Computing and Informatics, Vol. 39, 2020, 881–906, doi: 10.31577/cai_2020_5_881

OPTIMAL FEATURE SUBSET SELECTION BASED ON COMBINING DOCUMENT FREQUENCY AND TERM FREQUENCY FOR TEXT CLASSIFICATION

Thirumoorthy Karpagalingam, Muneeswaran Karuppaiah

Department of Computer Science and Engineering
Mepco Schlenk Engineering College, Sivakai
Tamilnadu, India
e-mail: {kthirumoorthy, kmuni}@mepcoeng.ac.in

Abstract. Feature selection plays a vital role to reduce the high dimension of the feature space in the text document classification problem. The dimension reduction of feature space reduces the computation cost and improves the text classification system accuracy. Hence, the identification of a proper subset of the significant features of the text corpus is needed to classify the data in less computational time with higher accuracy. In this proposed research, a novel feature selection method which combines the document frequency and the term frequency (FS-DFTF) is used to measure the significance of a term. The optimal feature subset which is selected by our proposed work is evaluated using Naive Bayes and Support Vector Machine classifier with various popular benchmark text corpus datasets. The experimental outcome confirms that the proposed method has a better classification accuracy when compared with other feature selection techniques.

Keywords: Feature selection, text classification, document frequency, term frequency

1 INTRODUCTION

Nowadays millions of million users contribute a huge amount of information in the form of unstructured text data: movie/product reviews and feedbacks, social media tweets, and personal blogs are stored in the WWW repository. The organization of those unstructured text documents is a challengeable task. Text classification is

used to organize those documents in a proper way and to extract the information from that unstructured text corpus. The text classification is a supervised learning algorithm, which uses the training data associated with class labels to assign the text document to the appropriate categories [25, 32]. Text classification is used in topic detection [4], spam e-mail filtering [14, 8], e-mail classification [10, 41], author identification [7, 34] and web page classification [2, 6]. The atomic element or indivisible unit of the text document set is called a feature or word or term. The text classification system uses the vector space model to represent the text. The text corpus D is a set of unstructured text documents and is denoted as $D = \{d_1, d_2, d_3, \ldots, d_n\}$. Features are extracted from the text corpus and are denoted as $T = [t_1, t_2, t_3, \ldots, t_m]$. Each document d_i, $1 < i < n$ is represented as a feature vector $\langle w_{i1}, w_{i2}, w_{i3}, \ldots, w_{im}\rangle$ where w_{ij} denotes the frequency of feature t_j appearing in document d_i [1, 23]. The typical Document Term Matrix (DTM) is generally of sparse in nature and is shown in Table 1.

Doc	t_1	t_2	...	t_i	...	t_m
d_1	w_{11}	w_{12}		w_{1i}		w_{1m}
d_2	w_{21}	w_{22}		w_{2i}		w_{2m}
$\vdots$						
d_n	w_{n1}	w_{n2}		w_{ni}		w_{nm}

Table 1. Document term matrix

The high dimension of the feature space may contain the uninformative/irrelevant features (noise features), which reduce the accuracy of the classification system [40]. The uninformative feature (noise feature) has no information about the category. For example, if a word appears in all the text documents in the text corpus, that word is not at all useful to predict the class label. In order to reduce the dimension of the feature space as well as to improve the accuracy in text classification problems, feature selection plays a vital role [5, 11, 12, 13, 17, 9, 16]. Let F be the feature set having 'f' number of features, then we can coin the $2^f - 1$ (except empty set) number of different subsets of features. If we work with all the subsets, it would increase the computing cost. Feature selection is a process to select the optimal best feature subset space from the original feature space [33]. There are two types of feature selection methods:

1. Filter-based and
2. Wrapper-based.

Filter based feature selection method uses the various scoring methodologies to assign the importance score to each feature, and top-N features are selected based on the relevance score. Filter based methods are independent of classification models. Computationally the filter methods are faster. The wrapper methods [15, 3, 39] are based on attribute subset selection. The wrapper-based methods depend on the

classification model and search algorithms. The hybrid feature selection [12, 36] method uses both filter-based and wrapper-based method.

The rest of the paper is organized as follows: Section 2 briefly describes various feature selection approaches. Section 3 focuses on the proposed feature selection method. The classifiers used in the experiments are discussed in Section 4. The experimental results from the various datasets are discussed in Section 5 followed by the concluding remarks in Section 6.

2 RELATED WORK

The primary goals of the feature selection methods are to select the most appropriate features and ignore the irrelevant and redundant features [31, 21, 22]. Many feature selection algorithms are proposed for feature selection in text classification [18, 19]. In this section, we will discuss the existing feature selection methods, including Document Frequency (DF), Balanced Accuracy (ACC2), Distinguishing Feature Selector (DFS), Normalized Difference Measure (NDM), and Mutual Information (MI), which all are based on document frequency, and Information Gain (IG) is based on entropy methods. The document frequency in the collection of various scenarios is described in the contingency Table 2. Let C_k and $\overline{C_k}$ be two different categories: k (positive class) and other than "k" categories (negative class). Also, let the term t_i and $\overline{t_i}$ represent the presence and absence of the given term, respectively.

Terms/Category	t_i (Presence of the Term)	$\overline{t_i}$ (Absence of the Term)
C_K (belongs to)	tp	fn
$\overline{C_K}$ (does not belong to)	fp	tn

Table 2. Contingency table

tp: true positive count denotes the number of documents, which contain the term t_i and those documents belong to the category C_k (positive class)

fn: false negative count denotes the number of documents, which do not contain the term t_i and those documents belong to the category C_k (positive class)

fp: false positive count denotes the number of documents, which contain the term t_i and those documents do not belong to the category C_k (negative class)

tn: true negative count denotes the number of documents, which do not contain the term t_i and those documents do not belong to the category C_k (negative class)

The brief summary of preliminary notations, which are used in this work, is shown in Table 3.

Notation	Values	Description
K		number of Category in the dataset
N	$\mathrm{tp}+\mathrm{fn}+\mathrm{fp}+\mathrm{tn}$	number of documents in dataset
N_i	$\mathrm{tp}+\mathrm{fp}$	number of documents containing term t_i
N_k	$\mathrm{tp}+\mathrm{fn}$	number of documents belonging to category C_k
N_{ik}	tp	number of documents containing term t_i and belonging to C_k
$P(t_i)$	$\frac{N_i}{N}$	probabilities of presence of the term t_i
$\mathrm{P}(\overline{t_i})$	$\frac{(N-N_i)}{N}$	probabilities of absence of the term t_i
$P(C_k)$	$\frac{N_k}{N}$	probability of class C_k
$P(C_k\|t_i)$	$\frac{N_{ik}}{N_i}$	conditional probabilities of class C_k given presence of term t_i
$P(C_k\|\overline{t_i})$	$\frac{fn}{(N-N_i)}$	conditional probabilities of class C_k given absence of term t_i
$P(t_i\|C_k)$	$\frac{N_{ik}}{N_k}$	conditional probabilities of the presence of term t_i given class C_k
$P(t_i\|\overline{C_K})$	$\frac{fp}{(N-N_k)}$	conditional probability of the presence of term t_i given class other than C_k
$P(\overline{t_i}\|C_k)$	$\frac{fn}{N_k}$	conditional probability of absence of term t_i given class C_k
tf_{ij}		term frequency (occurrence) of term t_i in the document d_j
tf_i	$\sum_{j=1}^{N} tf_{ij}$	term frequency (occurrence) of term t_i in the entire dataset
tf_{ik}		term frequency (occurrence) of term t_i in the Category C_k
μ_i	$\frac{tf_i}{N}$	mean term frequency of term t_i in the entire dataset
μ_{ik}	$\frac{tf_{ik}}{N_k}$	mean term frequency of term t_i in the Category C_k
σ_i	$\sqrt{\frac{\sum_{j=1}^{N}(tf_{ij}-\mu_i)^2}{N}}$	standard deviation of term frequency of term t_i in the entire dataset
σ_{ik}	see Equation (13)	standard deviation of term frequency of term t_i in the Category C_k

Table 3. Preliminary notation

2.1 Document Frequency (DF)

Document Frequency (DF) [26, 38, 28] of the term 't' is the simplest feature selection method and is based on the number of documents containing term 't'. The DF method considers that the rare frequency terms are non-informative for text categorization. It ignores the impact on category information. This method considers the impact on the positive class and ignores the negative class. The document frequency

of the term 't' is calculated as follows:

$$\mathrm{DF}(t) = \mathrm{tp} + \mathrm{fp}. \tag{1}$$

2.2 Balanced Accuracy (ACC2)

Accuracy is one of the feature ranking metric, which considers the difference between true positives and false positives of a term, and it supports strong positive features. Balanced Accuracy (ACC2) [11, 30] is a variant of accuracy which ranks the features based on the absolute difference of the true positive rate (tpr) and false positive rate (fpr). The tpr and fpr of the term 't' are calculated as follows:

$$\mathrm{tpr}(t) = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}}, \tag{2}$$

$$\mathrm{fpr}(t) = \frac{\mathrm{fp}}{\mathrm{tn} + \mathrm{fp}}. \tag{3}$$

Balanced Accuracy ignores the influence of term frequency, which uses the absolute difference between tpr and fpr as follows:

$$\mathrm{ACC2}(t) = |\mathrm{tpr}(t) - \mathrm{fpr}(t)|. \tag{4}$$

2.3 Information Gain (IG)

Information Gain (IG) [28, 24] is an entropy-based evaluation technique, which is used in machine learning applications. It refers to the difference between the information entropy produced by the presence and absence of a term in the document. The expression for IG is as follows:

$$\begin{aligned} IG(t_i) = &- \sum_{k=1}^{K} P(C_k) \log P(C_k) \\ &+ P(t_i) \sum_{k=1}^{K} P(C_k|t_i) \log P(C_k|t_i) \\ &+ P(\overline{t_i}) \sum_{k=1}^{K} P(C_k|\overline{t_i}) \log P(C_k|\overline{t_i}). \end{aligned} \tag{5}$$

2.4 Mutual Information (MI)

Mutual Information (MI) [28, 27] is a measure between two random variables, which quantifies the amount of information gained about one variable through another

variable. The computation of mutual information for a given term t_i is given as

$$MI(t_i, C_k) = \log_2 \frac{P(t_i|C_k)}{P(t_i)}, \tag{6}$$

$$MI(t_i) = \sum_{k=1}^{K} P(C_k) * MI(t_i, C_k). \tag{7}$$

2.5 Distinguishing Feature Selector

Uysal and Gunal [37] proposed the probabilistic based feature selection scheme as a Distinguishing Feature Selector (DFS). DFS assigns the high score to the term, which frequently occurs in one of the categories and does not occur in the other categories, is distinctive; A term which frequently occurs in all the class is irrelevant; it must be assigned with a low score. DFS ignores the significance of term frequency information in their scoring mechanism. DFS can be formulated

$$\mathrm{DFS}(t_i) = \sum_{k=1}^{K} \frac{P(C_k|t_i)}{1 + P(\overline{t_i}|C_k) + P(t_i|\overline{C_k})}. \tag{8}$$

DFS score of the feature lies between 0.5 and 1.0. The most discriminating terms have an importance score that is close to 1.0 and the least discriminating terms are assigned with the significance score is 0.5.

2.6 Normalized Difference Measure (NDM)

NDM algorithm [30] uses the ACC2 value divided by the minimum of tpr and fpr to indicate the importance of the feature. The value of NDM is estimated as follows:

$$\mathrm{NDM}(t) = \frac{|\mathrm{tpr}(t) - \mathrm{fpr}(t)|}{\min(\mathrm{tpr}(t), \mathrm{fpr}(t))}. \tag{9}$$

If $\min(\mathrm{tpr}, \mathrm{fpr}) = 0$, then to avoid divide by zero exception, it is replaced by small value. NDM method ignores the influence of term frequency. In addition, NDM equally ranks the terms having equal weight value regardless of the value of $|\mathrm{tpr} - \mathrm{fpr}|$.

All of the above-mentioned feature selection schemes ignore the significance of the term frequency. To illustrate the significance of term frequency, a sample data is shown in 4, which consists of 10 documents, grouped under 3 categories {C1, C2, C3}. There are seven distinct terms from the 10 documents and they are: {lion, tiger, bear, goat, deer, horse, panda}. The following handwork experiments on the sample dataset shows the significance of the term frequency.

Table 4 shows the Document Term Matrix (DTM) of the sample dataset.

Table 5 shows the importance score of above mentioned selection scheme.

List of issues for ignoring the term frequency:

Documents	**Category**	**lion**	**bear**	**tiger**	**goat**	**deer**	**horse**	**panda**
d_1	C_1	1	10	10	3	15	30	25
d_2	C_1	1	10	10	5	15	20	10
d_3	C_1	1	10	10	4	0	40	2
d_4	C_2	1	10	1	10	0	0	0
d_5	C_2	1	10	1	10	0	0	0
d_6	C_2	1	10	1	10	20	0	0
d_7	C_2	1	10	1	10	20	0	0
d_8	C_3	1	10	1	1	3	0	0
d_9	C_3	1	10	1	1	0	0	0
d_{10}	C_3	1	10	1	1	0	0	0

Table 4. Document term matrix of the sample dataset

Documents	**lion**	**bear**	**tiger**	**goat**	**deer**	**horse**	**panda**
DF	10	10	10	10	5	3	3
ACC2	0	0	0	0	0.142	0.628	0.628
IG	0	0	0	0	0.049	0.881	0.881
MI	0	0	0	0	−0.05	0.52	0.52
DFS	0.5	0.5	0.5	0.5	0.516	1.0	1.0
NDM	0	0	0	0	0.38	6.286	6.286

Table 5. Importance scores of the term in sample dataset

- The terms 'lion', 'tiger', 'bear', 'goat' appeared in all the documents. The existing feature selection method ACC2, NDM, IG and MI consider these terms are useless. And DFS says that all the four (lion, tiger, bear, goat) are the same. While comparing lion and tiger, tiger may contribute more to the category C1 also goat may contribute more to category C2 based on term frequency. So we cannot provide the equal weightage to these terms.
- The terms 'horse' and 'panda' only appeared in the category C1. ACC2, NDM, IG, DFS provide equal weightage. Based on the term frequency horse must have highest significance score.

In order to address these problems, we combine the document frequency and term frequency information to select the optimal feature subset.

3 PROPOSED WORK

The feature selection method assigns high significant scores to the more informative features and lower significant scores to less informative/irrelevant features. The above mentioned feature selection scheme ranks the feature based on how the term contributes to the categorization based on document frequency. We propose the new feature selection scheme which integrates the document frequency contribution and term frequency contribution. The proposed feature selection method FS-DFTF

assigns the significance score based on the following:

- FS-DFTF assigns a high significance score to a term which frequently occurs in a single category and does not occur in the other category, it is an informative term.
- FS-DFTF assigns a low significance score to a term which frequently occurs in all the categories, it is irrelevant feature.

We consider the term frequency distribution in two levels: i) the frequency of the term between the category level, and ii) the frequency of the term within the category level. The system design of our proposed work is depicted in Figure 1.

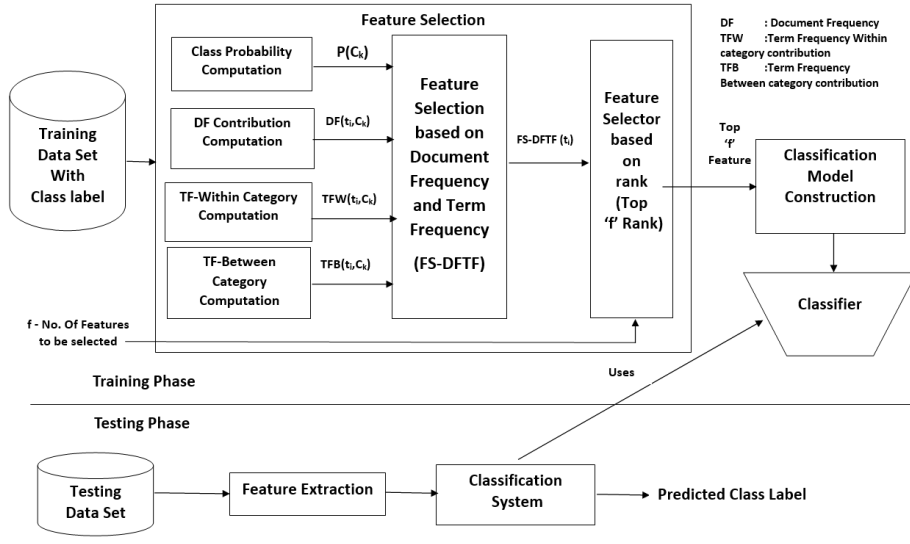


Figure 1. The architectural framework for text classification system employing FS-DFTF

In order to integrate the term frequency information, we are using the standard deviation of term frequency and mean of term frequency at both levels (between category level and within category level). The most informative feature must have high mean frequency and less standard deviation of term frequency. Based on this, we formulate the FS-DFTF as follows:

$$\text{FS} - \text{DFTF}(t_i) = \sum_{k=1}^{K} (P(C_k) * \theta(t_i, C_k) * \Phi(t_i, C_k) * \Psi(t_i, C_k)) \tag{10}$$

where $\theta(t_i, C_k)$ indicates the Document Frequency contribution of the term t_i in category C_k, $\Phi(t_i, C_k)$ indicates term frequency contribution between the category level, $\Psi(t_i, C_k)$ indicates term frequency contribution within the category level.

Document frequency contribution can be computed as follows:

$$\theta(t_i, C_k) = \frac{P(C_k|t_i) - P(C_k|\overline{t_i})}{1 + P(\overline{t_i}|C_k) + P(t_i|\overline{C_k})}. \tag{11}$$

In this work, we consider $P(C_k|\overline{t_i}) = 0$ if $P(\overline{t_i}) = 0$, to avoid division by zero errors. The computed document frequency contribution $\theta(t_i, C_k)$ over the sample dataset is shown in Table 6.

	$P(C_k\|t_i)$			$P(C_k\|\overline{t_i})$			$P(\overline{t_i}\|C_k)$			$P(t_i\|\overline{C_k})$			$\theta(t_i, C_k)$		
Term	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3
lion	0.3	0.4	0.3	0	0	0	0	0	0	1	1	1	0.15	0.2	0.15
bear	0.3	0.4	0.3	0	0	0	0	0	0	1	1	1	0.15	0.2	0.15
tiger	0.3	0.4	0.3	0	0	0	0	0	0	1	1	1	0.15	0.2	0.15
goat	0.3	0.4	0.3	0	0	0	0	0	0	1	1	1	0.15	0.2	0.15
deer	0.399	0.4	0.199	0.199	0.4	0.399	0.333	0.5	0.666	0.429	0.5	0.571	0.114	0	−0.089
horse	1	0	0	0	0.571	0.429	0	1	1	0	0.5	0.429	1	−0.229	−0.176
panda	1	0	0	0	0.571	0.429	0	1	1	0	0.5	0.429	1	−0.229	−0.176

Table 6. Document frequency contribution $\theta(t_i, C_k)$ over the sample dataset

Term frequency contribution between the category levels can be computed as follows:

$$\Phi(t_i, C_k) = \begin{cases} \frac{N_{ik}}{N_i} * \frac{tf_{ik}}{tf_i}, & \sigma_i > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

If the standard deviation term frequency of the term t_i is zero ($\sigma_i = 0$) then the term t_i appears in all the document with equal term frequency. For the classification task, that term t_i is not useful. Hence, we can say that $\Phi(t_i, C_k) = 0$. The computed term frequency contribution between the category levels $\Phi(t_i, C_k)$ over the sample dataset is shown in Table 7.

	N_i	N_{ik}			$\mathbf{tf}_i$	$\mathbf{tf}_{ik}$			σ_i	$\Phi(t_i, C_k)$		
Term		C_1	C_2	C_3		C_1	C_2	C_3		C_1	C_2	C_3
lion	10	3	4	3	10	3	4	3	0	0	0	0
bear	10	3	4	3	100	30	40	30	0	0	0	0
tiger	10	3	4	3	37	30	4	3	4.35	0.243	0.043	0.024
goat	10	3	4	3	55	12	40	3	4.09	0.065	0.290	0.016
deer	5	2	2	1	73	30	40	3	8.98	0.164	0.219	0.008
horse	3	3	0	0	90	90	0	0	15.24	1.0	0	0
panda	3	3	0	0	37	37	0	0	8.11	1.0	0	0

Table 7. Term frequency contribution between the category level $\Phi(t_i, C_k)$ over the sample dataset

The Term Frequency contribution within the category level can be computed as follows:

$$\Psi(t_i, C_k) = \frac{1}{\sigma_{ik}} * \left[\frac{\frac{N_{ik}}{N_k} * \mu_{ik}}{\max_{\{j=1,2,\ldots,m\}} \left\{ \frac{N_{jk}}{N_k} * \mu_{jk} \right\}} \right] \tag{13}$$

where $\sigma_{ik} = \sqrt{\frac{\sum_{j=1}^{N} (tf_{ij} - \mu_{ik})^2 * I_{jk}}{N_k}}$, $I_{jk} = \begin{cases} 1, & d_j \in C_k \\ 0, & \text{otherwise} \end{cases}$, m is the number of distinct terms present in the Category C_k. This part represents, how much the term frequency of term t_i contributes to the category C_k while comparing other terms within a category. If σ_{ik} is zero then the term t_i appears in all the document of category C_k with equal term frequency. In that case, we may ignore the σ_{ik} component of feature t_i. So,

$$\Psi(t_i, C_k) = \frac{\frac{N_{ik}}{N_k} * \mu_{ik}}{\max_{\{j=1,2,\ldots,m\}} \left\{ \frac{N_{jk}}{N_k} * \mu_{jk} \right\}}, \text{ if } \sigma_{ik} = 0. \tag{14}$$

If the standard deviation of term frequency of the term is zero ($\sigma_i = 0$) then the term t_i present in all the document with equal term frequency. That term is a not useful for classification task. So we can say that $\Phi(t_i, C_k) = 0$ and $\Psi(t_i, C_k) = 0$. The computed term frequency contribution within the category levels $\Psi(t_i, C_k)$ over the sample dataset is shown in Table 8.

	N_{ik}			μ_{ik}			σ_{ik}			$\Psi(t_i, C_k)$		
Term	**C1**	**C2**	**C3**	**C1**	**C2**	**C3**	**C1**	**C2**	**C3**	**C1**	**C2**	**C3**
lion	3	4	3	1	1	1	0	0	0	0	0	0
bear	3	4	3	10	10	10	0	0	0	0	0	0
tiger	3	4	3	10	1	1	0	0	0	0.333	0.1	0.1
goat	3	4	3	4	10	1	1	0	0	0.13	1	0.1
deer	2	2	1	10	10	1	8.660	11.547	1.732	0.222	0.5	0.033
horse	3	0	0	30	0	0	10	0	0	1	0	0
panda	3	0	0	12.333	0	0	11.676	0	0	0.411	0	0

Table 8. Term frequency contribution within the category level $\Psi(t_i, C_k)$ over the sample dataset

Finally, the FS-DFTF score of given sample datadet is shown in Table 9.

Term	**lion**	**bear**	**tiger**	**goat**	**deer**	**horse**	**panda**
FS-DFTF	0	0	0.004	0.023	0.001	0.3	0.123
FS-DFTF Rank	6	6	4	3	5	1	2

Table 9. FS-DFTF score of the sample dataset

In order to show the working principles of FS-DFTF, the sample dataset and related results are shown in the above mentioned Tables 4, 5, 6, 7, 8, 9. The real

performance of FS-DFTF on the popular benchmark datasets are presented briefly in the experimental work section.

3.1 The Pseudo Procedure for the FS-DFTF

Algorithm 1 describes the process of proposed work.

Algorithm 1: feature selection using FS-DFTF

Input:
D: Dataset with class label
f : Number of features to be selected
Output: F_{best}: selected f features

```
1  D ← textPreProcessing(D)
2  T : [t_1, t_2, t_3, ..., t_m] ← Tokenizer(D) // m-numbers of unique features in D
3  for each category C_k in D
4     P(C_k) = computeClassProbability(D)
5  end
6  for each term t_i in T
7     for each category C_k in D
8        θ(t_i, C_k) = computeDFContribution(D) // use eqn.(10)
9        Φ(t_i, C_k) = computeTFContributionBetweenCategoryLevel(D) // use
       eqn.(11)
10       Ψ(t_i, C_k) = computeTFContributionWithinCategoryLevel(D) // use
       eqn.(12)
11    end
12    FS_DFTF(t_i) = computeFSDFTF() //use eqn.(13)
13 end
14 F_best = SortAndSelect(FS_DFTF, f)
15 return   F_best
```

Line 1 shows the preprocessing. The preprocessing steps are lower casings, removing punctuations, numbers, stop words and finally stemming. Line 2 tokenize the entire dataset to find the unique feature list. Lines 3–13 compute the FS-DFTF score of each term t_i in the feature list. Line 14 sorts the feature based on FS-DFTF importance score value and selects the top 'f' feature.

4 DATASET AND EXPERIMENTAL SETUP

In this section, we briefly discuss the datasets, classification algorithm and the evaluation metric to evaluate the performance of the proposed feature selection measure for text classification.

4.1 Dataset

In this work, we have experimented with four distinct datasets (WebKB, SMS, BBC News and 10Newsgroups) used for the assessment of our proposed feature selection method [dataset[1]]. WebKB is a collection of web pages collected by the World Wide Knowledge Base. These pages were collected from computer science departments of various universities in 1997. The web pages are classified as various classes: student, faculty, staff, department, course, project, and other. For the experimental works we chose the class label (project, course, faculty, student) documents. Table 10 describes the properties of WebKB dataset. SMS Dataset is a collection of SMS, which is labeled as Spam or Ham. It contains 5 574 labeled SMS message. Description of the SMS dataset is shown in Table 11. BBC Dataset contains 2 225 text documents from the BBC news website, which are classified into five categories (business, entertainment, politics, sport, tech), Table 12 shows the document distribution of BBC dataset. Newsgroups Dataset: The documents of Newsgroups dataset contains approximately 15 000 news documents, which are manually classified into 20 groups. In this work we have experimented with ten categories. News document distribution among the selected categories is shown in Table 13.

S. No	Category	Training Docs	Testing Docs	Total Docs
1	project	336	168	504
2	course	620	310	930
3	faculty	750	374	1 124
4	student	1 097	544	1 641
	Total	2 803	1 396	4 199

Table 10. WebKB dataset

S. No	Category	Training Docs	Testing Docs	Total Docs
1	Spam	436	311	747
2	Ham	2 838	1 989	4 827
	Total	3 274	2 300	5 574

Table 11. SMS dataset

4.2 Classifier Used

In this proposed research, we have used the Naive Bayes (NB) and Support Vector Machine classifiers to classify the unstructured documents. The primary idea of NB classifier [5, 31, 35] is to use the joint probabilities of the terms and class labels of

[1] dataset: `http://ana.cachopo.org/datasets-for-single-label-text-categorization`

S. No	Category	Training Docs	Testing Docs	Total Docs
1	Business	281	229	510
2	Entertainment	223	163	386
3	Politics	219	198	417
4	Sport	292	219	511
5	Tech	210	191	401
	Total	1 225	1 000	2 225

Table 12. BBC dataset

S. No	Category	Training Docs	Testing Docs	Total Docs
1	rec. autos	594	395	989
2	rec. motorcycles	598	398	996
3	rec. sport. baseball	597	397	994
4	rec. sport. hockey	600	399	999
5	sci. crypt	595	396	991
6	sci. electronics	591	393	984
7	sci. med	594	396	990
8	sci. space	593	394	987
9	soc. religion. christian	598	398	996
10	talk. politics. guns	545	364	909
	Total	5 905	3 930	9 835

Table 13. 10Newsgroup dataset

the training set to determine the class label of a given unknown document. Given the document d_j, the probability with each category C_k is computed as follows:

$$P(C_k|d_j) = \frac{P(d_j|C_k)}{P(d_j)} * P(C_k). \tag{15}$$

The class label for the document d_j, can be evaluated by,

$$\text{Label}(d_j) = \max_{k=1,2,\ldots,K} \{P(C_k|d_j)\}. \tag{16}$$

Support vector machine (SVM) is one of the best supervised learning algorithm which is used for classification and regression [10, 18, 19, 22, 29, 20]. SVM finds a hyper-plane or set of hyper-planes to separate the classes in the high dimensional space. The main objective of SVM is to find the decision boundary that is maximally away from any data point. SVM classifier finds the maximum margin hyper plane, which separate the two classes and the border of hyper-plane is defined by support vectors.

4.3 Evaluation Metric

There are four standard evaluation measures namely Accuracy, Precision, Recall, F-Score that are used to evaluate the proposed feature selection model for classifying the unstructured documents [28]. True positives (TP) are instances when the actual category of the tuple was positive and the predicted is also positive. True negatives (TN) are instances when the actual category of the tuple was negative and predicted is also negative. False positives (FP) are instances when the actual category of the tuple was negative and predicted is positive. False negatives (FN) are instances when the actual category of the tuple was positive and predicted as negative. The accuracy of a classification model is calculated as how much percentage of testing dataset documents are accurately classified.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \tag{17}$$

The Precision (P) is a measure of exactness. It refers what percentage of tuples classified as positive are actually positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{18}$$

The Recall (R) is a measure of completeness. It refers what percentage of positive tuples are classified as positive are actually positive.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{19}$$

The F-Score (F) is defined as the weighted harmonic mean of the Precision and Recall.

$$F = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{20}$$

5 RESULTS

In this section, a deep analysis is carried out to compare the FS-DFTF feature selection scheme against various filter based feature selection methods (DF, ACC2, IG, MI, DFS, and NDM) in terms of classifier accuracy, precision, recall and F-score. We have taken the measurement of experiments using a computer equipped with 2.30 GHz, Intel Core i5 processor and 8 GB RAM memory. The experiments are conducted with features of varying sizes such as 10, 50, 100, 200, 300, 500 and 1 000. We have used Python 3.7.3 for programming and matplotlib library to plot the performance graph. The performance of the proposed work FS-DFTF is shown in Figures 2, 3, 4, 5 and Tables 14, 15, 16, 17, 18, 19, 20, 21 on the above mentioned dataset respectively. In all the graphs, the X-axis represents the number of selected features and the Y-axis represents the corresponding classifier performance

in terms of accuracy. In most experimental results, the proposed method shows better accuracy than other contrast ones.

5.1 Performance Comparisons on the WebKB Dataset

The accuracy of the NB classifier and SVM classifier on the WebKB dataset are shown in Figures 2 a) and 2 b), respectively. According to Figure 2, the proposed FS-DFTF method surpasses the individual performance of all other methods in terms of accuracy using Naive Bayes classifier and SVM classifier. For the WebKB dataset, the optimum feature size is 200 with an accuracy of 89.18 % for NB classifier and 87.97 % for SVM classifiers. We observe that the accuracy curve of FS-DFTF is higher than that of other methods for both Naive Bayes and linear SVM classifiers.

Table 14 shows the Precision, Recall and F-Score of Naive Bayes classifier using FS-DFTF, MI, DFS, DF, ACC2, IG and NDM on the Webkb dataset when top 200 features are selected in feature space. The results show that FS-DFTF method has a higher number of instances correctly classified (1 245 instances over 1 396) than the six existing techniques and it improves the classification performance.

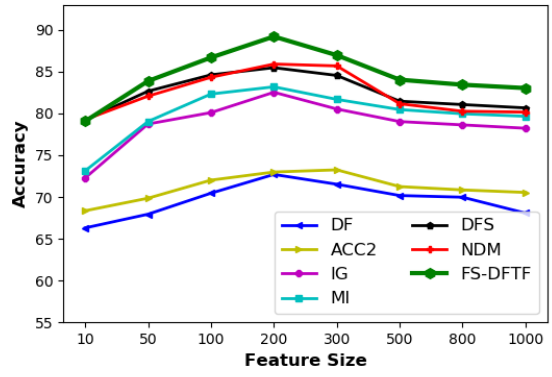
Algorithm	**Precision**	**Recall**	**F_1 Score**	**Accuracy in % (Correctly Classified Docs)**	**Error Rate in % (Incorrectly Classified Docs)**
DF	0.74	0.73	0.73	72.7 % (1 015)	27.3 % (381)
ACC2	0.74	0.73	0.73	72.99 % (1 019)	27.01 % (377)
IG	0.83	0.83	0.83	82.52 % (1 152)	17.48 % (244)
MI	0.84	0.83	0.83	83.17 % (1 161)	16.83 % (235)
DFS	0.86	0.85	0.86	85.46 % (1 193)	14.54 % (203)
NDM	0.86	0.86	0.86	85.89 % (1 199)	14.11 % (197)
FS-DETF	0.89	0.89	0.89	89.18 % (1 245)	10.82 % (151)

Table 14. Performance of FS-DFTF on WebKB dataset using NB classifier

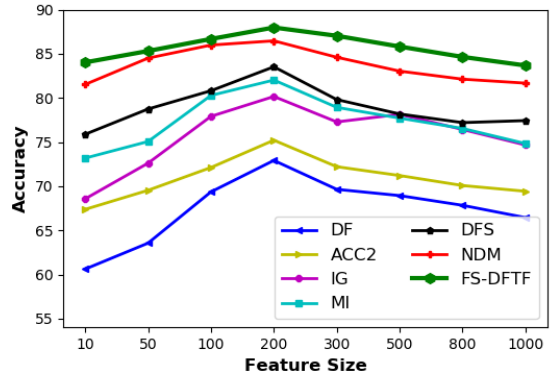
Table 15 shows the Precision, Recall and F-Score of SVM classifier using FS-DFTF, MI, DFS, DF, ACC2, IG and NDM on the Webkb dataset when top 200 features are selected in feature space. The results show that FS-DFTF method has a higher number of instances correctly classified (1 228 instances over 1 396) than the six existing techniques and it improves the classification performance.

5.2 Performance Comparisons on the SMS Dataset

Figure 3 shows the experimental results of text classification on the SMS dataset using NB and SVM classifiers. The curves in the figures indicate the various feature selection scheme. It can be seen from Figure 3 a) that the performance of FS-DFTF using the NB classifier is better than all other feature selection scheme. Also, Figure 3 b) shows that the performance of the proposed work using the SVM classifier has the highest accuracy while comparing other feature selection scheme. For the



a) NB classifier



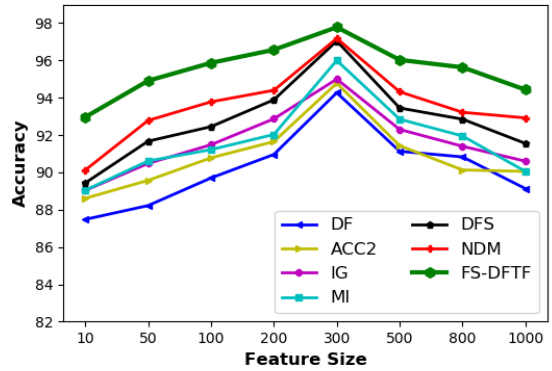
b) SVM classifier

Figure 2. Accuracy comparison for WebKb dataset using a) NB classifier b) SVM classifier

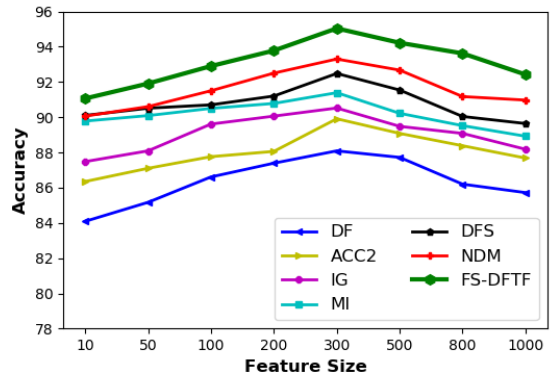
Algorithm	**Precision**	**Recall**	**F_1 Score**	**Accuracy in % (Correctly Classified Docs)**	**Error Rate in % (Incorrectly Classified Docs)**
DF	0.74	0.73	0.73	72.92 % (1 018)	27.08 % (378)
ACC2	0.77	0.75	0.76	75.21 % (1 050)	24.79 % (346)
IG	0.81	0.8	0.8	80.16 % (1 119)	19.84 % (277)
MI	0.83	0.82	0.82	82.02 % (1 145)	17.98 % (251)
DFS	0.84	0.84	0.84	83.52 % (1 166)	16.48 % (194)
NDM	0.87	0.86	0.87	86.46 % (1 207)	13.54 % (189)
FS-DETF	0.88	0.88	0.88	87.97 % (1 228)	12.03 % (168)

Table 15. Performance of FS-DFTF on WebKB dataset using SVM classifier

SMS dataset, the optimum feature size is 300 with an accuracy of 97.78 % for NB classifier and 95.04 % for SVM classifiers.



a) NB classifier



b) SVM classifier

Figure 3. Accuracy comparison for SMS dataset using a) NB classifier b) SVM classifier

While selecting top 300 features in feature space, the Precision, Recall and F-Score of Naive Bayes classifier using FS-DFTF, MI, DFS, DF, ACC2, IG and NDM on the Webkb dataset is shown in Table 16. The results show that FS-DFTF method has a higher number of instances correctly classified (2 249 instances over 2 300) than the six existing techniques and it improves the classification performance.

Table 17 shows the Precision, Recall and F-Score of SVM classifier using FS-DFTF, MI, DFS, DF, ACC2, IG and NDM on the SMS dataset when top 300 features are selected in feature space. The results show that FS-DFTF method has

Algorithm	Precision	Recall	F_1 Score	Accuracy in % (Correctly Classified Docs)	Error Rate in % (Incorrectly Classified Docs)
DF	0.95	0.94	0.95	94.27 % (2 168)	5.74 % (132)
ACC2	0.96	0.95	0.95	94.78 % (2 180)	5.21 % (120)
IG	0.96	0.95	0.95	95.0 % (2 185)	5.0 % (115)
MI	0.97	0.96	0.96	96.0 % (2 208)	4.0 % (92)
DFS	0.97	0.97	0.97	97.04 % (2 232)	2.96 % (68)
NDM	0.97	0.97	0.97	97.18 % (2 235)	2.82 % (65)
FS-DETF	0.98	0.98	0.98	97.78 % (2 249)	2.22 % (51)

Table 16. Performance of FS-DFTF on SMS dataset using NB classifier

a higher number of instances correctly classified (2 186 instances over 2 300) than the six existing techniques and it improves the classification performance.

Algorithm	Precision	Recall	F_1 Score	Accuracy in % (Correctly Classified Docs)	Error Rate in % (Incorrectly Classified Docs)
DF	0.92	0.88	0.89	88.09 % (2 026)	11.91 % (274)
ACC2	0.93	0.90	0.91	89.91 % (2 068)	10.09 % (232)
IG	0.93	0.91	0.91	90.52 % (2 082)	9.48 % (218)
MI	0.94	0.91	0.92	91.39 % (2 102)	8.61 % (198)
DFS	0.94	0.92	0.93	92.48 % (2 127)	7.52 % (173)
NDM	0.95	0.93	0.94	93.3 % (2 146)	6.7 % (154)
FS-DETF	0.96	0.95	0.95	95.04 % (2 186)	4.96 % (114)

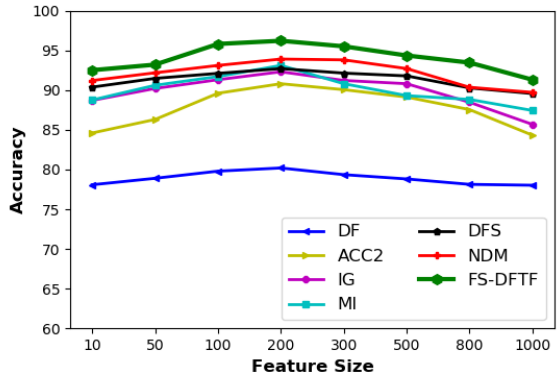
Table 17. Performance of FS-DFTF on SMS dataset using SVM classifier

5.3 Performance Comparisons on the BBC Dataset

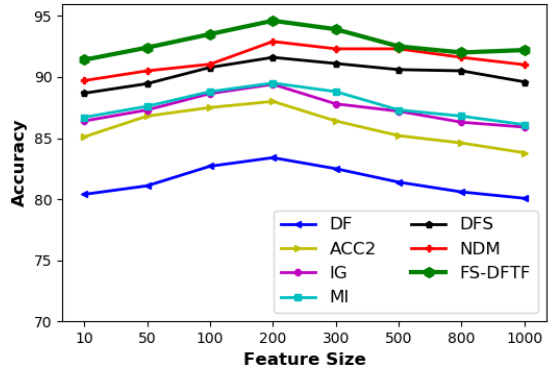
The accuracy of the NB classifier and SVM classifier on the BBC dataset are shown in Figures 4 a) and 4 b), respectively. According to Figure 4, the proposed FS-DFTF method surpasses the individual performance of all other methods in terms of accuracy using Naive Bayes classifier and SVM classifier. For the BBC dataset, the optimum feature size is 200 with an accuracy of 96.2 % for NB classifier and 94.6 % for SVM classifiers. We observe that the accuracy curve of FS-DFTF is higher than that of other methods for both Naive Bayes and linear SVM classifiers.

Table 18 shows the Precision, Recall and F-Score of Naive Bayes classifier using FS-DFTF, MI, DFS, DF, ACC2, IG and NDM on the BBC dataset when top 200 features are selected in feature space. The results show that FS-DFTF method has a higher number of instances correctly classified (962 instances over 1 000) than the six existing techniques and it improves the classification performance.

Table 19 shows the Precision, Recall and F-Score of SVM classifier using FS-DFTF, MI, DFS, DF, ACC2, IG and NDM on the BBC dataset when top 200 fea-



a) NB classifier



b) SVM classifier

Figure 4. Accuracy comparison for BBC dataset using a) NB classifier b) SVM classifier

Algorithm	**Precision**	**Recall**	**F_1 Score**	**Accuracy in % (Correctly Classified Docs)**	**Error Rate in % (Incorrectly Classified Docs)**
DF	0.8	0.8	0.8	80.2 % (802)	19.8 % (19.8)
ACC2	0.91	0.91	0.91	90.8 % (908)	9.2 % (92)
IG	0.92	0.92	0.92	92.3 % (92.3)	7.7 % (77)
MI	0.93	0.93	0.93	93.1 % (931)	6.9 % (69)
DFS	0.93	0.93	0.93	92.7 % (927)	7.3 % (73)
NDM	0.94	0.94	0.94	93.9 % (939)	6.1 % (61)
FS-DETF	0.96	0.96	0.96	96.2 % (962)	3.8 % (38)

Table 18. Performance of FS-DFTF on BBC News corpus using NB classifier

tures are selected in feature space. The results show that FS-DFTF method has a higher number of instances correctly classified (946 instances over 1 000) than the six existing techniques and it improves the classification performance.

Algorithm	Precision	Recall	F_1 Score	Accuracy in % (Correctly Classified Docs)	Error Rate in % (Incorrectly Classified Docs)
DF	0.83	0.83	0.83	83.4 % (834)	16.6 % (166)
ACC2	0.88	0.88	0.88	88.0 % (880)	12.0 % (120)
IG	0.89	0.89	0.89	89.4 % (894)	10.6 % (106)
MI	0.9	0.9	0.9	89.5 % (895)	10.5 % (105)
DFS	0.92	0.92	0.92	91.6 % (916)	8.4 % (84)
NDM	0.93	0.93	0.93	92.9 % (929)	7.1 % (71)
FS-DETF	0.95	0.95	0.95	94.6 % (946)	5.4 % (54)

Table 19. Performance of FS-DFTF on BBC News corpus using SVM classifier

5.4 Performance Comparisons on the 10Newsgroup Dataset

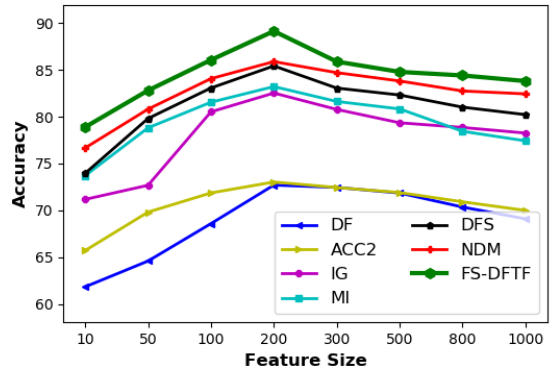
Figure 5 shows the experimental results of text classification on the 10Newsgroup dataset using NB and SVM classifiers. The curves in the figures indicate the various feature selection scheme. It can be seen from Figure 5 a) that the performance of FS-DFTF using the NB classifier is better than all other feature selection scheme. Also, Figure 5 b) shows that the performance of the proposed work using the SVM classifier has the highest accuracy while comparing other feature selection scheme. For the SMS dataset, the optimum feature size is 300 with an accuracy of 89.16 % for NB classifier and 86.77 % for SVM classifiers.

While selecting top 200 features in feature space, the Precision, Recall and F-Score of Naive Bayes classifier using FS-DFTF, MI, DFS, DF, ACC2, IG and NDM on the 10Newsgroup dataset is shown in Table 20. The results show that FS-DFTF method has a higher number of instances correctly classified (3 504 instances over 3 930) than the six existing techniques and it improves the classification performance.

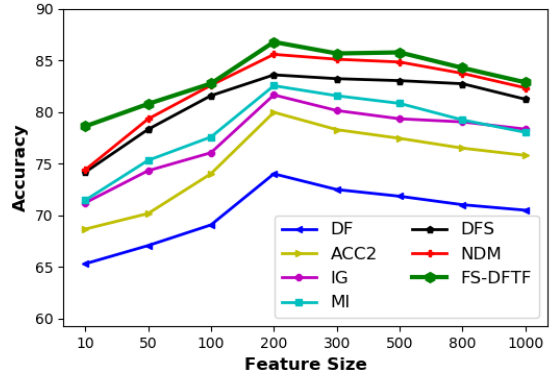
Table 21 shows the Precision, Recall and F-Score of SVM classifier using FS-DFTF, MI, DFS, DF, ACC2, IG and NDM on the 10Newsgroup dataset when top 200 features are selected in feature space. The results show that FS-DFTF method has a higher number of instances correctly classified (3 410 instances over 3 930) than the six existing techniques and it improves the classification performance.

6 VALIDITY THREATS

In this section, we discuss the validity threats for our proposed filter based feature selection scheme. We have identified two validity threats:



a) NB classifier



b) SVM classifier

Figure 5. Accuracy comparison for 10Newsgroup dataset using a) NB classifier b) SVM classifier

Algorithm	**Precision**	**Recall**	**F_1 Score**	**Accuracy in % (Correctly Classified Docs)**	**Error Rate in % (Incorrectly Classified Docs)**
DF	0.73	0.73	0.73	72.70 % (2 857)	27.30 % (1 073)
ACC2	0.73	0.73	0.73	73.03 % (2 870)	26.97 % (1 060)
IG	0.83	0.83	0.83	82.54 % (3 244)	17.46 % (686)
MI	0.83	0.83	0.83	83.23 % (3 271)	16.77 % (659)
DFS	0.85	0.85	0.85	85.44 % (3 358)	14.56 % (572)
NDM	0.86	0.86	0.86	85.90 % (3 376)	14.10 % (554)
FS-DETF	0.89	0.89	0.89	89.16 % (3 504)	10.84 % (426)

Table 20. Performance of FS-DFTF on 10Newsgroup dataset using NB classifier

Algorithm	Precision	Recall	F_1 Score	Accuracy in % (Correctly Classified Docs)	Error Rate in % (Incorrectly Classified Docs)
DF	0.74	0.74	0.74	74.02 % (2 909)	25.98 % (1 021)
ACC2	0.8	0.8	0.8	79.97 % (3 143)	20.03 % (787)
IG	0.82	0.82	0.82	81.65 % (3 209)	18.35 % (721)
MI	0.83	0.83	0.83	82.54 % (3 244)	17.46 % (686)
DFS	0.84	0.84	0.84	83.59 % (3 285)	16.41 % (645)
NDM	0.86	0.86	0.86	85.57 % (3 363)	14.43 % (567)
FS-DETF	0.87	0.87	0.87	86.77 % (3 410)	13.23 % (520)

Table 21. Performance of FS-DFTF on 10Newsgroup dataset using SVM classifier

Less or no contribution of term frequency (TF) to the text corpus. The Sarcasm headlines dataset[2] is a collection of sarcastic headlines. It contains more than 25 000 headlines. These headlines are classified into two categories (Sarcastic, Non sarcastic). In this text corpus, each document is a news headline which contains non repeated words. As a result, the term frequency (TF) does not contribute to assign the significance score to a term.

Computational cost. Even though, the performamce of the proposed FS-DFTF feature selection scheme outperformed the other feature selection scheme, the proposed method takes more computation time. Because, the proposed work uses both DF contribution and TF contribution to assign the significance score to each term which incurs some additional computational cost.

7 CONCLUSION

In this work, we propose a new filter based feature selection scheme which combines the document frequency and term frequency of the term. The performance of the proposed work FS-DFTF was investigated against well known filter based feature selection techniques using various well known benchmark datasets, two popular classification algorithms and four performance evaluation measures. The results of an in-depth experimental analysis noticeably indicate that FS-DFTF based feature selection scheme is better than other filter techniques.

Acknowledgement

The authors would like to thank the Management and Principal of Mepco Schlenk Engineering College (Autonomous), Sivakasi for providing us the state-of-the-art facilities to carry out this proposed research work in the Mepco Research Centre in collaboration with Anna University Chennai, Tamil Nadu, India.

[2] `https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection`

Conflict of Interest Statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

REFERENCES

[1] Aggarwal, C. C.—Zhai, C.: A Survey of Text Classification Algorithms. In: Aggarwal, C., Zhai, C. (Eds.): Mining Text Data. Springer US, Boston, MA, 2012, pp. 163–222, doi: 10.1007/978-1-4614-3223-4_6.

[2] Anagnostopoulos, I.—Anagnostopoulos, C.—Loumos, V.—Kayafas, E.: Classifying Web Pages Employing a Probabilistic Neural Network. IEE Proceedings – Software, Vol. 151, 2004, No. 3, pp. 139–150, doi: 10.1049/ip-sen:20040121.

[3] Banati, H.—Bajaj, M.: Firefly Based Feature Selection Approach. IJCSI International Journal of Computer Science Issues, Vol. 8, 2011, No. 4, pp. 473–480.

[4] Bracewell, D. B.—Yan, J.—Ren, F.—Kuroiwa, S.: Category Classification and Topic Discovery of Japanese and English News Articles. In: Seda, A., Boubekeur, M., Hurley, T., Mac an Airchinnigh, M., Schellekens, M., Strong, G. (Eds.): Proceedings of the Irish Conference on the Mathematical Foundations of Computer Science and Information Technology (MFCSIT 2006). Electronic Notes in Theoretical Computer Science, Vol. 225, 2009, pp. 51–65, doi: 10.1016/j.entcs.2008.12.066.

[5] Chen, J.—Huang, H.—Tian, S.—Qu, Y.: Feature Selection for Text Classification with Naïve Bayes. Expert Systems with Applications, Vol. 36, 2009, No. 3, Part 1, pp. 5432–5435, doi: 10.1016/j.eswa.2008.06.054.

[6] Chen, R. C.—Hsieh, C. H.: Web Page Classification Based on a Support Vector Machine Using a Weighted Vote Schema. Expert Systems with Applications, Vol. 31, 2006, No. 2, pp. 427–435, doi: 10.1016/j.eswa.2005.09.079.

[7] Cheng, N.—Chandramouli, R.—Subbalakshmi, K. P.: Author Gender Identification from Text. Digital Investigation, Vol. 8, 2011, No. 1, pp. 78–88, doi: 10.1016/j.diin.2011.04.002.

[8] Dada, E. G.—Bassi, J. S.—Chiroma, H.—Abdulhamid, S. M.—Adetunmbi, A. O.—Ajibuwa, O. E.: Machine Learning for Email Spam Filtering: Review, Approaches and Open Research Problems. Heliyon, Vol. 5, 2019, No. 6, Art. No. e01802, doi: 10.1016/j.heliyon.2019.e01802.

[9] Dasgupta, A.—Drineas, P.—Harb, B.—Josifovski, V.—Mahoney, M. W.: Feature Selection Methods for Text Classification. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07), ACM, 2007, pp. 230–239, doi: 10.1145/1281192.1281220.

[10] Drucker, H.—Wu, D.—Vapnik, V. N.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks, Vol. 10, 1999, No. 5, pp. 1048–1054, doi: 10.1109/72.788645.

[11] Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research, Vol. 3, 2003, No. 1, pp. 1289–1305.

[12] Günal, S.: Hybrid Feature Selection for Text Classification. Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 20, 2012, No. Sup. 2, pp. 1296–1311.

[13] Gunal, S.—Edizkan, R.: Subspace Based Feature Selection for Pattern Recognition. Information Sciences, Vol. 178, 2008, No. 19, pp. 3716–3726, doi: 10.1016/j.ins.2008.06.001.

[14] Günal, S.—Ergin, S.—Gülmezoğlu, M. B.—Gerek, Ö. N.: On Feature Extraction for Spam E-Mail Detection. In: Gunsel, B., Jain, A. K., Tekalp, A. M., Sankur, B. (Eds.): Multimedia Content Representation, Classification and Security (MRCS 2006). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4105, 2006, pp. 635–642, doi: 10.1007/11848035_84.

[15] Gunal, S.—Gerek, O. N.—Ece, D. G.—Edizkan, R.: The Search for Optimal Feature Set in Power Quality Event Classification. Expert Systems with Applications, Vol. 36, 2009, No. 7, pp. 10266–10273, doi: 10.1016/j.eswa.2009.01.051.

[16] Guru, D. S.—Suhil, M.—Pavithra, S. K.—Priya, G. R.: Ensemble of Feature Selection Methods for Text Classification: An Analytical Study. In: Abraham, A., Muhuri, P. K., Muda, A. K., Gandhi, N. (Eds.): Intelligent Systems Design and Applications (ISDA 2017). Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 736, 2018, pp. 337–349, doi: 10.1007/978-3-319-76348-4_33.

[17] Guyon, I.—Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, Vol. 3, 2003, pp. 1157–1182.

[18] Hsu, C. W.—Lin, C. J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks, Vol. 13, 2002, No. 2, pp. 415–425, doi: 10.1109/72.991427.

[19] Huang, C. L.—Wang, C. J.: A GA-Based Feature Selection and Parameters Optimization for Support Vector Machines. Expert Systems with Applications, Vol. 31, 2006, No. 2, pp. 231–240, doi: 10.1016/j.eswa.2005.09.024.

[20] Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveirol, C. (Eds.): Machine Learning (ECML'98). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1398, 1998, pp. 137–142, doi: 10.1007/bfb0026683.

[21] Kohavi, R.—John, G. H.: Wrappers for Feature Subset Selection. Artificial Intelligence, Vol. 97, 1997, No. 1-2, pp. 273–324, doi: 10.1016/s0004-3702(97)00043-x.

[22] Kumar, M. A.—Gopal, M.: A Comparison Study on Multiple Binary-Class SVM Methods for Unilabel Text Categorization. Pattern Recognition Letters, Vol. 31, 2010, No. 11, pp. 1437–1444, doi: 10.1016/j.patrec.2010.02.015.

[23] Lan, M.—Tan, C. L.—Su, J.—Lu, Y.: Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, 2009, No. 4, pp. 721–735, doi: 10.1109/tpami.2008.110.

[24] Lee, C.—Lee, G. G.: Information Gain and Divergence-Based Feature Selection for Machine Learning-Based Text Categorization. Information Processing and Management, Vol. 42, 2006, No. 1, pp. 155–165, doi: 10.1016/j.ipm.2004.08.006.

[25] Lewis, D. D.—Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization. Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, US, 1994, pp. 81–93.

[26] Li, B.—Yan, Q.—Xu, Z.—Wang, G.: Weighted Document Frequency for Feature Selection in Text Classification. 2015 International Conference on Asian Language Processing (IALP), Suzhou, China, 2016, pp. 132–135, doi: 10.1109/IALP.2015.7451549.

[27] Liu, H.—Sun, J.—Liu, L.—Zhang, H.: Feature Selection with Dynamic Mutual Information. Pattern Recognition, Vol. 42, 2009, No. 7, pp. 1330–1339, doi: 10.1016/j.patcog.2008.10.028.

[28] Manning, C. D.—Raghavan, P.—Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, USA, 2008.

[29] Mesleh, A. M.—Kanaan, G.: Support Vector Machine Text Classification System: Using Ant Colony Optimization Based Feature Subset Selection. 2008 International Conference on Computer Engineering and Systems, Cairo, Egypt, 2008, pp. 143–148, doi: 10.1109/icces.2008.4772984.

[30] Rehman, A.—Javed, K.—Babri, H. A.: Feature Selection Based on a Normalized Difference Measure for Text Classification. Information Processing and Management, Vol. 53, 2017, No. 2, pp. 473–489, doi: 10.1016/j.ipm.2016.12.004.

[31] Kim, S.-B.—Han, K.-S.—Rim, H.-C.—Myaeng, S.-H.: Some Effective Techniques for Naive Bayes Text Classification. IEEE Transactions on Knowledge and Data Engineering, Vol. 18, 2006, No. 11, pp. 1457–1466, doi: 10.1109/tkde.2006.180.

[32] Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol. 34, 2002, No. 1, pp. 1–47, doi: 10.1145/505282.505283.

[33] Shang, W.—Huang, H.—Zhu, H.—Lin, Y.—Qu, Y.—Wang, Z.: A Novel Feature Selection Algorithm for Text Categorization. Expert Systems with Applications, Vol. 33, 2007, No. 1, pp. 1–5, doi: 10.1016/j.eswa.2006.04.001.

[34] Stamatatos, E.: Author Identification: Using Text Sampling to Handle the Class Imbalance Problem. Information Processing and Management, Vol. 44, 2008, No. 2, pp. 790–799, doi: 10.1016/j.ipm.2007.05.012.

[35] Su, J.—Sayyad-Shirabad, J.—Matwin, S.: Large Scale Text Classification Using Semi-Supervised Multinomial Naive Bayes. Proceedings of the 28th International Conference on Machine Learning (ICML 2011), Bellevue, Washington, USA, 2011, pp. 97–104.

[36] Thirumoorthy, K.—Muneeswaran, K.: Optimal Feature Subset Selection Using Hybrid Binary Jaya Optimization Algorithm for Text Classification. Sādhanā, Vol. 45, 2020, No. 1, Art. No. 201, pp. 1–13, doi: 10.1007/s12046-020-01443-w.

[37] Uysal, A. K.—Gunal, S.: A Novel Probabilistic Feature Selection Method for Text Classification. Knowledge-Based Systems, Vol. 36, 2012, pp. 226–235, doi: 10.1016/j.knosys.2012.06.005.

[38] Xu, Y.—Wang, B.—Li, J.—Jing, H.: An Extended Document Frequency Metric for Feature Selection in Text Categorization. In: Li, H., Liu, T., Ma, W. Y., Sakai, T., Wong, K. F., Zhou, G. (Eds.): Information Retrieval Technology (AIRS

2008). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4993, 2008, pp. 71–82, doi: 10.1007/978-3-540-68636-1_8.

[39] Yang, J.—Honavar, V.: Feature Subset Selection Using a Genetic Algorithm. IEEE Intelligent Systems and Their Applications, Vol. 13, 1998, No. 2, pp. 44–49, doi: 10.1109/5254.671091.

[40] Wu, Y.—Zhang, A.: Feature Selection for Classifying High-Dimensional Numerical Data. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), 2004, Vol. 2, pp. II–II, doi: 10.1109/cvpr.2004.1315171.

[41] Yu, B.—Zhu, D.-H.: Combining Neural Networks and Semantic Feature Space for Email Classification. Knowledge-Based Systems, Vol. 22, 2009, No. 5, pp. 376–381, doi: 10.1016/j.knosys.2009.02.009.

Thirumoorthy Karpagalingam received his M.E. degree from the Arulmigu Kalasalingam College of Engineering, Krishnankoil, Tamilnadu, India and his B.E. degree from the Kamaraj College of Engineering and Technology, India. He worked as Assistant Professor in Computer Science and Engineering Department of the Mepco Schlenk Engineering College, Sivakasi for 12 years and currently he is pursuing Ph.D. in Mepco Schlenk Engineering College, Sivakasi, India under Anna University, Chennai, India. His research areas include text mining. His publications have appeared in various leading journals and conferences.

Muneeswaran Karuppaiah is presently Senior Professor in the Department of Computer Science and Engineering at Mepco Schlenk Engineering College, Sivakasi, India. He completed his doctorate in M.S. University, Tirunelveli, India, his post graduate degree from PSG College of Technology, Coimbatore, India and Bachelor degree from Thiagarajar College of Engineering, Madurai, India. He has nearly 33 years of teaching experience. He has published nearly 52 papers in reputed international journals with 531 citations and 95 papers in international and national conferences. He has published a book "Compiler Design" in Oxford University Press. He is also a life member of organizations such as the Computer Society of India (CSI), Indian Society for Technical Education (ISTE) and Institution of Electronics and Telecommunication Engineers (IETE).