5)	0.83 (4)	10.37 (5)	8.18 (6)
emu	0.21 (5)	0.33 (2)	0.71 (7)	8.06 (6)	7.62 (7)
mouse	0.20 (6)	0.31 (3)	0.94 (2)	17.7 (4)	8.64 (5)
turtle	0.24 (4)	0.27 (4)	1 (1)	21.83 (1)	11.29 (2)

as positive or negative for the j^{th} class, thus the selection of odd numbers viz. 1, 3, or 5 as powers is a better choice. The resultant weight will always be positive due to ensemble of W_{1j} and W_{2j} (see Table 6). The logarithm of the resultant sum is taken to normalize the weight, but in a few cases if the resultant normalized weight is negative then it becomes positive when it is summed with $\sum_{j=1}^{j=r} W_{3j}^4$.

Association of a word to specific class (W_3) The main motto for computation of an association value, i.e. W_{3j} of a word t_i (see (10)) is to discriminate the common words more effectively and increase the weight of positive and negative nature of words optimally. In this context, an association value of word t_i is computed for each class that depends upon frequency of the words in the class. Thus, if a word is absent in a class its association value will be maximum (i.e. 1) for that class, i.e. more the frequency of a word in a class, less will be its association value in that class. E.g. the word "*toad*" is present in C2 class, while absent in the C1 and C3 (see Table 5). As a result, the association value for C1 and C3 is 1, whereas 0.6 for C2. The resultant value of W_{3j} is always in the range of 0 and 1. Thus, 4^{th} power of W_{3j} is a lesser value for the class in which the word is most frequent, whereas it is maximum 1 for the class in which the word is absent. Therefore, the resultant association value of "*toad*" is $1^4 + 0.6^4 + 1^4 = 2.13$. As the word "cat" is present in all three classes, its resultant association value is $0.25^4 + 0.73^4 + 0.74^4 = 0.59$. Similarly, the word "emu"

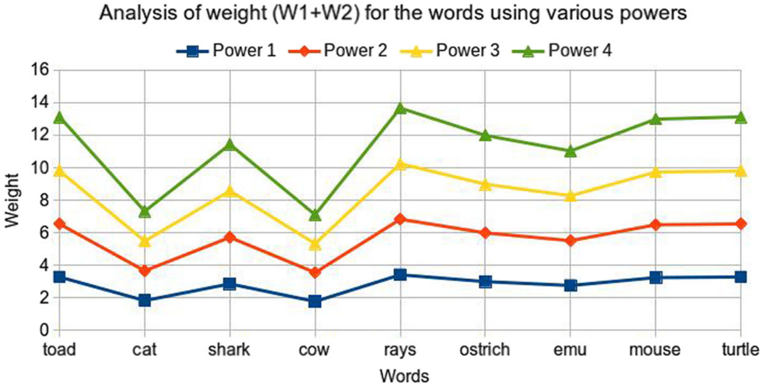

Fig. 2 Analysis of weight $W_1 + W_2$ for various powers

is present in C2 and C3 class, while absent in C1 class, is considered as negative word for class C3. Its resultant association value is $1^4 + 0.42^4 + 0.26^4 = 1.04$. It can be observed by above examples that the $\sum_{j=1}^{j=r} W_{3j}^4$ has assigned highest weight to the rare positive words, higher weight to the rare negative words, and least weight to the common and sparse words. Figure 3 shows the characteristics of various powers of W_{3j}, e.g. the nature of 4^{th} and 5^{th} power is similar, but it slightly differ with 3^{rd} power and noticeable change with 2^{nd} and 1^{st} power. The goodness of fit of the classification model is better for 4^{th} power rather than $1, 2, 3,\ or\ 5^{th}$ power.

Ensemble of $\log \sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3$ with $\sum_{j=1}^{j=r} W_{3j}^4$ As shown in Table 6, using $\log \sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3$ alone suffers in discrimination of common words. E.g. the frequency of common word “cat” is more than “cow”, therefore “cat” is more common than “cow” and it should get a lower rank than “cow”, but $\log \sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3$ has assigned 8^{th} rank to cat and 9^{th} rank to cow. To overcome this issue, $\sum_{j=1}^{j=r} W_{3j}^4$ is summed with $\log \sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3$ which assigns 8^{th} rank to “cow” and 9^{th} rank to “cat”. The result proves that $\log \sum_{j=1}^{j=r}(W_{1j}+W_{2j})^3+\sum_{j=1}^{j=r} W_{3j}^4$ discriminates common words more appropriately and increases the proportional weight of rare positive and negative nature words better than $\log \sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3$. Further, if multiplication is used in place of addition, i.e. $\log \sum_{j=1}^{j=r}(W_{1j} + W_{2j})^3 \times \sum_{j=1}^{j=r} W_{3j}^4$, then the resultant weight will be much higher, which causes over-fitting of less informative words by the classification model and degrade the performance. Thus, the proposed CAS method shown in (7) is empirically evaluated and found most suitable for the selection of most informative words.

5 Experimental setup and performance evaluation

All the experiments have been carried out on a machine with Intel core i7, 8GB RAM, 1.8 GHz Processor in UBUNTU 14.04 64-bit OS. The steps of text document classification, i.e. Tokenization, preprocessing of the words of the corpus (D), feature extraction

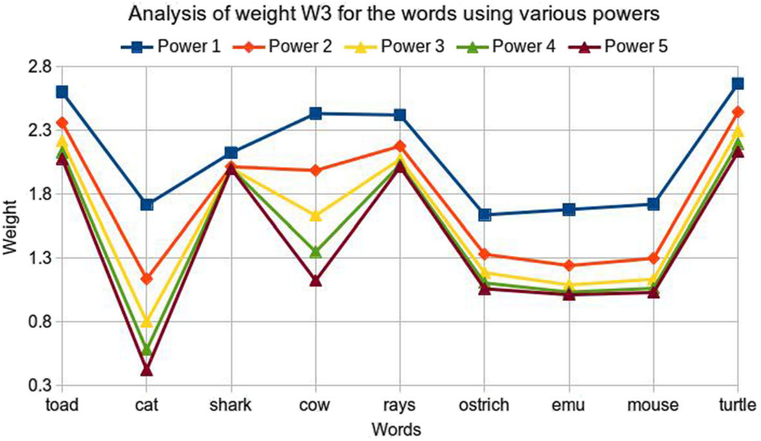

Fig. 3 Analysis of weight W_3 for various powers

Table 8 Details of the datasets

Dataset	Categories Name	# Class
Webkb[a]	cornell.course, cornell.faculty, cornell.project, cornell.student, cornell.other, texas.course, texas.faculty, texas.project, texas.student, texas.other, washington. course, washington.faculty, washington.project, washington.student, washington.other, wisconsin. course, wisconsin.faculty, wisconsin.project, wisconsin.student, wisconsin.other, misc.course, misc.faculty, misc.project, misc.student, misc.other	25
20Newsgroup[b] (Mitchell 1997; Joachims 1998)	talk.religion.misc, talk.politics.misc, alt.atheism, talk.politics.guns, talk.politics.mideast, comp.os.ms-windows.misc, comp.sys.mac.hardware, comp.graphics, misc.forsale, comp.sys.ibm.pc.hardware, sci.electronics, comp.windows.x, sci.space, rec.autos,sci.med, sci.crypt, rec.sport.baseball, rec.motorcycles, soc.religion.christian, rec.sport.hockey	20
Ohsumed10[c] (Agnihotri et al. 2016)	C01, C04, C06, C08, C10, C12, C14, C20, C21, C23	10
Ohsumed23[d] (Joachims 1996)	C01, C02, C03, C04, C05, C06, C07, C08, C09, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23	23

[a] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz

[b] https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

[c] http://trec.nist.gov/data/t9_filtering.html

[d] http://disi.unitn.it/moschitti/corpora.htm

$(t[m] \subset t[p])$, feature selection $(t[k] \subset t[m]$, and $NG[f] \subset NG[g])$ classification, and performance analysis are performed in Python 2.7. The nltk, scipy, numpy, ipython notebook, scikitlearn, matplotlib, etc. packages of python2.7 are used in experimental analysis.[5] The entire corpus is sliced into multiple arrays of each class, in spite of loading entire corpus into a single array. It speeds up the computing process and resolves the memory related issues.

5.1 Data set

In this paper, four standard text datasets, viz. Webkb, 20Newsgroup, Ohsumed10, and Ohsumed23 evaluate the performance of the proposed CAS method. The detailed summary of datasets is given in Table 8. For experimental studies, the corpus D is divided into two subsets using the holdout method in which 75% data is placed in training set (D_{train}) and remaining 25% in test set (D_{test}). The “Beautifulsoup” package[6] of python2.7 is used to extract the text contents of the documents by removing tags, html links, punctuation marks, and white spaces from the documents. Subsequently, all the stop words (defined in python natural language tool kit) are removed and as a result $t[m] \subset t[p]$ words are remained for

[5] http://nbviewer.ipython.org/gist/rjweiss/7158866

[6] https://pypi.python.org/pypi/beautifulsoup4

further processing. Initially, the CAS score of all m numbers of words (t_m) are computed and arranged in descending order to select the top most, k informative words ($t[k] \subset t[m]$, where $k < m$). For experimentation, the value of k is selected as 100, 200, 500, 1000, 2000, 3000, 5000, and 10000. The set $NG[g]$ of g (where $g > k$) N-grams of length 1-3 are generated from the k CAS filtered words. The TF-IDF Vectorizer of python ”scikitlearn” package[7] is used to normalize the weight of $NG[g]$ N-grams. The Apriori algorithm is used to select the top most, b (where $b < g$) discriminating N-grams $NG[b]$ using the CHI method. The vocabulary V of b unique N-grams (as discussed in Section 4) are used to train the model.

5.2 Performance evaluation

In this paper, the benchmarked Macro and Micro averaged F_1 measures are used to evaluate the performance of MNB and LSVM (Uysal and Kursat 2016). The F measure (F_β and F_1) can be interpreted as a weighted harmonic mean of the precision and recall. The F_β score weight recall more than precision by a factor of beta. A F_β measure reaches its best value at 1 and its worst score at 0. If $\beta = 1$ then F_β and F_1 are equivalent, and the recall and the precision are equally important.[8] The accuracy gives same weight to all the classes but it is not suitable especially for imbalanced datasets. The macro F_1 measure computes metrics for each label, and finds their unweighted mean and does not consider label imbalance. Whereas, micro F_1 calculates metrics globally by counting the total true positives, false negatives and false positives. The (11)–(16) shows precision (i.e. Macro in (11) and Micro in (12)), recall (i.e. Macro in (13) and Micro in (14)), Accuracy (i.e. (15)) and F measure (i.e. (16)).

$$Precision_{Macro} = \frac{1}{n(C)} \sum_{C=1}^{C=r} \frac{TP_C}{TP_C + FP_C} \tag{11}$$

$$Precision_{Micro} = \frac{\sum_{C=1}^{C=r} TP_C}{\sum_{C=1}^{C=r} TP_C + \sum_{C=1}^{C=r} FP_C} \tag{12}$$

$$Recall_{Macro} = \frac{1}{n(C)} \sum_{C=1}^{C=r} \frac{TP_C}{TP_C + FN_C} \tag{13}$$

$$Recall_{Micro} = \frac{\sum_{C=1}^{C=r} TP_C}{\sum_{C=1}^{C=r} TP_C + \sum_{C=1}^{C=r} FN_C} \tag{14}$$

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} \tag{15}$$

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{16}$$

Where $C = 1$ to r represents r class labels and $n(C)$ is the count of the total number of classes. Let TP, FP, FN, and TN are the counts of true positives, false positives, false negatives, and true negatives respectively.

[7] http://scikit-learn.org/stable/modules/

[8] http://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics

Table 9 Peak Performance of methods in four Datasets

			Macro-F1 Avg.			Micro-F1 Avg.		
Classifier	S.NO.	Dataset	Peak value	No. of features	Method	Peak value	No. of features	method
	1.	Webkb	62.44%	1000	CAS	84.68%	500	CAS
LSVM	2.	20Newsgroup	94.81%	10000	CAS	94.46%	5000	CAS
	3.	Oshumed23	42.98%	2000	CAS	44.73%	3000	CAS
	4.	Oshumed10	53.59%	2000	CAS	53.98%	1000	CHI
	1.	Webkb	42.91%	1000	CAS	74.61%	100	CHI
MNB	2.	20Newsgroup	62.44%	1000	CAS	83.88%	200	CAS
	3.	Oshumed23	40.86%	3000	CAS	43.85%	5000	CAS
	4.	Oshumed10	52.85%	2000	CAS	52.53%	1000	CAS

6 Results and discussions

The proposed CAS method has obtained a significant improvement in the results for both LSVM and MNB classifiers (see Table 9). The experimental results are better than earlier works of Guo et al. (2009), Rehman et al. (2015). Table 9 shows that in all the datasets, the proposed CAS method has given highest Macro_F1 measure for both classifiers. Whereas, CHI has given highest Micro_F1 measure for Ohsumed10 (using LSVM) and Webkb (using MNB) datasets.

Figures 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 show the Macro_F1 and Micro_F1 obtained for different number of features for all four datasets classified by LSVM and MNB. The average rank of the CAS, MI, IG, DFS, and CHI methods are shown in the Tables 10. The lowest value indicates highest ranks. In most of the cases, the average rank of CAS is highest. The performance of MI is always at 5^{th} position, whereas there is a close competition among DFS, IG, and CHI methods for 2^{nd}, 3^{rd}, and 4^{th} positions respectively.

Table 7 presents the comparison of word scores on an example dataset (see Table 2) for various methods, viz. MI, IG, DFS, CHI, and CAS. The ranges of assigned weight by these

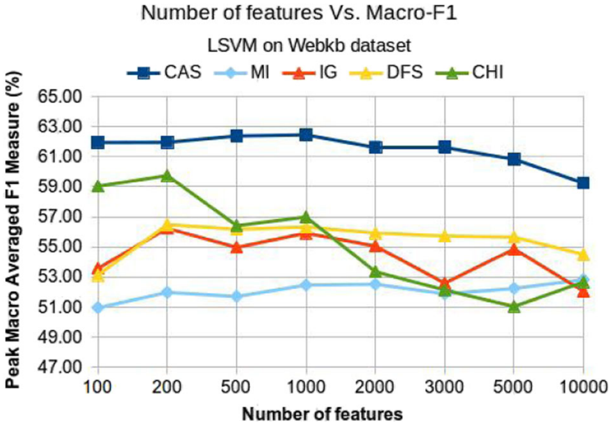

Fig. 4 Macro F1 for LSVM in Webkb Dataset

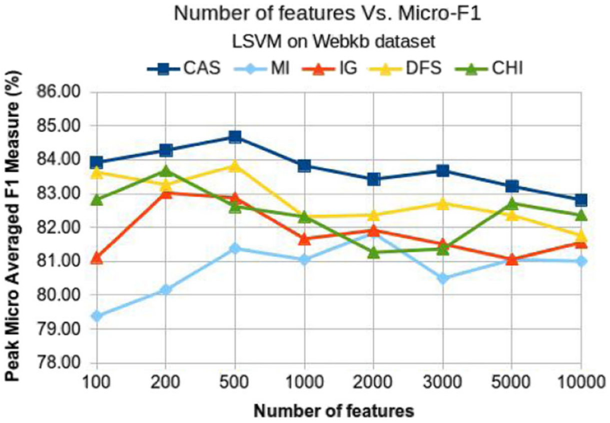

Fig. 5 Micro_F1 for LSVM in Webkb Dataset

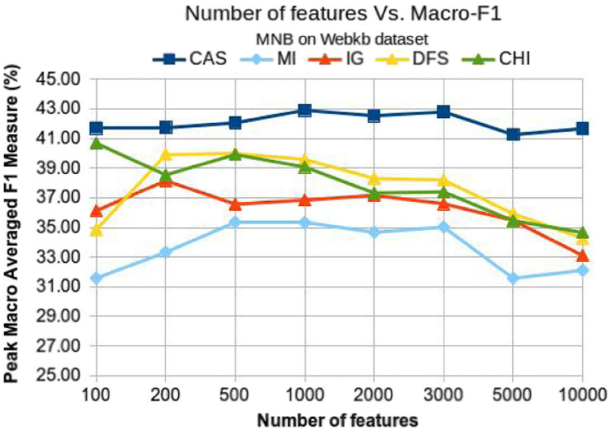

Fig. 6 Macro_F1 for MNB in Webkb Dataset

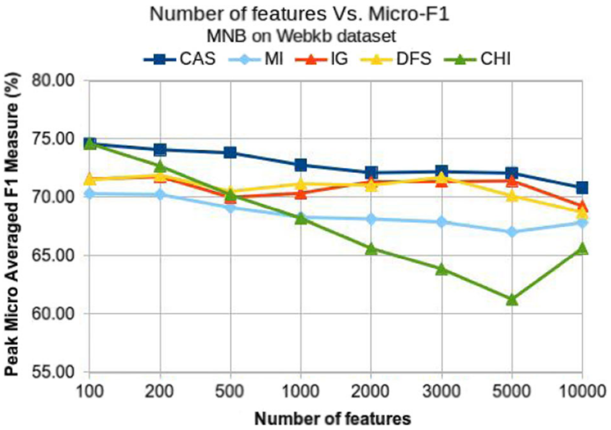

Fig. 7 Micro_F1 for MNB in Webkb Dataset

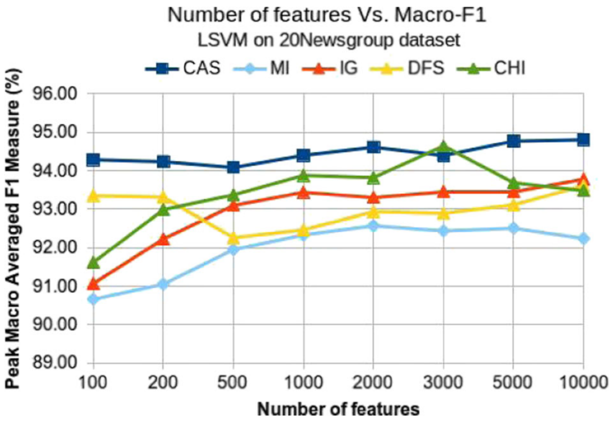

Fig. 8 Macro_F1 for LSVM in 20Newsgroup dataset

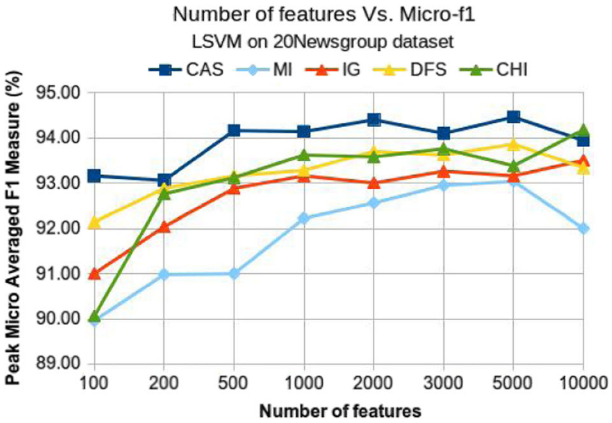

Fig. 9 Micro_F1 for LSVM in 20Newsgroup

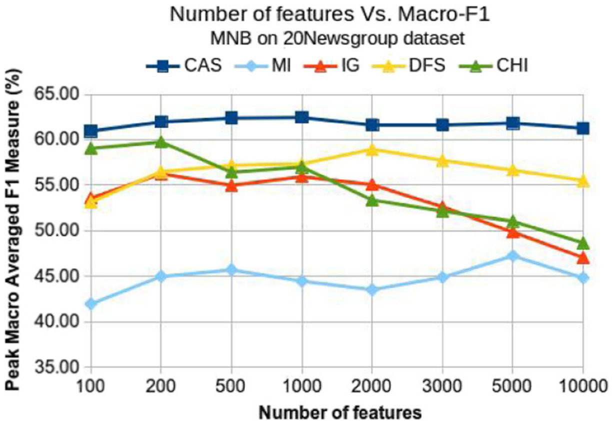

Fig. 10 Macro_F1 for MNB in 20Newsgroup dataset

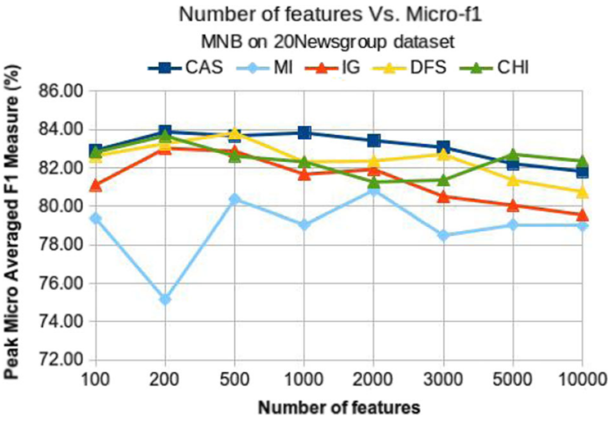

Fig. 11 Micro_F1 for MNB in 20Newsgroup dataset

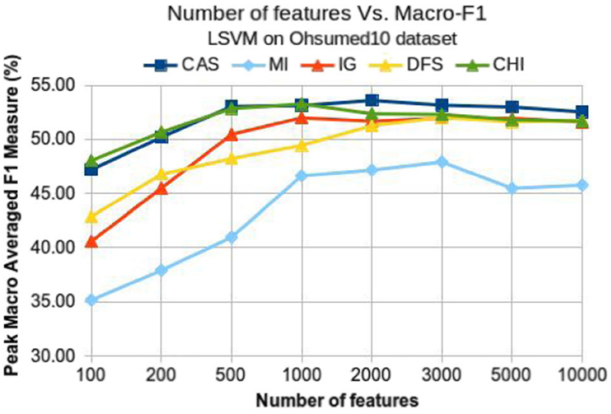

Fig. 12 Macro_F1 for LSVM in Ohsumed10 dataset

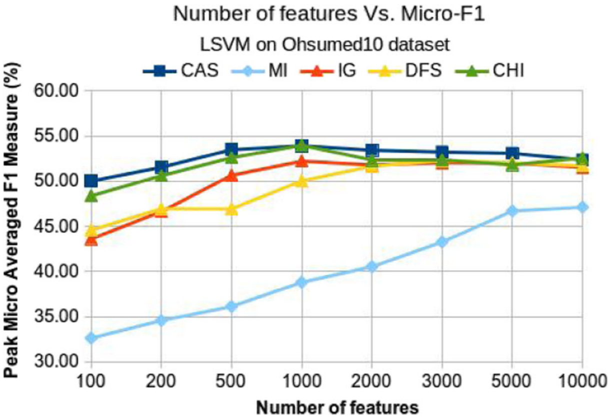

Fig. 13 Micro_F1 for LSVM in Ohsumed10 dataset

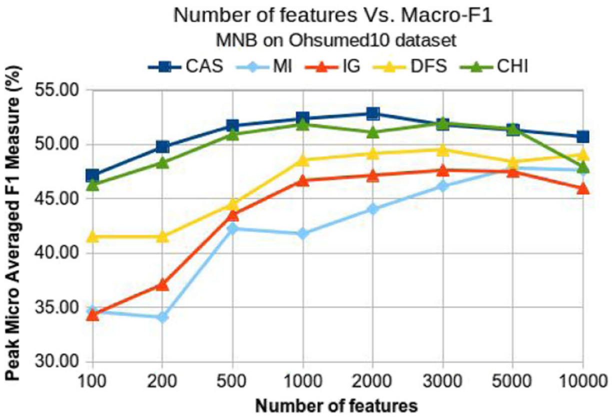

Fig. 14 Macro_F1 for MNB in Ohsumed10 dataset

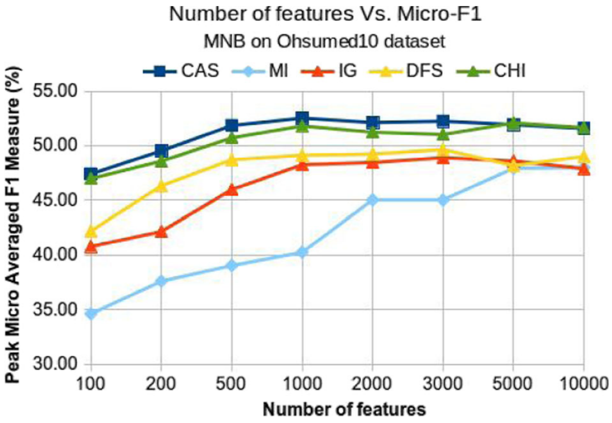

Fig. 15 Micro_F1 for MNB in Ohsumed10 dataset

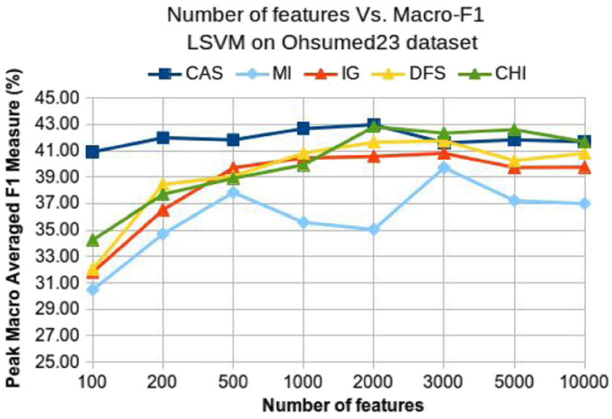

Fig. 16 Macro_F1 for LSVM in Ohsumed23 dataset

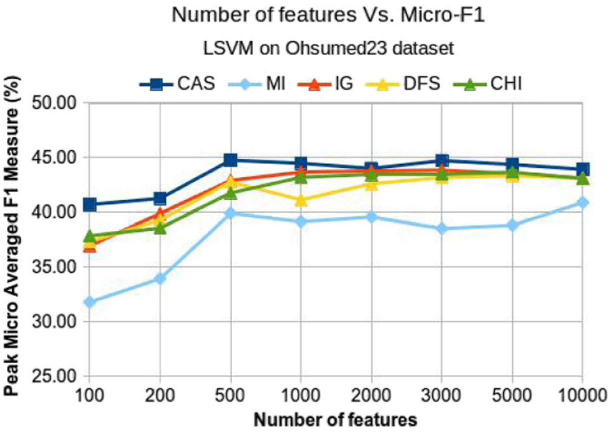

Fig. 17 Micro_F1 for LSVM in Ohsumed23 dataset

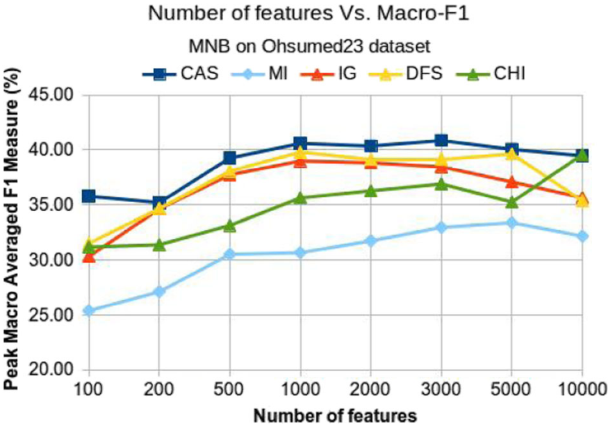

Fig. 18 Macro_F1 for MNB in Ohsumed23 dataset

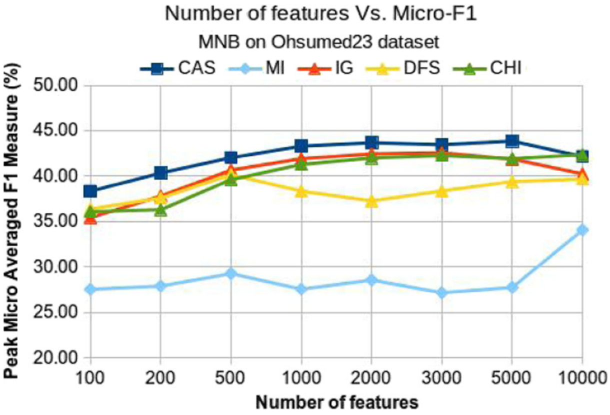

Fig. 19 Micro_F1 for MNB in Ohsumed23 dataset

Table 10 Average rank of methods

Dataset	CAS	MI	IG	DFS	CHI	CAS	MI	IG	DFS	CHI
	Rank of LSVM using Macro_F1					Rank of LASVM using Micro_F1				
Webkb	1	4.625	3.625	2.625	2.875	1	4.75	3.5	2.375	2.875
20Newsgroup	1.125	5	3.125	3.375	2.375	1.125	5	3.75	2.375	2.75
Ohsumed23	1.375	5	3.75	2.5	2.375	1	5	2.375	3.5	3.125
Ohsumed10	1.375	5	3.25	3.5	1.875	1.25	5	3.5	3.25	2
	Rank of MNB using Macro_F1					Rank of MNB using Micro_F1				
Webkb	1	5	3.75	2.375	2.875	1	5	2.875	2.75	3.75
20Newsgroup	1	5	3.625	2.375	3	1.25	5	3.625	2.5	2.375
Ohsumed23	1.125	5	3.125	2.25	3.5	1.125	5	2.625	3.375	2.875
Ohsumed10	1.25	5	4	2.875	1.875	1.25	4.875	4	3.125	1.75

methods is summarized in Table 11. The observed key points by analyzing the results of example dataset (see Table 2) are as follows:

1. **Strength of MI:** Whether the word is most frequent or less frequent, the MI assigns high weight to rare positive words.
   Weakness of MI: The MI assigns low weight to negative, common as well as sparse words. The distance among the weight of rare positive and negative words are not adequate, i.e. there is a low variance in the distance; e.g. in the weight of positive words viz. “toad”, “shark”, and “rays” as 0.25, 0.35, and 0.24 respectively. Similarly, the weight of negative nature words, i.e. “ostrich”, “emu”, and “mouse” are equal to 0.17, 0.21, and 0.20 respectively. The variance in the weight of common and sparse words with positive and negative words is also very low, e.g. “cow” and “cat” as 0.1 and 0.14 respectively.
2. **Strength of IG:** The IG assigns very less weight to common and sparse terms; e.g. “cow” and “cat” as 0.03 and 0.06 respectively.
   Weakness of IG: The IG assigns high weight to most frequent words, whether positive or negative, but medium weight to the less frequent positive and negative words.
3. **Strength of DFS:** The weight assignment process of DFS is similar to IG with few improvements. It assigns highest weight to the most frequent words in the range of 0.8 to 1. E.g. the positive words, “toad” and “turtle” get an equal weight 1, but IG assigns 4^{th} rank to the “turtle”. DFS differentiates common words better than MI, IG, and CHI. E.g. word, “cow” gets a higher rank than “cat”.

Table 11 Analysis of assigned weight by the methods in various ranges

Methods	Most frequent words				Less frequent words			
	Positive	Negative	Common	Sparse	Positive	Negative	Common	Sparse
CAS	Very high	Medium	Low	Low	Very high	Medium	Low	Low
MI	High	Low	Low	Low	High	Low	Low	Low
IG	High	High	Very low	Very low	Medium	Medium	Very low	Very low
DFS	Very high	Very high	Low	Low	Medium	Medium	Low	Low
CHI	Very high	Very High	low	Very low	Medium	Medium	Low	Very low

Weakness of DFS: The DFS assigns highest weight to the most frequent words, whether positive or negative, but medium weight to the less frequent positive and negative words. E.g. the negative word, “mouse” gets higher weight as 0.94 than the weight of positive word “rays” as 0.9.

4. **Strength of CHI:** The CHI assigns weight to the most frequent words in very high range than negative, common, and sparse words.
   Weakness of CHI: It doesn’t discriminate positive and negative nature of words proportionally. E.g. negative word, “mouse” gets higher weight as 17.7 than the weight of positive word “shark” as 7.67.
5. **Strength of CAS:** The CAS assigns highest weight to the words of a positive nature, medium weight to the words of a negative nature, and lower weight to the common/sparse words. The assigned weight of the words by the CAS method are more appropriate than other methods.
   Weakness of CAS: A larger set of informative words with higher weight are present in a group of classes. Whereas, a subset of the informative words with much smaller weight are present in a few classes. The CAS, MI, IG, DFS, and CHI methods select the top most, f features by sorting the words in descending order based on their weight. As a result, substantial words of a few classes are either partially or completely eliminated.

7 Conclusion

This paper introduced a new text feature selection method named *Correlative Association Score (CAS)*. The main objective of the CAS was to identify the nature of the words, i.e. positive, negative, common, or sparse. The words of negative nature for a class are also important to identify the class label of documents. The presence of negative nature words in the document assured that the document doesn’t belong to that class for which the word is negative. In this context, CAS combined the strength, likelihood and the association of words. It helped in identification of mutually associated words towards many classes. It has constructed an improved final feature set than state-of-the-art methods, viz. Mutual Information (MI), Information Gain (IG), Distinguishing Feature Selector (DFS), and Chi square (CHI). CAS assigned a much higher weight to the words of a positive and negative nature than common and sparse. The feature selection process was carried out under different conditions, i.e. feature set of varying sizes, dataset of diverse characteristics, classification algorithms, and success measures. The promising results based on Macro_F1 and Micro_F1 success measures proved the effectiveness of the proposed CAS method.

References

Agnihotri, D., Verma, K., & Tripathi, P. (2014). Pattern and cluster mining on text data. In *IEEE Computer Society, CSNT, Bhopal In Fourth International Conference on Communication Systems and Network Technologies* (pp. 428–432). doi:10.1109/CSNT.2014.92.

Agnihotri, D., Verma, K., & Tripathi, P. (2016). Computing symmetrical strength of n-grams: a two pass filtering approach in automatic classification of text documents. *SpringerPlus*, *5*(942), 1–29.

Agnihotri, D., Verma, K., & Tripathi, P. (2017). Variable global feature selection scheme for automatic classification of text documents. *Expert Systems with Applications, Elsevier*, *81*, 268–281. doi:10.1016/j.eswa.2017.03.057, http://www.sciencedirect.com/science/article/pii/S0957417417302208.

Dewang, R.K., & Singh, A.K. (2017). State-of-art approaches for review spammer detection: a survey. Journal of Intelligent Information Systems, 1–34. doi:10.1007/s10844-017-0454-7.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, *3*, 1289–1305.

Guo, H., Zhou, L.Z., & Feng, L. (2009). Self-switching classification framework for titled documents. *Journal Of Computer Science And Technology, Springer*, *24*(4), 615–625.

Joachims, T. (1996). A probabilistic analysis of the rocchio algorithm with tfidf for text classification. Technical Report CMU-CS-96-118, Department of Computer Science, Carnegie Mellon University.

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features, Springer Berlin, pp 137–142. doi:10.1007/BFb0026683.

Kevin, B., & Moshe, L. (2013). Uci machine learning repository. http://www.archiveicsuciedu/ml901.

Lamirel, J.C., Cuxac, P., Chivukula, A.S., & Hajlaoui, K. (2015). Optimizing text classification through efficient feature selection based on quality metric. *Journal of Intelligent Information Systems*, *45*(3), 379–396. doi:10.1007/s10844-014-0317-4.

Lewis, D.D., & Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval* (pp. 81–93). Las Vegas.

Manning, C.D., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval*. NY: Cambridge University Press.

Mitchell, T. (1997). Machine learning. McGraw Hill.

Mladenic, D., & Grobelnik, M. (1999). Feature selection for unbalanced class distribution and naive bayes. In *Proceeding of the 16th International Conference on Machine Learning* (pp. 258–267). SF.

Rehman, A., Kashif, J., Babri, H.A., & Mehreen, S. (2015). Relative discrimination criterion- a novel feature ranking method for text data. *Expert Systems with Applications, Elsevier*, *42*, 3670–3681.

Sebastiani, F. (2002). Machine learning in automated text classification. *ACM Computing Surveys*, *34*(1), 1–47.

Uysal, A.K., & Gunal, S. (2012). A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems, Elsevier*, *36*, 226–235.

Uysal, A.K., & Kursat, A. (2016). An improved global feature selection scheme for text classification. *Expert Systems with Applications, Elsevier*, *43*, 82–92.

Yang, Y., & Pedersen, J.O. (1997). A comparative study on feature selection in text classification. In *Proceedings of the 14th International Conference on Machine Learning* (pp. 412-420). USA.