Knowledge-Based Systems 24 (2011) 1024–1032

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm

Harun Uğuz *

Department of Computer Engineering, Selçuk University, Konya, Turkey

ARTICLE INFO

Article history:
Received 8 September 2010
Received in revised form 23 April 2011
Accepted 23 April 2011
Available online 29 April 2011

Keywords:
Text categorization
Feature selection
Genetic algorithm
Principal component analysis
Information gain

ABSTRACT

Text categorization is widely used when organizing documents in a digital form. Due to the increasing number of documents in digital form, automated text categorization has become more promising in the last ten years. A major problem of text categorization is its large number of features. Most of those are irrelevant noise that can mislead the classifier. Therefore, feature selection is often used in text categorization to reduce the dimensionality of the feature space and to improve performance. In this study, two-stage feature selection and feature extraction is used to improve the performance of text categorization. In the first stage, each term within the document is ranked depending on their importance for classification using the information gain (IG) method. In the second stage, genetic algorithm (GA) and principal component analysis (PCA) feature selection and feature extraction methods are applied separately to the terms which are ranked in decreasing order of importance, and a dimension reduction is carried out. Thereby, during text categorization, terms of less importance are ignored, and feature selection and extraction methods are applied to the terms of highest importance; thus, the computational time and complexity of categorization is reduced. To evaluate the effectiveness of dimension reduction methods on our purposed model, experiments are conducted using the *k*-nearest neighbour (KNN) and C4.5 decision tree algorithm on Reuters-21,578 and Classic3 datasets collection for text categorization. The experimental results show that the proposed model is able to achieve high categorization effectiveness as measured by precision, recall and *F*-measure.

1. Introduction

The number of text documents in digital format is progressively increasing and text categorization becomes the key technology to organize text data. Text categorization is defined as assigning new documents to a set of pre-defined categories based on the classification patterns [2,29]. Although many information retrieval applications [3] such as filtering and searching for relevant information can benefit from text categorization research, a major problem of text categorization is the high dimensionality of the feature space due to a large number of terms. This problem may cause the computational complexity of machine learning methods used for text categorization to be increased and may bring about inefficiency and results of low accuracy due to redundant or irrelevant terms in the feature space [20,41,46]. For a solution to this problem, two techniques are used: feature extraction and feature selection.

Feature extraction is a process that extracts a set of new features from the original features into a distinct feature space [38]. Some feature extraction methods have been successfully used in text categorization, such as principal component analysis (PCA) [16,30], latent semantic indexing [33], clustering methods [31], etc. Among the many methods that are used for feature extraction, PCA has attracted a lot of attention. PCA [15] is a statistical technique for reduction of dimensionality that aims at minimizing loss in variance in the original data. It can be viewed as a domain independent technique for feature extraction, which is applicable to a wide variety of data [16].

Feature selection is a process that selects a subset from the original feature set according to some criteria of feature importance [22]. A number of feature selection methods are successfully used in a wide range of text categorizations. Yang and Pedersen [40] compared five feature selection methods for text categorization including information gain (IG), χ^2 statistic document frequency, term strength, and mutual information. They reported that IG is the most effective method among the compared feature selection methods. In addition to these feature selection methods, biologically inspired algorithms such as genetic algorithm (GA) [7,32,45] and ant colony optimization algorithm [1] have been successfully used in the literature for text categorization.

Genetic algorithm is an optimization method mimicking the evolution mechanism of natural selection. GA performs a search in complex and large landscapes and provides near-optimal solutions for optimization problems [32].

* Tel.: +90 332 223 19 26; fax: +90 332 241 06 35.
E-mail addresses: harun_uguz@selcuk.edu.tr, harun_uguz@hotmail.com

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.04.014

Text categorization is the task of classifying a document into predefined categories based on the contents of the document [4]. In recent years, more and more methods have been applied to the text categorization task based on statistical theories and machine learning, such as KNN [21,34,39], Naive Bayes [4,23], Rocchio [13], decision tree [6,9], support vector machine (SVM) [14,21,44], neural network [19,42], and so on. In this study, the C4.5 decision tree and KNN methods, which are used for text categorization, are used as classifiers.

In the current study, a two-stage feature selection and feature extraction are used to reduce the high dimensionality of a feature space composed of a large number of terms, remove redundant and irrelevant features from the feature space and thereby decrease the computational complexity of the machine learning algorithms used in the text categorization and increase performances thereof. In the first stage, each term in the text is ranked depending on their importance for the classification in decreasing order using the IG method. Therefore, terms of high importance are assigned to the first ranks and terms of less importance are assigned to the following ranks. In the second stage, the PCA method selected for feature selection and the GA method selected for feature extraction are applied separately to the terms of highest importance, in accordance with IG methods, and a dimension reduction is carried out. In this way, during text categorization, terms of less importance are ignored, feature selection and feature extraction methods are applied to the terms of the highest importance, and the computational time and complexity of the category are reduced. To evaluate the effectiveness of dimension reduction methods, experiments are conducted on Reuters-21,578 and Classic3 datasets collection for text categorization. The experimental results show that the proposed model is able to achieve high categorization effectiveness as measured by precision, recall and *F*-measure.

The rest of this paper is organized as follows. Section 2 presents a brief overview of the research methodologies and the experimental setting used. The effectiveness of the purposed method and experimental results for the categorization of a text document are demonstrated in Section 3, and finally, the paper is concluded in Section 4.

2. Research methodologies

The parts of proposed text categorization structure are shown in Fig. 1. These parts are explained in the following subsections:

2.1. Datasets

In this section, Reuters-21,578 and the Classic3 datasets used in the experiments are described and analysed.

2.1.1. Reuters-21,578 dataset

There are some public datasets that can be used as test collections for text categorization. The most widely used is the Reuters collection, which contains documents collected from Reuters news agency. The Reuters-21,578 collection [18] is a set of economic news published by Reuters in 1987. This collection includes 21,578 documents that are organized in 135 categories. In this experiment, the six categories including a minimum of 500 terms are selected. There are 8158 documents belonging to the chosen categories. The distributions of the number of documents in the six categories are shown in Table 1. According to Table 1, the distribution of documents into the categories is unbalanced. Maximum and minimum categories occupy 45.88% and 6.13% of the dataset, respectively.

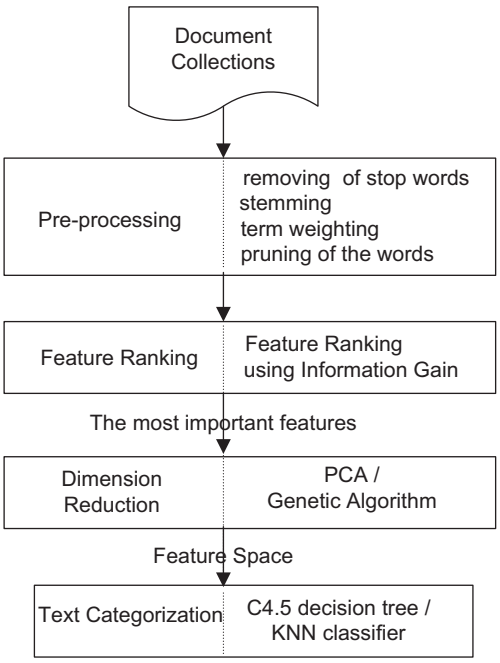

Fig. 1. Purposed text categorization structure.

Table 1
Distributions of the six categories for Reuters-21,578 Dataset.

Category name	Number of document
Earn	3743
Acquisition	2179
Money-fx	633
Crude	561
Grain	542
Trade	500

2.1.2. Classic3 dataset

We implemented the second experiment on the Classic3 dataset, a document collection from the SMART project at Cornell University (ftp://ftp.cs.cornell.edu/pub/smart). The Classic3 dataset is frequently used to evaluate performance of text categorization algorithms because it contains a known number of fairly well-separated groups. It contains three categories, i.e., 1398 CRANFIELD documents from aeronautical system papers, 1033 MEDLINE documents from medical papers, and 1460 CISI documents from information retrieval papers. The distribution of documents into the categories is balanced since all the categories are represented equally well in the dataset.

2.2. Pre-Processing

2.2.1. Removing of stop-words

Words such as conjunctions and pronouns that are not related to the concept of the text are called stop-words. This process involves removing certain common words such as 'a', 'an', 'the', etc., that occur commonly in all documents. It is important to removing these high-frequency words because they may misclassify the documents. In the study, stop words are removed in accordance with the existing stop word list (http://www.unine.ch/Info/clef/), which consists of 571 words.

2.2.2. Stemming

The stemming process leaves out the root forms of the words. Thereby, terms sharing the same root that seem like different

words due to their affixes can be determined. For example, "computer," "computing," "computation," and "computes" all have the same comput root. Porter's stemming algorithm [26] is used for stemming.

2.2.3. Term weighting

After the words are transformed into terms, the presentation form of the document, which means the expression thereof, terms have to be determined. This process is called term weighting. Thereby, each document could be written in a vector form depending on the terms they contained. This document vector will generally be in the following format:

$$d = \{w_1, \ldots, w_i, \ldots, w_{|T|}\}, \tag{1}$$

where w_i is the weight of the term with number i in the d document, T is the term set, and $|T|$ is the cardinality of T.

To obtain the term vector of T, the *tfidf* is generally used as its weight scheme. Accordingly, let the term frequency tf_i be the number of occurrences of t_i in the document, and let the document frequency df_i be the number of the document in which the t_i term is seen at least once. The inverse document frequency idf_i is calculated as shown in Eq. (2) using df_i [28]

$$idf_i = \log\left(\frac{|D|}{df_i}\right), \tag{2}$$

where $|D|$ the number of all of the documents in the training set and w_i is calculated in accordance with Eq. (3).

$$w_i = tf_i.idf_i. \tag{3}$$

2.2.4. Pruning of the words

The pruning process basically filters less frequent features in a document collection. The term vector is very high-dimensional and sparse. Also, it is seen that a number of elements in the term vector is "0". Therefore, we prune the words that appear less than two times in the documents. This process decreases the term vector dimension further.

2.3. Feature ranking with Information gain

Information gain is one of the popular approaches employed as a term importance criterion in the text document data [13,40]. The idea is based on information theory [24]. The information gain of term t is defined in Eq. (4)

$$IG(t) = -\sum_{i=1}^{|C|} P(c_i)\log P(c_i) + P(t)\sum_{i=1}^{|C|} P(c_i|t)\log P(c_i|t) + P(\bar{t}) \times \sum_{i=1}^{|C|} P(c_i|\bar{t})\log P(c_i|\bar{t}), \tag{4}$$

where c_i represents the ith category, $P(c_i)$ is the probability of the ith category, $P(t)$ and $P(\bar{t})$are the probabilities that the term t appears or not in the documents, respectively, $P(c_i|t)$ is the conditional probability of the ith category given that term t appeared, and $P(c_i|\bar{t})$ is the conditional probability of the ith category given that term t does not appeared.

In this study, before dimension reduction, each term within the text is ranked depending on their importance for the classification in decreasing order using the IG method. Thereby, in the process of text categorization, terms of less importance are ignored, and dimension reduction methods are applied to the terms of highest importance.

2.4. Dimension reduction methods

At the end of the pre-processing step, terms of high importance in documents are acquired through the IG method. In this manner, even though the number of terms in the document is reduced, the main problem for the text categorization is the high dimensionality of the feature space. Therefore, to reduce the feature space dimension and the computational complexity of the machine learning algorithms used in the text categorization and increase the performances thereof, the GA and PCA dimension reduction methods are applied. The aim of these methods is to minimize information loss while maximising reduction in dimensionality. The PCA method is used alternatively to the GA for the reduction of the feature space dimension.

2.4.1. Principal component analysis

PCA is a statistical technique that is used for extracting information from a multi-variety dataset. This process is performed by identifying the principal components of original variables with linear combinations. Whereas the original dataset with the maximum variability is represented with the first principal component, the dataset from the remaining dataset with the maximum variability is represented with the second principal component. The process goes on consecutively as such, with the dataset from the remaining dataset with the maximum variability being represented with the next principal component. And feature extraction methods are applied to the terms of the highest importance, and m represents the number of all principal components, and p represents the number of the significant principal components among all principal components, p may be defined as the number of those principal components of the m-dimensional dataset with the highest variance values. It is clear therein that $p \leqslant m$.Therefore, the PCA may be defined as the data-reducing technique. In other words, PCA is the technique used to produce the lower-dimensional version of the original dataset [43]. Details of the PCA can be reached from [15].

The most significant stage in the application of the PCA is the determination of the number of principal component. The p number of principal components to be chosen among all of the principal components should be the principal components to represent the data at their very best. There are certain criteria in determining the optimal number of principal components. The broken-stick model, Velicier's partial correlation procedure, cross-validation, Barlett's test for equality of eigen-values, Kaiser's criterion, Cattell's screen-test, and cumulative percentage of variance are such a few criteria [8,35]. In this study, cumulative percentage of variance criteria has been applied to determine the number of principal components, for its simplicity, and for its eligible performance [35]. According to this criterion, principal components are chosen based on their cumulative percentage of variance higher than a prescribed threshold value. Although a sensible threshold is very often in the range between 70% and 90%, it can sometimes be higher or lower depending on the practical details of a particular dataset. However, it should be noticed that some authors point out that there is no ideal solution to the problem of dimensionality in a PCA [15]. Therefore, the choice of threshold is often selected heuristically [37]. In this study, the threshold value in both datasets is specified as 75% in all applications performed via a PCA.

2.4.2. Genetic algorithm for feature selection

The genetic algorithm is an optimization method mimicking the evolution [12]. This algorithm, which is an effective optimization method in wide search spaces, is preferred because it is the appropriate method for the solution of the problem.

To apply the genetic algorithm, the problem should first be adapted to the genetic algorithm. In other words, the basic structures of the genetic algorithm, such as genes, chromosomes, and

Table 2
Genetic algorithm parameters.

Modeling description	Setting
Population size	30
Selection technique	Roulette wheel
Crossover type	Two point crossover
Crossover rate	0.9
Mutation rate	0.001
Iteration number	500

population, should be determined. In this phase, coding, selection, crossover, mutation, and fitness functions should be chosen. Details for mastering the art of the genetic algorithm are published elsewhere [10,11].

Although, terms of high importance in documents are acquired through IG method, the main problem in our application still is the high dimensionality of the feature space. Since given a feature set U via IG method is high dimensionality, it is impractical to evaluate all the possible subsets of U. Due to this deficiency, GA-based feature selection method is adopted in this study. Accordingly, GA is used to provide near-optimal solutions for feature selection. The objective of the GA-based feature selection is to find the optimal subset of a given feature set U that maximizes classification performance in this study.

Genetic algorithm parameters used in our work are given in the Table 2. GA parameters in Table 2 are empirically determined in our implementation.

The details of our implementation are given in the following subsections.

2.4.2.1. Individual's encoding. In the GA-based approach to feature selection, a candidate feature set can be represented by a binary string called a chromosome. Chromosomes comprising population are encoded in the form of binary vector in a manner to compose of genes as the number of feature in each feature space. The ith bit in the chromosome represents the presence of the ith feature.

Initialization of the population is commonly done by seeding the population with random values. If the value of the gene, which is coded in binary system, is ''1'', it means that the corresponding feature is selected, in the contrary, if the value of gene is ''0'', it means that the corresponding feature is not selected. In the proposed GA, each chromosome is initialized randomly, with each chromosome in the population coded to a binary. The length of chromosome is equal to the total number of features.

2.4.2.2. Fitness function. Fitness function is used to decide which individuals are fit to optimum solution. Every individual has its own fitness value. A higher value of fitness means that the individual is more appropriate as a problem solution; on the other hand, a lower value of fitness means that the individual is less appropriate as a problem solution.

After the initialization of population, the encoded chromosomes are searched to optimize a fitness function. In this study, the fitness value of each chromosome is evaluated according to its average value of F-measure (Eq. (10)) on a set of testing data using a C4.5 decision tree or KNN classifier, and then the return values of the fitness function are sorted from small to large.

2.4.2.3. Selection. The object of the selection process is to choose the individuals of the next generation according to the selected fitness function and selection method among the existing population. In the selection process, the transfer possibility of the fittest individual's chromosome to the next generation is higher than others. The decision of the individual's characteristic which will be transferred to the next generation is based on the values evaluated from the fitness function and shows the quality of the individual. The Roulette Wheel Selection method which is the most general and most easily applied [11] one is chosen in this work.

2.4.2.4. Crossover. In the pre-crossover phase, individuals are determined by using a mating process. Forming the new generation is called 'crossover'. The most widely used method is forming two new individuals from the two chromosomes. In our study, two-point crossover is used in the crossover procedure.

2.4.2.5. Mutation. To increase the variety of the chromosomes which are applied on crossover, process mutation process can be applied. Mutation introduces local variations to the individuals for searching different solution spaces and keeps the diversity of the population. In our study, the number of chromosomes that will be mutated is determined according to the mutation rate and their values are changed from '1' to '0' or '0' to '1' respectively.

2.5. Text categorization methods

In this study, two separate classifier methods are used in text categorization; the C4.5 decision tree and KNN methods are used due to their simplicity and accuracy in text categorization. These methods are separately applied to the classification of datasets in which the dimension acquired at the end of the GA and PCA application is reduced. The reason for using a classifier is to compare the performances of the both methods in the text categorization. Brief descriptions of these methods are given, as follows.

2.5.1. KNN classifier

The KNN [5] algorithm is a well-known instance-based approach that has been widely applied to text categorization due to its simplicity and accuracy [17,39].

To categorize an unknown document, the KNN classifier ranks the document's neighbours among the training documents and uses the class labels of the k most similar neighbours. Similarity between two documents may be measured by the Euclidean distance, cosine measure, etc. The similarity score of each nearest neighbour document to the test document is used as the weight of the classes of the neighbour document. If a specific category is shared by more than one of the k-nearest neighbours, then the sum of the similarity scores of those neighbours is obtained from the weight of that particular shared category [25]. A detailed procedure of KNN can be referred to in Cover and Hart [5].

At the phase when classification is done by means of the KNN, the most important parameter affecting classification is k-nearest neighbour number. Usually, the optimal value of k is empirically determined. In our study, k value is determined so that it would give the least classification error ($k = 3$ is determined). In addition, in the phase of finding the k-nearest neighbourhood, Euclidean distance is used as the distance metric.

2.5.2. C4.5 decision tree classifier

The decision tree is a well-known machine learning approach to automate the induction of classification trees based on training data [27]. In a typical decision tree training algorithm, there are usually two phases. The first phase is tree growing where a tree is built by greedily splitting each tree node. Because the tree can overfit the training data, in the second phase, the overfitted branches of the tree are removed [6]. C4.5 is a univariate decision tree algorithm. At each node, only one attribute of the instances are used for decision making. Details of C4.5 can be reached from Fuhr and Buckley [9].

In our application, by using C4.5 decision tree algorithms, in the pruning phase, the post-pruning method is used to decide when to stop expanding a decision tree. The confidence factor is used for

pruning the tree. In our study, the confidence factor is assigned as 0.25. The pruned trees consist of 4 leaves and 8 nodes.

2.6. Evaluation of the performance

The *F*-measure, precision and recall are usually employed to evaluate the accuracy of text categorization results. These measures are used to evaluate the accuracy of the result of the KNN and C4.5 classifiers for text categorization. The *F*-measure is a harmonic combination of the precision and recall values used in information retrieval [36]. Precision is the proportion of the correctly proposed documents to the proposed documents, while recall is the proportion of the correctly proposed documents to the test data that have to be proposed [20]. In this study, the *F*-measure, precision and recall are not separated; they are calculated for each category, and the average values of the measures are used.

Precision P_i and recall R_i of category i are defined in Eqs. (5) and (6), respectively.

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{5}$$

$$R_i = \frac{TP_i}{TP_i + FN_i}, \tag{6}$$

where TP_i,FP_i and FN_irepresent the number of true positives, false positives, and false negatives, respectively. Then, the average precision (P) and recall (R) measures are calculated as Eqs. (7) and (8), respectively.

$$P = \frac{\sum_{i=1}^{N} d_i.P_i}{\sum_{i=1}^{N} d_i}, \tag{7}$$

$$R = \frac{\sum_{i=1}^{N} d_i.R_i}{\sum_{i=1}^{N} d_i}, \tag{8}$$

where d_i is the number of documents category i contains. N is the number of categories.

The *F*-measure F_i of category i is defined in Eq. (9).

$$F_i = \frac{2.P_i.R_i}{P_i + R_i}. \tag{9}$$

Then, the average *F*-measure (F) is calculated as Eq. (10).

$$F = \frac{\sum_{i=1}^{N} d_i.F_i}{\sum_{i=1}^{N} d_i}, \tag{10}$$

where d_i is the number of documents category i contains. N is the number of categories.

3. Results

Experiments are conducted for text categorization on two different datasets to examine the performance of the proposed method, dimension reduction and classifier techniques. Pre-processing, dimension reduction and classification processes are implemented by the Matlab software package. A *10 fold* cross validation procedure is preferred for the classification stages. All experiments are run on a machine with 2.8 GHz CPU, 4 GB of RAM, 500 GB HDD space, and the Windows 7 operation system.

3.1. Results on Reuters-21,578 dataset

3.1.1. Pre-processing

The pre-processing process is performed in four stages. The first step consisted of removing the stop words because they are useless for the classification. In the study, stop words are removed in accordance with the existing stop word list of 571 (http://www.u-nine.ch/Info/clef/) words. After removing the stopwords, the dataset contains 10764 unique words. In the second step, the Porter algorithm [26] is used for stemming. In the third step, the document vectors are built with the *tfidf* weighting scheme. In the fourth step, to reduce the size of the term set, we discard terms that appeared in less than two documents. The total number of terms finally extracted is 7542. Thereby, a document-term matrix is acquired with a dimension of 8158×7542 at the end of preprocessing.

3.1.2. Feature ranking, dimension reduction and text categorization with C4.5 and KNN classifiers results on Reuters-21,578 dataset

At this stage, in order to test the efficiency of proposed IG-GA and IG-PCA based feature reduction methods and to evaluate the success of these methods individually, features are selected from feature space at different ratios (1–10% of features with IG). We test all applications by using the *10-fold* cross validation. The results in terms of precision, recall and *F*-measure are the averaged values calculated across all *10-fold* cross validation experiments.

To examine overall performance of without dimension reduction, initially, KNN and C4.5 decision tree classifiers are applied on the whole of the document-term feature space. The experimental results with the KNN and C4.5 decision tree classifier are summarized in Table 3. As seen in Table 3, in applications made without using any dimension reduction method, the highest accuracy is obtained when the C4.5 classifier is used.

After that, feature ranking is applied via the IG method to reduce the high dimension of the feature space. In this phase, the effects of the individual feature ranking operation by the IG method on classifier performance are examined. Accordingly, features are ranked in decreasing order using the IG method. Of the features ranked by IG (the highest important features), 1–10% are separately classified from the C4.5 and KNN classifiers. Table 4 shows the classification performances at the end of feature ranking operation performed by IG. According to Table 4, the highest accuracy with the KNN classifier is obtained when 4% of the ranked features are used. In addition, the highest accuracy with C4.5 classifier is obtained when 6% of the ranked features are used. When the classifier performances are compared, the KNN algorithm shows a higher performance than the C4.5 decision tree algorithm. If Table 4 is compared with Table 3, we can see that the highest accuracies are obtained at the end of feature ranking operations made by IG. Furthermore, it is seen that using features (1–10%) ranked with IG instead of all features positively contributed to the classifier performances in an affirmative manner. As for the feature ranking, average improvement in *F*-measures for C4.5 classifier is 9% and average improvement in *F*-measures for KNN classifier is 13%.

Finally, the effects of IG–GA and IG–PCA based methods on classifier performances are examined. Accordingly, dimension reduction process is applied separately by GA and PCA to the 1–10% of features ranked according to importance for classification by IG.

Table 5 shows the classification performances at the end of the feature ranking and feature selection operation performed using the IG–GA method. According to Table 5, the highest accuracy is obtained when 4% and 6–9% of the ranked features of the KNN and C4.5 classifiers are used, respectively. When Tables 4 and 5 are analysed, it is evident that although fewer features are selected via the IG–GA method, precision, recall and the *F*-measure values

Table 3
The performance (average value of precision, recall and *F*-measure) of KNN and C4.5 decision tree classifier on Reuters-21,578 dataset.

Classifier	No. of features	Precision	Recall	*F*-measure
KNN	7542	73.36	95.59	83.02
C4.5	7542	84.64	89.23	86.88

Table 4
The performance (average value of precision, recall and *F*-measure) of KNN and C4.5 decision tree classifier with IG on Reuters-21,578 dataset.

Percentage of feature %	KNN				C4.5 decision tree			
	Number of Features	Precision (%)	Recall (%)	*F*-measure (%)	Number of features	Precision (%)	Recall (%)	*F*-measure (%)
1	75	95.14	94.71	94.93	75	94.50	94.63	94.57
2	151	94.26	97.38	95.80	151	94.82	94.84	94.83
3	226	94.03	97.62	95.79	226	94.86	94.63	94.74
4	302	94.87	97.86	96.34	302	95.48	95.38	95.43
5	377	94.04	97.73	95.85	377	94.83	96.02	95.42
6	453	93.33	97.54	95.39	453	95.61	95.40	95.51
7	528	91.74	97.86	94.70	528	95.21	94.47	94.84
8	603	91.07	97.78	94.31	603	95.32	95.27	95.30
9	679	90.63	97.92	94.13	679	95.24	94.52	94.88
10	754	90.14	97.65	93.74	754	95.18	95.43	95.30

Table 5
The performance (average value of precision, recall and *F*-measure) of KNN and C4.5 decision tree classifier with IG–GA method on Reuters-21,578 dataset.

Percentage of feature %	KNN				C4.5 decision tree			
	Number of features	Precision (%)	Recall (%)	*F*-measure (%)	Number of features	Precision (%)	Recall (%)	*F*-measure (%)
1	42	95.37	94.68	95.03	45	96.20	93.40	94.78
2	83	96.64	95.99	96.31	78	95.98	94.47	95.22
3	121	97.50	96.93	97.21	116	95.39	94.60	95.00
4	169	98.17	97.52	97.84	175	95.95	95.65	95.80
5	197	97.73	97.60	97.66	201	96.41	95.40	95.90
6	241	97.42	97.73	97.57	244	96.51	95.40	95.96
7	286	97.16	97.84	97.50	281	96.11	95.65	95.88
8	317	97.04	98.05	97.54	328	96.40	95.11	95.75
9	352	97.04	98.10	97.57	355	95.84	96.08	95.96
10	384	96.93	97.78	97.35	380	95.72	95.51	95.61

are higher only when compared to feature selection carried out via the IG method. Moreover, when Tables 3–5 are examined, it can be observed that the highest accuracy with the least number of features is obtained by the proposed IG–GA method.

Table 6 shows the classification performances at the end of the feature ranking and feature extraction operation performed by the IG–PCA method. According to Table 6, the highest accuracy is obtained when 4% and 6% of the ranked features for the KNN and C4.5 classifiers are used, respectively. Similar to the IG–GA method, although fewer features are selected via the IG–PCA method, precision, recall and *F*-measure values are higher only in comparison to a feature selection carried out via the IG method. When Tables 3–6 is examined, it can be observed that the IG–GA method shows higher classifier accuracy in comparison with the IG and IG–PCA method.

As understood from these results, when there are many irrelevant or redundant features in the feature space, performing a feature ranking, feature extraction and feature selection method could remove them, thereby, improving classifier performance. The results show that higher classification accuracy is obtained with less number of features when IG-GA and IG-PCA methods are used as hybrid. Furthermore, using IG, PCA and GA methods as hybrid, improves the classification efficiency and accuracy compared with individual usage of IG method.

For the performance of classifiers with dimension reduction methods, the C4.5 decision tree algorithm seems to perform worse than the KNN algorithm. However, one of the advantages of the C4.5 decision tree algorithm is its potential for data exploration purposes. Consequently, it is seen that a higher classifier performance is acquired with fewer features through the purposed two-stage dimension reduction process.

3.2. Results on Classic3 dataset

3.2.1. Pre-processing

Similarly to the application carried out on the Reuters-21,578, stop words are removed in accordance with the existing stop word list with 571 (http://www.unine.ch/Info/clef/) words. After removing stopwords, the dataset contains 11398 unique words. The Porter algorithm [26] is used for stemming. Then, the document vectors are built with a *tfidf* weighting scheme. In order to reduce

Table 6
The performance (average value of precision, recall and *F*-measure) of KNN and C4.5 decision tree classifier with IG-PCA method on Reuters-21,578 dataset.

Percentage of feature %	KNN				C4.5 decision tree			
	Number of features	Precision (%)	Recall (%)	*F*-measure (%)	Number of features	Precision (%)	Recall (%)	*F*-measure (%)
1	36	93.97	93.75	93.86	36	95.68	93.53	94.60
2	71	96.47	95.54	96.00	71	95.17	94.76	94.97
3	103	96.91	96.37	96.64	103	94.96	94.66	94.81
4	134	97.77	97.03	97.40	134	95.66	95.40	95.53
5	162	97.24	97.11	97.18	162	95.54	95.62	95.58
6	193	96.87	97.46	97.16	193	95.70	95.67	95.68
7	222	94.70	94.70	97.16	222	95.68	95.35	95.52
8	250	96.80	97.73	97.26	250	95.76	95.32	95.54
9	278	96.29	97.76	97.02	278	95.66	95.40	95.53
10	303	96.34	97.65	96.99	303	95.48	95.35	95.42

Table 7
The performance (average value of precision, recall and *F*-measure) of KNN and C4.5 decision tree classifier on Classic3 dataset.

Classifier	Number of features	Precision	Recall	*F*-measure
KNN	6679	60.22	98.99	74.89
C4.5	6679	85.12	89.20	85.19

the size of the term set, we discard terms which appear in less than two documents and the total number of terms extracted finally is 6679. Thereby, a document-term matrix is acquired in the dimension of 3891 × 6679 at the end of pre-processing.

3.2.2. Feature ranking, dimension reduction and text categorization with C4.5 and KNN classifiers results on Classic3 dataset

Similar to the application carried out on the Reuters-21,578 dataset, initially, the KNN and C4.5 decision tree classifiers are applied on the whole of the document-term feature space. The experimental results with the KNN and C4.5 decision tree classifiers are summarized in Table 7. As shown in Table 7, in applications made without using any dimension reduction methods, the highest accuracy is obtained when the C4.5 classifier is used.

After that, feature ranking and dimension reduction techniques are applied individually and as a hybrid to reduce the high dimension of the feature space the success of the IG, hybrid PCA and GA methods in text categorization is tested separately by using the KNN and C4.5 decision tree classifiers. The effects of IG, IG-GA and IG-PCA based hybrid methods on text categorization performances are examined in Tables 8–10, respectively.

Table 8 shows the classification performances at the end of the feature ranking operation performed by the IG. As seen in Table 8, the highest accuracy with the KNN classifier is obtained when 2% of the ranked features are used. Similarly, the highest accuracy with the C4.5 classifier is obtained when 5% of the ranked features are used. When the classifier performances are compared, the C4.5

Table 8
The performance (average value of precision, recall and *F*-measure) of KNN and C4.5 decision tree classifier with IG on Classic3 dataset.

Percentage of feature %	KNN				C4.5 decision tree			
	Number of features	Precision (%)	Recall (%)	*F*-measure (%)	Number of features	Precision (%)	Recall (%)	*F*-measure (%)
1	67	86.28	90.97	88.56	67	90.45	88.50	89.46
2	134	86.61	94.82	90.53	134	87.02	92.29	89.58
3	200	85.27	95.45	90.07	200	86.88	92.86	89.77
4	267	83.91	96.21	89.64	267	87.27	92.67	89.89
5	334	83.04	95.89	89.01	334	87.29	92.86	89.99
6	401	83.07	94.25	88.31	401	86.96	92.29	89.55
7	468	80.61	97.16	88.11	468	86.23	92.99	89.48
8	534	80.52	97.41	88.16	534	86.27	92.86	89.44
9	601	80.17	98.04	88.21	601	86.27	92.86	89.44
10	668	78.23	98.29	87.12	668	86.06	93.18	89.48

Table 9
The performance (average value of precision, recall and *F*-measure) of KNN and C4.5 decision tree classifier with IG–GA method on Classic3 dataset.

Percentage of feature %	KNN				C4.5 decision tree			
	Number of features	Precision (%)	Recall (%)	*F*-measure (%)	Number of features	Precision (%)	Recall (%)	*F*-measure (%)
1	44	88.89	91.47	90.16	46	90.50	90.27	90.39
2	81	91.75	95.58	93.63	83	93.56	94.57	94.06
3	118	92.53	97.03	94.73	115	94.01	96.15	95.07
4	152	92.81	97.85	95.26	155	94.97	96.53	95.74
5	196	93.04	98.74	95.80	194	95.29	97.09	96.18
6	238	92.86	98.55	95.62	235	95.42	96.02	95.72
7	276	92.79	99.12	95.85	279	95.50	96.53	96.01
8	314	92.85	99.31	95.97	316	96.28	96.34	96.31
9	347	92.74	99.31	95.91	344	95.48	97.35	96.40
10	376	93.48	97.85	95.62	379	95.85	97.73	96.78

Table 10
The performance (average value of precision, recall and *F*-measure) of KNN and C4.5 decision tree classifier with IG–PCA method on Classic3 dataset.

Percentage of feature (%)	KNN				C4.5 decision tree			
	Number of features	Precision (%)	Recall (%)	*F*-measure (%)	Number of features	Precision (%)	Recall (%)	*F*-measure (%)
1	39	88.57	89.58	89.07	39	90.33	90.33	90.33
2	75	92.31	94.06	93.18	75	92.52	94.57	93.53
3	110	92.48	96.34	94.37	110	94.42	95.07	94.74
4	144	92.68	97.54	95.04	144	94.37	96.27	95.31
5	177	93.09	97.92	95.44	177	94.75	96.90	95.82
6	208	92.69	97.66	95.11	208	94.92	95.70	95.31
7	238	92.86	98.61	96.65	238	95.08	96.40	95.73
8	266	93.30	98.48	95.82	266	95.15	96.72	95.93
9	293	92.94	98.99	95.87	293	95.24	97.22	96.22
10	319	92.37	98.67	95.42	319	95.43	97.60	96.50

decision tree algorithm seems to perform worse than the KNN algorithm. When analyzing Tables 7 and 8, it is seen that using ranking features (1% to 10%) via IG instead of all features contributed to the classifier performances in a positive manner. As for the dimension reduction, average improvement in *F*-measures for C4.5 classifier is 5% and average improvement in *F*-measures for KNN classifier is 15%.

Table 9 shows the classification performances at the end of feature ranking and feature selection operation performed by IG–GA method. As seen in Table 9, the highest accuracy is obtained when 6% and 10% of the ranked features for the KNN and C4.5 classifiers are used, respectively. As evident from Tables 7–9, it can be observed that the highest accuracy with the least number of features is obtained by the proposed IG–GA method. In other words, using the IG and GA methods as a hybrid, improves the classification efficiency and accuracy compared with the using the IG method individually.

Table 10 shows the classification performances at the end of the feature ranking and feature extraction operation performed by the IG–PCA method. These results show that using the IG and PCA methods as a hybrid improves the classification efficiency and accuracy compared with individually using the IG method. When Tables 6–9 are analyzed, it can be observed that the IG–GA method shows a higher classifier accuracy in comparison to IG or the IG–PCA method.

With respect to the classifiers' performances, the C4.5 decision tree algorithm shows a higher performance than the KNN algorithm. Consequently, it is seen that a higher classifier performance is acquired with fewer features through hybrid methods.

4. Conclusion

In this study, a two-stage feature selection and feature extraction is used to reduce the high dimensionality of a feature space composing of a large number of terms, remove redundant and irrelevant features from the feature space and thereby improve the performance of text categorization. In the first stage, each term within the text is ranked depending on their importance for the classification using the IG method classification. In the second stage, the GA and PCA feature selection and feature extraction methods are applied separately on the terms, which are ranked in decreasing order of importance, and a dimension reduction is carried out. Thereby, during the text categorization, terms with less importance are ignored, feature selection and feature extraction methods are applied on the terms with highest importance, and the computational time and complexity of the method are reduced. To evaluate the effectiveness of the dimension reduction methods on our proposed model, experiments are conducted using the KNN and C4.5 decision tree algorithms on the Reuters-21,578 and Classic3 datasets collection for text categorization. As a result of the experimental studies, it is seen that using features reduced via dimension techniques instead of all features positively contributed to classifier performance. When there are many irrelevant or redundant features in the feature space, performing a feature selection method could remove them, and thus the classifier performance can be improved. Also, it is revealed that the success of text categorization performed through the C4.5 decision tree and KNN algorithms using fewer features selected via IG-PCA and IG-GA is higher than the success acquired using features selected via IG. Two-stage feature selection methods can improve the performance of text categorization. That is to say, the dimension reduction carried out via a GA and PCA by denoting the features of the highest importance determined via IG increased the text categorization success. Consequently, a higher classifier performance is acquired with fewer features through a two-stage feature selection method.

Acknowledgement

This study has been supported by Scientific Research Project of Selcuk University.

References

[1] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, Text feature selection using ant colony optimization, Expert Systems with Applications 36 (2009) 6843–6853.
[2] M.G.H. AlZamil, A.B. Can, ROLEX-SP: rules of lexical syntactic patterns for free text categorization, Knowledge-Based Systems 24 (2011) 58–65.
[3] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrival, Addison-Wesley, 1999.
[4] Y.-C. Chang, S.-M. Chen, C.-J. Liau, Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method, Expert Systems with Applications 34 (2008) 1948–1953.
[5] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.
[6] F.J. Damerau, T. Zhang, S.M. Weiss, N. Indurkhya, Text categorization for a comprehensive time-dependent benchmark, Information Processing and Management 40 (2004) 209–221.
[7] M.E. ElAlami, A filter model for feature subset selection based on genetic algorithm, Knowledge-Based Systems 22 (2009) 356–362.
[8] L. Ferr, Selection of components in principal component analysis: a comparison of methods, Computing and Statistical Data Analysis 19 (1995) 669–682.
[9] N. Fuhr, C. Buckley, A probabilistic learning approach for document indexing, ACM Transactions on Information Systems 9 (3) (1991) 223–248.
[10] M. Gen, R. Cheng, Genetic Algorithms and Engineering Optimization, vol. 68, Wiley Interscience Publication, 2000.
[11] D.E. Goldberg, Genetic Algorithm in search, optimization and machine learning, Addison-Wesley, Reading, MA, 1989.
[12] J. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, 1975.
[13] T. Joachims, A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization, in: Proceedings of Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, 1997, pp. 143–151.
[14] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceedings of the 10th European Conference on Machine Learning, pp. 137–142.
[15] T. Jolliffe, Principal Component Analysis, ACM Computing Surveys, Springer-Verlag, 1986. pp. 1–47.
[16] S.L.Y. Lam, D.L. Lee, Feature reduction for neural network based text categorization, in: Sixth International Conference on Database Systems for Advanced Applications (DASFAA'99), 1999, p. 195.
[17] W. Lam, Y. Han, Automatic textual document categorization based on generalized instance sets and a metamodel, Proceeding of the IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 628–633.
[18] D.D. Lewis, Reuters-21578 text categorization test collection, distribution 1.0. <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>, 1997.
[19] C.H. Li, S.C. Park, Combination of modified BPNN algorithms and an efficient feature selection method for text categorization, Information Processing and Management 45 (2009) 329–340.
[20] Y. Li, D.F. Hsu, S.M. Chung, Combining multiple feature selection methods for text categorization by using rank-score characteristics, in: 21st IEEE International Conference on Tools with Artificial Intelligence, 2009, pp. 508–517.
[21] H. Liu, V. Keselj, Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests, Data and Knowledge Engineering 61 (2007) 304–330.
[22] L. Liu, J. Kang, J. Yu, Z. Wang, A comparative study on unsupervised feature selection methods for text clustering, in: Proceeding of NLP-KE'O5, 2005, pp. 597–601.
[23] A. McCallum, K. Nigam, A comparison of event models for Naive Bayes text classification, in: AAAI'98 Workshop on Learning for Text Categorization, 1998, pp. 41–48.
[24] T. Mitchel, Machine Learning, McGraw-Hill, New York, 1997.
[25] V. Mitra, C.-J. Wang, S. Banerjee, Text classification: a least square support vector machine approach, Applied Soft Computing 7 (2007) 908–914.
[26] M.F. Porter, An algorithm for suffix stripping, Program (Automated Library and Information Systems) 14 (3) (1980) 130–137.
[27] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.
[28] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing and Management 24 (5) (1988) 513–523.
[29] F. Sebastiani, A tutorial on automated text categorisation, in: Proceedings of the ASAI-99, in: 1st Argentinian Symposium on Artificial Intelligence, Buenos Aires, AR., 1999, pp. 17–35.

[30] A. Selamat, S. Omatu, Web page feature selection and classification using neural Networks, Information Sciences 158 (2004) 69–88.
[31] N. Slonim, N. Tishby, Document clustering using word clusters via the information bottleneck method, in: Proceedings of SJGIR'00, 2000, pp. 208–215.
[32] W. Song, S.C. Park, Genetic algorithm for text clustering based on latent semantic indexing, Computers and Mathematics with Applications 57 (2009) 1901–1907.
[33] J.-T. Sun, Z. Chen, H.-J. Zeng, Y. Lu, C.-Y. Shi, W.-Y. Ma, Supervised latent semantic indexing for document categorization, in: ICDM, IEEE Press, 2004, pp. 535–538.
[34] S. Tan, An effective refinement strategy for KNN text classifier expert, Systems with Applications 30 (2) (2006) 290–298.
[35] S. Valle, W. Li, S.J. Qin, Selection of the number of principal components: the variance of the reconstruction error criterion with a comparison to other methods, Ind, Engineering Chemistry Research 38 (1999) 4389–4401.
[36] C.J. Van Rijsbergen, Information Retrieval, second ed., Buttersworth, London, 1979.
[37] K. Warne, G. Prasad, S. Rezvani, L. Maguire, Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion, Engineering Applications of Artificial Intelligence 17 (2004) 871–885.
[38] N. Wyse, R. Dubes, A.K. Jain, A critical evaluation of intrinsic dimensionality algorithms, Pattern Recognition in Practice (1980) 415–425.
[39] Y. Yang, An evaluation of statistical approaches to text categorization, Information Retrieval 1 (1) (1997) 76–88.
[40] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 412–420.
[41] B. Yu, Z. Xu, C. Li, Latent semantic analysis for text categorization using neural network, Knowledge-Based Systems 21 (2008) 900–904.
[42] M.L. Zhang, Z.H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge and Data Engineering 18 (10) (2006) 1338–1351.
[43] Y.X. Zhang, Artificial neural networks based on principal component analysis input selection for clinical pattern recognition analysis, Talanta 73 (2007) 68–75.
[44] W. Zhang, T. Yoshida, X. Tang, Text classification based on multi-word with support vector machine, Knowledge-Based Systems 21 (2008) 879–886.
[45] W. Zhao, Y. Wang, D. Li, A dynamic feature selection method based on combination of GA with *K*-means, in: 2nd International Conference on Industrial Mechatronics and Automation, 2010, pp. 271–274.
[46] C. Zifeng, X. Baowen, Z. Weifeng, J. Dawei, X. Junling, CLDA: feature selection for text categorization based on constrained LDA, in: International Conference on Semantic Computing (ICSC 2007), 2007, pp. 702–712.