Expert Systems With Applications 57 (2016) 232–247

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Ensemble of keyword extraction methods and classifiers in text classification

Aytuğ Onan[a,*], Serdar Korukoğlu[b], Hasan Bulut[b]

[a] Celal Bayar University, Department of Computer Engineering, 45140 Muradiye, Manisa, Turkey
[b] Ege University, Department of Computer Engineering, 35100 Bornova, Izmir, Turkey

ARTICLE INFO

Article history:
Received 4 January 2016
Revised 22 March 2016
Accepted 26 March 2016
Available online 29 March 2016

Keywords:
Keyword extraction
Text classification
Ensemble learning
Scientific text classification

ABSTRACT

Automatic keyword extraction is an important research direction in text mining, natural language processing and information retrieval. Keyword extraction enables us to represent text documents in a condensed way. The compact representation of documents can be helpful in several applications, such as automatic indexing, automatic summarization, automatic classification, clustering and filtering. For instance, text classification is a domain with high dimensional feature space challenge. Hence, extracting the most important/relevant words about the content of the document and using these keywords as the features can be extremely useful. In this regard, this study examines the predictive performance of five statistical keyword extraction methods (most frequent measure based keyword extraction, term frequency-inverse sentence frequency based keyword extraction, co-occurrence statistical information based keyword extraction, eccentricity-based keyword extraction and TextRank algorithm) on classification algorithms and ensemble methods for scientific text document classification (categorization). In the study, a comprehensive study of comparing base learning algorithms (Naïve Bayes, support vector machines, logistic regression and Random Forest) with five widely utilized ensemble methods (AdaBoost, Bagging, Dagging, Random Subspace and Majority Voting) is conducted. To the best of our knowledge, this is the first empirical analysis, which evaluates the effectiveness of statistical keyword extraction methods in conjunction with ensemble learning algorithms. The classification schemes are compared in terms of classification accuracy, *F*-measure and area under curve values. To validate the empirical analysis, two-way ANOVA test is employed. The experimental analysis indicates that Bagging ensemble of Random Forest with the most-frequent based keyword extraction method yields promising results for text classification. For ACM document collection, the highest average predictive performance (93.80%) is obtained with the utilization of the most frequent based keyword extraction method with Bagging ensemble of Random Forest algorithm. In general, Bagging and Random Subspace ensembles of Random Forest yield promising results. The empirical analysis indicates that the utilization of keyword-based representation of text documents in conjunction with ensemble learning can enhance the predictive performance and scalability of text classification schemes, which is of practical importance in the application fields of text classification.

1. Introduction

Automatic keyword extraction is the process of identifying key terms, key phrases, key segments or keywords from a document that can appropriately represent the subject of the document (Beliga, Mestrovic, & Martincic-Ipsic, 2015). The Web is a very rich source of information which is progressively expanding. Hence, the number of digital documents available has been progressively expanding and the manual keyword extraction can be an infeasible task. Keyword extraction is an important research direction in text mining, natural language processing and information retrieval. Since keyword extraction provides a compact representation of the document, many applications, such as automatic indexing, automatic summarization, automatic classification, automatic clustering, and automatic filtering can benefit from the keyword extraction process (Zhang et al., 2008).

Automatic keyword generation process can be broadly divided into two categories as keyword assignment and keyword extraction (Siddiqi & Sharan, 2015). In keyword assignment, a set of possible keywords is selected from a controlled vocabulary of words, whereas keyword extraction identifies the most relevant words

* Corresponding author. Tel.: +90 232 3887221, +90 544 810 70 80; fax: +90 232 3399405.
E-mail addresses: aytug.onan@cbu.edu.tr, aytugonan@hotmail.com (A. Onan), serdar.korukoglu@ege.edu.tr (S. Korukoğlu), hasan.bulut@ege.edu.tr (H. Bulut).

http://dx.doi.org/10.1016/j.eswa.2016.03.045
0957-4174/

available in the examined document (Beliga et al., 2015). Keyword extraction methods can be broadly grouped into four categories as statistical approaches, linguistic approaches, machine learning approaches and other approaches (Han & Kamber, 2006).

Text classification is an important subfield of text mining which assigns a text document into one or more predefined classes or categories. Several forms of text collections, such as news articles, digital libraries and Web pages are important sources of information (Han & Kamber, 2006). Hence, text classification is an important research direction in library science, information science and computer science (Jain, Raghuvanshi, & Shrivastava, 2012). Many applications of text mining can be modelled as a text classification problem. These applications include news filtering, organization, document organization, retrieval, opinion mining (sentiment analysis), and spam filtering (Aggarwal & Zhai, 2012).

High dimensional feature space is a typical challenge of text classification applications (Joachims, 2002). When all the words of the training documents are used as the features, text classification process becomes computationally intensive task (Onan & Korukoğlu, 2015). Hence, keywords of a text collection, which are the most important/relevant words about the content of the documents, can be good candidates to select as features in classification model construction (Liu & Wang, 2007; Rossi, Maracini, & Rezende, 2014). Machine learning algorithms, such as Naïve Bayes, k-nearest neighbour algorithm, support vector machines and artificial neural networks, have been successfully applied in classifying text documents (Sebastiani, 2002). Ensemble methods are a set of learning algorithms, which combine the decisions of these algorithms so that a more robust classification model can be built with higher predictive performance (Dietterich, 2000).

Considering these issues, this paper examines the effectiveness of statistical keyword extraction methods, base learning algorithms and ensemble methods in scientific text document classification. To the best of our knowledge, this is the first attempt, which empirically evaluates the effectiveness of statistical keyword extraction methods in conjunction with ensemble learning algorithms. In comparative evaluation, five popular ensemble methods (Boosting, Bagging, Dagging, Random Subspace and Voting) are utilized. Naïve Bayes algorithm, support vector machines, logistic regression and Random Forest algorithm are utilized as the base learning algorithms. In the experimental analysis, the domain independent statistical keyword extraction framework proposed in (Rossi et al., 2014) is utilized. In summary, the experimental study aims to answer the following research questions:

(1) Which configuration of statistical keyword extraction, classification and ensemble learning algorithms yield the highest performance in scientific text document classification?
(2) Is there an optimal number of keywords to represent the text documents and which number of keywords obtains promising results?

To the best of our knowledge, this is the first extensive empirical analysis which examines the predictive performance of statistical keyword extraction methods in conjunction with ensemble learning algorithms. The presented classification scheme, which integrates Bagging ensemble of Random Forest with the most-frequent based keyword extraction method, yields very promising results on scientific text classification. The rest of this paper is organized as follows. Section 2 briefly reviews the literature on keyword extraction and ensemble methods. Section 3 presents the statistical keyword extraction methods utilized in the experimental evaluations. Section 4 briefly describes the classification algorithms and Section 5 describes the ensemble learning methods. Section 6 presents the experimental results, discussion and statistical analysis of empirical results on ACM document collection. Section 7 presents the results of ensemble classification schemes on a larger text document collection. Finally, Section 8 presents the concluding remarks.

2. Literature review

This section briefly reviews the literature on keyword extraction methods and the ensemble methods.

2.1. Related work on keyword extraction

In statistical keyword extraction methods, statistical measures, such as n-gram statistics, word frequency and TF-IDF measure are utilized to identify keywords. The statistical keyword extraction methods can be domain-independent and do not require training data (Beliga et al., 2015). Matsuo and Ishizuka (2003) presented a statistical keyword extraction method from a single document. Initially, frequent terms are extracted. Then, co-occurrence between each term and the frequent terms are evaluated. Based on the co-occurrence distributions, the significance of a term in the document is determined. The method does not require a training corpus and can yield comparable results to TF-IDF measure. Turney (2003) presented an improved key phrase extraction algorithm, which uses statistical association among the key phrases to improve the coherence of the obtained keywords. In order to measure the association between key phrases, web mining is utilized. In another statistical keyword extraction method, text document is represented as an undirected graph (Palshikar, 2007). The vertices of the graph contains words of the document, whereas the edges are assigned values based on a statistical measure of dissimilarity between the two words.

In linguistic approaches, linguistic features of the document are utilized to identify keywords. These include lexical, syntactic, semantic and discourse analysis (Zhang et al., 2008). The linguistic keyword extraction methods are domain-dependent (Siddiqi & Sharan, 2015). Hulth (2003) examined the incorporation of linguistic knowledge, such as syntactic features to the keyword extraction process. The experimental results indicated that linguistic features can obtain improvements over the use of only statistical measures, such as term frequency or n-grams. HaCohen-Kerner (2003) presented a keyword extraction model from abstracts and titles. In the model, text representation schemes, such as unigrams, bigrams and trigrams are utilized. Nguyen and Kan (2007) presented a key phrase extraction algorithm from scientific publications. In this method, linguistic features, such as the positions of phrases in the text documents, salient morphological phenomena are taken into account. Krapivin, Autayeu, Marchese, Blanzieri, and Segata (2010) incorporated natural language processing methods to automatic key phrase extraction from scientific papers to enhance the performance of machine learning algorithms, such as support vector machines and Random Forests. The experimental results are obtained on ACM dataset. The evaluations are done with expert-assigned key phrases and key phrase extraction algorithm (KEA).

In machine learning approaches, a learning algorithm, such as support vector machines, Naïve Bayes, decision tree, is used to construct a classification model. In model construction, a training set of documents with tags are used and the model is validated via a test set of documents. The drawback of the machine learning based feature extraction models is the need to obtain a tagged set of documents. Witten, Paynter, Frank, Gutwin, and Nevill-Manning (1999) presented a simple and efficient key phrase extraction algorithm (KEA) which utilizes Naïve Bayes algorithm for domain-based key phrase extraction. In this method, possible key phrases are determined by lexical methods and good key phrases are obtained by the machine learning algorithm. HaCohen-Kerner, Gross, and Masa (2005) examined the effectiveness of several automatic

key phrase extraction and learning methods on scientific articles in English. The key phrase extraction methods are evaluated by different machine learning methods and the experimental results indicated that C4.5 algorithm yields the highest predictive performance for the domain. Medelyan and Witten (2006) presented an improved automatic key phrase extraction model called KEA++. In this model, semantic information of terms and phrases obtained from domain-specific thesaurus is utilized to automatically extract key phrases. Zhang et al. (2008) modelled the keyword extraction problem as a string labelling task and utilized conditional random fields method to label. The experimental results indicated that conditional random fields method yields better results in keyword extraction compared to conventional machine learning algorithms, such as support vector machines and linear regression.

Graph-based methods are unsupervised keyword extraction methods. Mihalcea and Tarau (2004) presented a graph-based text processing, keyword and sentence extraction algorithm, called TextRank. In TextRank algorithm, undirected/directed weighted co-occurrence networks with changing window sizes are utilized. In order to extract keywords by TextRank algorithm, the text is divided into tokens. Then, part of speech tags are assigned to the tokens. Then, a node is created to represent each token or some tokens corresponding to the particular part of speech tag. For the co-occurring words of a particular window, a link is constructed between the two nodes (Seifert, Ulbrich, & Granitzer, 2011). Litvak and Last (2008) examined the performance of supervised and unsupervised graph-based keyword extraction methods for the extractive summarization of the documents. In order to represent text documents, graph-based syntactic representation is utilized. Grineva, Grinev, and Lizorkin (2009) presented a key term extraction model which models text documents as a graph of semantic relationships between terms. In the constructed model, the most relevant terms about the topics of the document tend to build densely interconnected components, whereas non-relevant terms about the topics tend to construct weakly interconnected components. In order to obtain thematic partitions from the graph structure, a graph community detection technique is utilized. In the method, information obtained from Wikipedia is utilized to weight terms and examine the semantic relationships among them.

In another study, Huan, Tian, Zhou, Ling, and Huang (2006)) presented an automatic key phrase extraction algorithm, which can be used in both unsupervised and supervised learning tasks. The presented algorithm models each text document as a semantic network. The key phrases are extracted based on the structural dynamics of the semantic network. A recent study on keyword extraction presented a model based on fractal patterns (Najafi & Darooneh, 2015). The results indicate that the most relevant terms about the topic of the text document have fractal dimensions different from one, whereas unimportant terms have a fractal dimension value of one. Based on this observation, the significance of words is determined based on fractal dimensions.

2.2. Related work on the ensemble methods

Ensemble learning is a promising research direction of machine learning research. Ensemble methods have been successfully utilized in a wide range of applications, e.g. predicting stock returns (Tsai, Lin, Yen, & Chen, 2011), bankruptcy prediction (Kim & Kang, 2012), credit scoring (Abellan & Mantas, 2014), activity recognition (Catal, Tufekci, Pirmit, & Kocabag, 2015), microarray gene expression data classification (Reboiro-Jato, Diaz, Glez-Penaa, & Fdez-Riverola, 2014), sentiment analysis (De Silva, Hruschka, & Hruschka, 2014), and intrusion detection (Aburomman & Reaz, 2016). The rest of this section presents the review of classifier ensembles with a special emphasis on text classification.

Prabowo and Thelwall (2009) examined the performance of ensemble learning in text sentiment classification. In the empirical analysis, the general inquirer based classifier, rule-based classifier, statistics based classifier, induction-based classifier and support vector machines are combined in several different ways. The experimental results indicate that ensemble learning can enhance the predictive performance of classifiers in text classification. In another study, the predictive performance of ensemble learning for sentiment classification has been examined by taking different feature sets (such as part of speech information, world relation features and feature weighting schemed) into account (Xia, Zong, & Li, 2011). In this scheme, Naïve Bayes, maximum entropy and support vector machines were employed as the base-learners. To combine the base-learners, three ensemble strategies (the fixed combination, the weighed-combination and meta-classifier combination) were employed. Similarly, De Silva et al. (2014) analysed the performance of two feature representation schemes (bag-of-words and feature hashing-based representation) in sentiment analysis on Twitter data. In the empirical analysis, different classification schemes are obtained with the combination of lexicons, bag-of-words, emoticons and feature hashing. The experimental results indicate that the classifier ensemble formed by Naïve Bayes, support vector machines, Random Forest and logistic regression can improve the predictive performance of classification model. Another study examined the predictive performance of three ensemble learning algorithms (Bagging, Random Subspace and Boosting) on five base-learners (Naïve Bayes, maximum entropy, decision tree, k-nearest neighbour and support vector machines) for sentiment classification. The experimental analysis indicate that Random Subspace yields better predictive performance on text sentiment classification (Wang, Sun, Ma, Xue, & Gu, 2014). Fersini, Messina, and Pozzi (2014) presented a Bayesian model averaging based ensemble method for sentiment classification. In this scheme, ensemble learning was employed to reduce the noise sensitivity in text sentiment. Besides, a heuristic combination strategy was proposed to determine the classifiers to be included in the classifier ensemble. Yang, Zhang, and Li (2011) presented an ensemble learning based approach for classifying text streams. To eliminate the need for the manual labelling of text documents, keywords were used as features. In this scheme, the base-learners were built by keywords and unlabelled documents. Besides, an ensemble algorithm was presented to deal with concept drifting of text data streams. Zhang and He (2015) presented an ensemble classification scheme for sentence-level polarity classification. In this scheme, the latent topic feature set and word embedding based feature representation methods were employed to deal with data sparseness problem encountered in short text documents. The ensemble learning based method was employed to enhance the performance of polarity classification scheme.

2.3. Motivations and limitations of the study

The high dimensional feature space is a typical challenge of text classification task. The classification algorithms based on keyword extraction for text can be a very promising solution. The main motivation of this study is to empirically evaluate the predictive performance of classification algorithms and ensemble learning methods when the scientific text documents are represented by a set of keywords. The high dimensional feature space involves representing a text document in a compact and efficient way. The development of efficient text classification algorithms based on keyword-based representation may enhance the predictive performance and scalability of the classification algorithms, which is of practical importance in the application fields of text classification. In this study, the performances of five statistical keyword extraction methods are examined. The methods utilized in the empirical

analysis are selected among the domain-independent keyword extraction methods, which have comparable high performance and do not involve using a corpus. In the empirical analysis, five classification algorithms (Naïve Bayes, support vector machines, logistic regression and Random Forest) are utilized as the base learning algorithms. Machine learning classifiers that are utilized in the empirical analysis are selected from different machine learning groups (Kotsiantis, Zaharakis, & Pintelas, 2006). Hence, the diversity of the ensemble was assured by the utilization of different models.

The presented research aims to identify whether there is an optimal number of keywords to represent text documents. However, the 20 different number of keywords ranging from 5 to 100 are considered in the experimental analysis. In addition, there are many different keyword extraction methods from several categories, such as statistical, linguistic and machine learning based approaches. The experimental analysis presented in this paper, however, considers only the performance of statistical keyword extraction methods. The performance comparison of the methods from diverse keyword extraction methods can also be a good research direction.

To summarize, the ensemble of different feature selection methods, classifiers and ensemble methods is a promising research direction in machine learning. In this regard, there are a number of works devoted to the empirical analysis of the ensemble learning methods in other text classification fields, such as sentiment analysis (Wang et al., 2014; Xia et al., 2011). The literature review on the ensemble learning methods with a special emphasis on text classification is presented in Section 2.2. The predictive performance of different classifiers, different representation schemes (such as bag of words, part of speech information, world relation features, etc.) and different ensemble learning methods (such as fixed combination, the weighed-combination and meta-classifier combination, Bagging, Random Subspace and Boosting) have been extensively analysed in the literature (De Silva et al., 2014; Wang et al., 2014; Xia et al., 2011). As mentioned in advance, this is the first comprehensive experimental analysis of statistical keyword extraction-based text representation in conjunction with ensemble learning algorithms.

3. Keyword extraction methods

Keyword extraction methods can be broadly divided into two categories as domain-dependent and domain-independent keyword extraction methods. Domain-dependent keyword extraction methods require to keep track of all the words within the text collection, whereas the domain-independent keyword extraction methods do not require the analysis of the entire text collections (Rossi et al., 2014). Domain-independent keyword extraction methods can have comparable high performance and do not require using of a corpus (Matsuo & Ishizuka, 2003). In contrast, domain-dependent keyword extraction methods can be extremely costly for large text collections. Term frequency-inverse document frequency (TF-IDF), mutual information and log-likelihood are some representatives of domain-dependent keyword extraction methods and term frequency-inverse sentence frequency (TF-ISF), co-occurrence statistical information (CSI), TextRank and eccentricity-based keyword extraction methods are some representatives of domain-independent keyword extraction methods. In the experimental analysis, five statistical keyword extraction methods are taken into account. These methods include most frequent based keyword extraction, term frequency-inverse sentence frequency (TF-ISF) based keyword extraction, co-occurrence statistical information based keyword extraction (CSI), eccentricity-based keyword extraction (EB) and TextRank algorithm based keyword extraction (TR). The rest of this section briefly describes these methods.

3.1. Most frequent based keyword extraction

The most frequent based keyword extraction method (MF) identifies the most frequently occurring terms of the text document as the keywords. In order to represent text documents, sentence-term matrix is utilized. In this representation, a term's occurrence value for a particular sentence takes the value of one or zero depending on the term's occurrence or non-occurrence on that sentence, respectively. Based on this representation, a frequency score is calculated for each term by computing the number of occurrences of the term in the matrix. For each term (t_k), the frequency score is calculated as given by Eq. (1) (Rossi et al., 2014):

$$MF(t_k) = \sum_{s_l \in S} occurence_{t_k, s_l} \tag{1}$$

where $occurence_{t_k, s_l}$ denotes a particular term's occurrence in a sentence s_l.

3.2. Term frequency-inverse sentence frequency

Term frequency-inverse sentence frequency (TF-ISF) is a statistical measure, which is an adaptation of the term frequency-inverse document frequency (TF-IDF) measure to the sentences of text documents. In term frequency-inverse sentence frequency based keyword extraction, each sentence of the text document is regarded as a vector of TF-ISF weights (Neto, Santos, Kaestner, & Freitas, 2000). TF-ISF measure of a term t_k in a sentence s is computed by the product of the frequency of the term by inverse sentence frequency of the term as given by Eqs. (2) and (3) (Fiori, 2014):

$$TF{-}ISF(S) = \sum_{t \in S} freq(t_k) \times isf(t_k) \tag{2}$$

$$isf(t_k) = 1 - \frac{\log(n(t_k))}{\log(n)} \tag{3}$$

where n denotes the number of sentences in the text collection.

3.3. Co-occurrence statistical information based keyword extraction

Co-occurrence statistical information based keyword extraction (CSI) method aims to identify important terms based on the distribution of words co-occurrences in the same sentences and the distribution within the general distribution (Matsuo & Ishizuka, 2004). In this method, frequent terms are identified first. Then, co-occurrences in the same sentences are determined by considering co-occurrences between each term and the frequent terms. Co-occurrence distribution is utilized to determine the importance of a particular term in the text document. In order to measure the degree of bias of a probability distribution between a term t_k and the frequent terms, χ^2 (chi-square) statistics is used (Matsuo & Ishizuka, 2003) which is computed as given by Eq. (4):

$$\chi^2(w) = \sum_{g \in G} \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g} \tag{4}$$

where p_g denotes the expected probability of a frequent term $g \in G$, n_w denotes the total number of co-occurrences of term w and frequent terms G (Matsuo & Ishizuka, 2003, 2004). In order to improve the performance of basic χ^2 statistics based keyword extraction scheme, two further refinements are presented. First, a more robust variant of χ^2 statistics is formulated to handle with sentences of variety of lengths properly. Second, a clustering of terms is obtained based on co-occurring terms. Text documents can have the sentences with varying lengths. The length of the sentence may influence the co-occurrence relation between terms and frequent terms. Terms of longer sentences tend to co-occur with many terms. In contrast, the terms of shorter sentences tend

to co-occur with a few terms. In addition, a term, which co-occurs, with another term may take a higher value for χ^2 statistics owing to adjunct terms which are not really important (Matsuo & Ishizuka, 2004). Hence, in order to enhance the robustness of χ^2 statistics, the two aforementioned issues should be taken into consideration. To enhance the performance of χ^2 statistics, a more robust χ^2 statistics denoted by χ'^2 is formulated as given by Eq. (5):

$$\chi'^2(w) = \chi^2(w) - \max_{g \in G}\left(\frac{(freq(w,g) - n_w p_g)^2}{n_w p_g}\right) \quad (5)$$

In general, co-occurrence statistical information based keyword extraction (CSI) method consists of six stages (Matsuo & Ishizuka, 2004). First, stemming is applied on the text document and Apriori algorithm is utilized to extract phrases. Then, frequent terms are selected from the extracted phrases. In the third stage, the frequent terms are clustered based on Jensen–Shannon divergence (Matsuo & Ishizuka, 2004). Then, expected probability is computed. Next, χ'^2 statistics is computed for each term as given by Eq. (5). Finally, a number of keywords with the highest χ'^2 statistics value are extracted as the keywords.

3.4. Eccentricity based keyword extraction

In eccentricity based keyword extraction (EB) (Palshikar, 2007), graph-theory based concepts of vertex centrality is utilized to model keyword extraction problem. In this scheme, text documents are represented as an undirected, edge-labelled graph whose vertices are words of the text document, whereas the edges correspond to dissimilarity measure computed between two words. Based on the assumption of having the most relevant keywords about the content of the document in the central vertices, the importance of words are modelled based on their centrality in the graph. In this method, eccentricity based centrality measure is utilized to extract keywords. A term graph is an undirected, edge-labelled graph $G=(V, E, w)$ where each vertex in V represents a particular term of the document, E represents the set of edges and w represents the edge weight function (Palshikar, 2007). For a term graph G, the distance between two terms t_k and t_l (denoted by $d(t_k, t_l)$ is computed by summing the edge weights on the shortest path from t_k and t_l in G. Eccentricity of a term t_k is the maximum distance from t_k to any other terms in G. In order to compute the eccentricity of a given vertex to all other vertices, Dijkstra's single source shortest path algorithm is utilized. Words with the lower eccentricity values can be considered to be more important. Hence, words are ordered based on their eccentricity values and the words with lower eccentricity values are extracted as the keywords. The presented graph-based keyword extraction scheme is utilized in conjunction to eccentricity based measure. However, other centrality measures, such as closeness and proximity can also be used in this scheme (Palshikar, 2007).

3.5. TextRank algorithm

TextRank algorithm (TR) (Mihalcea & Tarau, 2004) is a graph-based ranking model for text processing. TextRank algorithm can be utilized in a number of natural language processing tasks, such as keyword extraction and sentence extraction from text documents. Graph-based ranking methods aim to identify the importance of a vertex with a graph based on the information obtained recursively from the graph. For extracting keywords with TextRank algorithm, the text is tokenized and the part of speech tags are assigned to each token. Then, for all lexical units that pass the syntactic filter, a node is generated in the graph and an edge is generated between two nodes if the words are co-occurring within a particular window of words. In this way, an undirected and unweighted graph representation is obtained from text. On this graph structure, TextRank graph-based ranking model is applied. In the algorithm, the score for each term is determined based on the votes that are cast for it and the score of the terms that are casting the votes. The score for each term t_i is calculated as given by Eq. (6) (Mihalcea & Tarau, 2004):

$$S(t_i) = (1-d) + d \times \sum_{j \in In(t_i)} \frac{1}{|Out(t_j)|} S(t_j) \quad (6)$$

where $In(t_i)$ denotes the set of vertices that point to it, $Out(t_i)$ denotes the set of vertices that term t_i points to and d denotes a constant factor which is usually set to 0.85.

4. Classification algorithms

Machine learning algorithms have been successfully utilized in text classification. Machine learning classifiers can be broadly classified as decision trees (such as C4.5, ID3 and Random Forest), rule-based methods (such as RIPPER, PART and genetic algorithms), perceptron-based methods (such as artificial neural networks, radial basis function networks), statistical learning methods (such as Bayesian Networks and Naïve Bayes classifier), instance-based classifiers (such as k-nearest neighbour algorithm) and support vector machines (Kotsiantis et al., 2006). Naïve Bayes classifier, support vector machines and decision trees are widely employed for text classification problems (Ikonomakis, Kotsiantis, & Tampakas, 2005). Hence, Naïve Bayes, support vector machines, logistic regression and Random Forest algorithm are taken as the base learners in the experimental evaluations.

4.1. Naive Bayes classifier

Naïve Bayes algorithm (NB) is a statistical learning algorithm which is based on Bayes' theorem. The algorithm has a clear semantics in representing, using and learning probabilistic knowledge (Han & Kamber, 2006). It is based on the class conditional independence assumption which simplifies the required computational cost. Hence, the algorithm can scale well and can be easily employed in a number of domains, including text mining. The algorithm yields high accuracy and speed on large datasets and gives comparable results to other classification algorithms, such as decision trees and neural networks (John & Langley, 1995).

4.2. Support vector machines

Support vector machines (SVM) are classification algorithms that can be used to classify both linear and non-linear data (Vapnik, 1995). In support vector machines, a non-linear matching method is applied so that original data set is transformed into a higher dimension. In this dimension, a hyperplane, which can appropriately partition the data, is examined. This hyperplane is the decision boundary for partitioning the data into classes (Joachims, 1998). The main objective of support vector machines is to identify an optimal decision boundary that can be used to classify different classes. Text mining usually involves high dimensional feature space, few irrelevant features, sparse document vectors and linearly separable categorization (Han & Kamber, 2006). Hence, support vector machines are viable classification algorithms for text classification.

4.3. Logistic regression

Logistic regression (LR) is a linear classification algorithm. In the logistic regression, the probability of some event's happening is modelled as a linear function of a set of predictor variables (Kantardzic, 2011). Linear regression can perform well. Yet,

the membership values obtained by the linear regression cannot be always in the range of 0–1. Hence, these values are not in appropriate range of probabilities. Besides, least-squares regression assumes that errors are both statistically independent and normally distributed with the same standard deviations (Witten, Frank, & Hall, 2011). Logistic regression gets rid of the aforementioned problems and obtains a linear model based on transformed target variable.

4.4. Random Forest algorithm

Random Forest algorithm (RF) is an ensemble of classification and regression trees induced from bootstrap samples of the training data (Breiman, 2001). In the algorithm, the generalization error of the classifier depends on the power of the individual trees and the association between trees. In the tree induction process, a random feature selection is utilized, which enhances the ability of the model to deal with noisy or irrelevant data. The algorithm yields comparable results to AdaBoost algorithm.

5. Ensemble methods

Ensemble methods are popular research directions in machine learning and pattern recognition (Onan, 2016; Ranawana & Palade, 2006). Ensemble methods aim to combine decisions from a set of weak learning algorithms (base learners) so that the accuracy and robustness of the built classification model can be enhanced. The generalization ability of ensemble methods is better compared to the single base learners. There are statistical, computational and representational reasons to build multiple classifier systems (Rokach, 2010). Ensemble methods can be broadly assigned into two groups as dependent and independent methods based on the utilized structure in the construction of ensembles (Kuncheva, 2014). In dependent methods, the output of a base learner is used to construct the next classifier. Hence, the knowledge obtained in the earlier iterations can be transferred to the learning of the later iterations. In contrast, the base learners in independent methods are constructed independently and the outputs of these learning algorithms are combined via a combination method, such as majority voting or meta-learning methods (Breiman, 1996). In the experimental evaluations, five ensemble methods are considered. These methods are AdaBoost algorithm, Bagging algorithm, Dagging algorithm, Random Subspace algorithm and Voting scheme. These methods are briefly discussed here.

5.1. AdaBoost algorithm

Boosting algorithm is a widely employed ensemble learning method to enhance the predictive performance of weak learning algorithms. In this method, a weak learning algorithm is ran recursively on the different sampling distributions of the training data. In this way, a single robust classification model can be built from the weak learning algorithms. AdaBoost (adaptive boosting) algorithm (Kuncheva, 2014) is an ensemble method which improves the boosting algorithm by an iterative process in which more focus is dedicated to the difficult patterns. First, all the patterns in the training set are assigned the same weight value. During the process, the weight values for misclassified instances are increased whereas the weights for correctly classified instances are decreased. In this way, the weak learning algorithm dedicates more iterations and classifiers to the patterns that are harder to classify (Rokach, 2010). The stages of AdaBoost algorithm are outlined in Fig. 1.

5.2. Bagging algorithm

Bagging (Bootstrap aggregating) (Breiman, 1996) is an ensemble method which aims to build a robust/improved composite classifier with high predictive performance by combining the classifiers trained on different training sets. The general structure of the algorithm is summarized in Fig. 2. In this method, each weak learning algorithm is trained on a different training set obtained by a replacement from the training set, where sizes of samples are kept equal to the size of the original training set. In order to obtain new training sets, the simple random sampling with replacement is utilized. This method yields the diversity required for the ensemble learning. The results of the individual classifiers are combined by majority voting or weighted majority voting.

5.3. Dagging algorithm

Dagging algorithm (Ting & Witten, 1997) is an ensemble method. As in the case of Bagging algorithm, the method aims to obtain a robust classification model by combining the weak learners trained on different samples of the training set. The method has a similar structure for training and classification phase as summarized in Fig. 2. Dagging method, however, uses disjoint, stratified samples instead of bootstrapping. The method is an effective method when the individual classifiers have a bad time complexity. In Dagging algorithm, the outputs of the weak learning algorithms are combined via the majority voting combination rule.

5.4. Random Subspace algorithm

Random Subspace algorithm (Ho, 1998) is an ensemble learning method which combines multiple classifiers trained on randomly selected feature subspaces. The method aims to avoid overfitting while enhancing the predictive performance. Bagging algorithm builds a robust classification model by combining the weak learners that are trained on different samples of the training set. This is also valid for Random Subspace algorithm, but instead of instance space, different samples are obtained on the feature space. The general structure of Random Subspace algorithm is summarized in Fig. 3.

5.5. Voting algorithm

Voting is the simplest form of combining the base learning algorithms. There are several ways to combine the outputs of base classification algorithms. These fusion methods include majority voting, weighted majority voting, Naïve Bayes combination rule, behavioural knowledge space method and probabilistic approximation (Kuncheva, 2014). In the simple majority voting, the binary outputs of the k base classification algorithms are combined such that the class with the highest number of votes is determined as the output of the ensemble (Kittler, Hatef, Duin, & Matas, 1998).

6. Experimental results on ACM document collection

6.1. ACM document collection

To make a comprehensive experimental evaluation about the performance of statistical keyword extraction methods on the document collections in scientific text classification (categorization), eight collections of the ACM Digital Library are used. In the empirical analysis, the statistical keyword extraction framework presented in Rossi et al. (2014) is adopted. All of the eight datasets have documents in five classes. In Table 1, the basic descriptive information (the number of classes, the names of classes, and the number of documents per class) for ACM document collection is

Training Phase

1. Initialize the parameters
   - Set the weights $w^1 = [w_1, \ldots, w_N], w_j^1 \in [0,1], \sum_{j=1}^{N} w_j^1 = 1$.
   - Initialize the ensemble $D = \emptyset$.
   - Pick L, the number of classifiers for training
2. For $k = 1, \ldots, L$
   - Take a sample S_k from Z using distribution w^k
   - Build a classifier D_k using S_k as the training set.
   - Calculate the weighted ensemble error at kth step by the following formula: $\varepsilon_k = \sum_{j=1}^{N} w_j^k l_k^j$ ($l_k^j = 1$ if D_k misclassifies z_j and $l_k^j = 0$ otherwise)
   - If $\varepsilon_k = 0$ or $\varepsilon_k \geq 0.5$, ignore D_k, reinitialize the weights w_j^k to 1/N and continue. Else, calculate $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$, where $\varepsilon_k \in (0,0.5)$,
   - Update individual weights: $w_j^{k+1} = \frac{w_j^k \beta_k^{(1-l_k^j)}}{\sum_{i=1}^{N} w_i^k \beta_k^{(1-l_k^i)}}$ (j=1,…,N)
3. Return D and $\beta_1, \ldots, \beta_L$

Classification Phase

4. Calculate the support for class ω_t by: $\mu_t(x) = \sum_{D_k(x)=w_t} ln\left(\frac{1}{\beta_k}\right)$
5. The class with maximum support is chosen as the label for x.

Fig. 1. The general structure for AdaBoost algorithm (Kuncheva, 2014).

Training Phase

1. Initialize the parameters
   - Initialize the ensemble $D = \emptyset$.
   - Pick L, the number of classifiers for training
2. For $k = 1, \ldots, L$
   - Take a bootstrap sample S_k from Z.
   - Build a classifier D_k using S_k as the training set.
   - Add the classifier to the current ensemble $D = D \cup D_k$
3. Return D.

Classification Phase

4. Run D_1,…,D_L on the input x.
5. The class with maximum number of votes is chosen as the label for x.

Fig. 2. The general structure for Bagging algorithm (Kuncheva, 2014).

Input: *S:* training set, *B:* number of subspaces, *p*:* subspace size.
Output: *E:* Ensemble Method
$E \leftarrow 0$
For $i = 1, \ldots, B$
- Select random subspace (S^i) based on training set (S) and subspace size (p^*).
- Train C^i classifier on selected random subspace (S^i).
- Include C^i classifier to the E ensemble method.

Return E

Fig. 3. The general structure for Random Subspace algorithm (Panov & Dzeroski, 2007).

presented. Total number of documents for ACM-1, ACM-2, ACM-3, ACM-4, ACM-5, ACM-6, ACM-7 and ACM-8 collections are 401, 411, 424, 394, 471, 439, 471 and 495, respectively. The five statistical keyword extraction methods (frequent measure based keyword extraction, term frequency-inverse sentence frequency based keyword extraction, co-occurrence statistical information based keyword extraction, eccentricity-based keyword extraction and TextRank algorithm) are applied to the datasets to extract keywords from the text collections. For the datasets utilized in the empirical analysis, five different statistical keyword extraction methods with different number of keywords (800 different configurations for ACM document collection) are obtained by the Statistical Keyword Extraction Tool (SKET). In order to analyse the effect of different number of keywords on text classification, different number of keywords ranging from 5 to 100 is extracted. For all of the eight datasets utilized in the experimental analysis, five different statistical keyword extraction methods with twenty different numbers of keywords (5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95 and 100) per document are considered. Hence, 100 different configurations are generated for each of the eight datasets. In this way, 800 different configurations are obtained for the ACM document collection. In Table 2, the average number of distinct terms extracted by the statistical keyword extraction methods with different number of keywords per document and the total number of distinct terms with bag of words representation (without using any keyword-based representation) for ACM document collection are presented. The total number of terms is reduced by about 98% by extracting 5 keywords per

Table 1
The basic descriptive information for ACM document collection (Rossi et al., 2014).

Col.	Class	#Docs.	Col.	Class	#Docs.
ACM-1	3D technologies	91	ACM-5	Tangible and embedded interaction	81
	Visualization	72		Management of data	96
	Wireless mobile multimedia	82		User interface software and technology	104
	Solid and physical modelling	74		Information technology education	87
	Software engineering	82		Theory of computing	103
ACM-2	Rationality and knowledge	86	ACM-6	Computational geometry	89
	Simulation	84		Access control models and technologies	90
	Software reusability	72		Computational molecular biology	71
	Virtual reality	83		Parallel programming	96
	Web intelligence	86		Integrated circuits and system design	93
ACM-3	Computer architecture education	78	ACM-7	Database systems	104
	Networking and communications systems	75		Declarative programming	101
	Privacy in the electronic society	98		Parallel and distributed simulation	98
	Software and performance	81		Mobile systems, applications and services	95
	Web information and data management	92		Network and system support for games	73
ACM-4	Embedded networked sensor systems	50	ACM-8	Mobile ad hoc networking and computing	90
	Information retrieval	71		Knowledge discovery and data mining	105
	Parallel algorithms and architectures	98		Embedded systems	102
	Volume visualization	104		Hypertext and hypermedia	93
	Web accessibility	71		Microarchitecture	105

Table 2
Average number of terms extracted by statistical keyword extraction methods per document.

Number of keywords	ACM-1	ACM-2	ACM-3	ACM-4	ACM-5	ACM-6	ACM-7	ACM-8
Bag of words (all)	38,827	46,470	37,287	51,557	38,574	51,743	52,540	47,275
5	841	869	807	761	947	860	876	924
10	1476	1481	1398	1304	1555	1487	1519	1581
15	1969	1956	1861	1729	2016	1980	2022	2079
20	2374	2357	2248	2088	2412	2385	2438	2481
25	2713	2705	2586	2404	2741	2742	2786	2824
30	3020	3023	2863	2695	3032	3055	3093	3131
35	3130	3206	2954	2940	3452	3428	3363	3405
40	3374	3484	3179	3179	3711	3703	3627	3671
45	3616	3753	3387	3410	3942	3960	3882	3914
50	3836	4018	3589	3632	4173	4193	4113	4143
55	4046	4269	3783	3844	4386	4413	4337	4369
60	4260	4525	3977	4060	4585	4630	4566	4597
65	4450	4720	4169	4275	4784	4822	4760	4797
70	4633	4880	4363	4461	4947	5007	4919	4975
75	4821	5068	4543	4633	5123	5186	5118	5162
80	4999	5235	4720	4788	5285	5358	5276	5302
85	5146	5413	4877	4938	5458	5521	5435	5469
90	5295	5578	5043	5109	5601	5680	5609	5617
95	5467	5737	5190	5249	5755	5848	5766	5778
100	5626	5911	5342	5414	5927	6006	5920	5925

documents and the total number of terms is reduced by about 87% by extracting 100 keywords per document. In SKET toolkit, several pre-processing steps can be done. These steps include word-stemming, stop-words removal and frequency cut. In the toolkit, Porter's stemmer is used to reduce the words to their word stem (Porter, 1989). The stop-words removal procedure of the toolkit was adopted from Pretextool. The frequency cut enables user to eliminate a number of the less frequent terms. In order to represent text documents, sentences or a sliding window of words can be regarded as text segments (Matsuo & Ishizuka, 2003; Mihalcea & Tarau, 2004; Palshikar, 2007). As mentioned in advance, we regarded sentences as text segments. In this representation, a set of words that are separated by stop marks are considered as the sentences. Besides, the terms that are seen in only one text segment are eliminated. The datasets are also available in a pre-processed format.

6.2. Evaluation measures

In order to evaluate the predictive performance of statistical keyword extraction methods, classification algorithms and ensemble methods, classification accuracy, *F*-measure and area under curve (AUC) are utilized as the evaluation metrics. Classification accuracy (ACC) is one of the most widely employed metrics in examining the classifiers. It is the proportion of true positives and true negatives over the total number of instances as given by Eq. (7):

$$ACC = \frac{TN + TP}{TP + FP + FN + TN} \tag{7}$$

where *TN, TP, FP* and *FN* represents the number of true negatives, the number of true positives, the number of false positives and the number of false negatives, respectively. Precision (*PRE*) is the proportion of the true positives against the sum of true positives and false positives as given by Eq. (8):

$$PRE = \frac{TP}{TP + FP} \tag{8}$$

Recall (*REC*) is the proportion of the true positives against the true positives and false negatives as given by Eq. (9):

$$REC = \frac{TP}{TP + FN} \tag{9}$$

F-measure takes values between 0 and 1. It is the harmonic mean of precision and recall as determined by Eq. (10):

$$F-measure = \frac{2 \times PRE \times REC}{PRE + REC} \quad (10)$$

In the empirical analysis, macro-averaged *F*-measure values are presented.

The area under curve (AUC) is another common metric for evaluating the classifiers. It is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. It takes on values from 0 to 1. The higher values of AUC indicate better performance of the classification algorithms.

6.3. Experimental procedure

In the experimental analysis, 10-fold cross validation method is used. In this method, the original data set is randomly divided into 10 mutually exclusive folds. Training and testing process is repeated 10 times and each part is tested and trained 10 times and the average results for 10-fold are reported. For classifier design, learning algorithms from different classification approaches are selected. Hence, Naïve Bayes algorithm, support vector machines, logistic regression and Random Forest algorithm are used as the baseline classifiers. In ensemble construction, five popular ensemble methods (AdaBoost, Bagging, Dagging, Random Subspace and Voting) are examined. The experimental evaluations are carried on a PC with a Intel Core i7 CPU 3.40 GHz with 8.00GB RAM. The experiments are performed with the machine learning toolkit WEKA (Waikato Environment for Knowledge Analysis) version 3.7.11 (Witten et al., 2011). It is an open-source platform that contains many machine learning algorithms implemented in JAVA. The default parameters of WEKA are assigned based on the empirically obtained values and they tend to perform well (Amancio et al., 2014). Hence, default parameters of WEKA are used for the classifier design and ensemble construction. In WEKA toolkit, the default number of iterations to get statistically meaningful results is 10. For a 10-fold cross-validation scheme, this means 100 calls of one classifier with training data and tested against test data (Witten et al., 2011). Hence, we have adopted this scheme to minimize the side effect of the variability of the training set and all the experimental results reported in the study are repeated 100 times. The experimental results in Section 6.4 presents the average results for this scheme. First, text documents are pre-processed. In preprocessing, word-stemming, stop-words removal and frequency cut can be applied to the text documents. The details and applied methods for pre-processing are presented in Section 6.1. After preprocessing, text documents are structured into a segment-term matrix. In this study, sentences are considered as text segments. Hence, we build a sentence-term matrix from the text document, where the occurrence of a particular keyword is represented by one and the lack of a particular keyword is represented by zero. By this way, a representation of text documents in a matrix is obtained. Based on this representation, a score value is calculated for each term by using the five statistical keyword extraction methods utilized in the empirical analysis. By this way, score values indicating the merit of a particular term are obtained. These terms are sorted and the most informative/useful terms are selected as the keywords. In the experimental analysis, twenty different number of keywords (5–100) are taken into consideration.

6.4. Results and discussion on ACM collection

This section presents the classification accuracy, *F*-measure and the area under curve (AUC) measure values of classification algorithms and ensemble methods by the keywords obtained by different statistical keyword extraction methods (the most frequent measure based keyword extraction, term frequency-inverse sentence frequency based keyword extraction, co-occurrence statistical information based keyword extraction, eccentricity-based keyword extraction and TextRank algorithm) on 800 different configurations of ACM document collection. In the tables, the best (the highest) results obtained by a particular keyword extraction method are indicated as only boldface, the second best results obtained by a particular keyword extraction method are indicated as both boldface and italics. The highest predictive performance among all the compared results is indicated as both boldface and underline. The first concern of the study is to examine the predictive performance of statistical keyword extraction methods in scientific text classification. As it can be observed from Table 3, the highest (best) classification accuracies are obtained by the most frequent based keyword extraction method. The second best classification accuracies are generally obtained by TextRank algorithm. The third best classification accuracies are obtained by eccentricity-based keyword extraction and the fourth best classification accuracies are obtained by term frequency-inverse sentence frequency method. The worst results in terms of predictive performance are obtained by co-occurrence statistical information based keyword extraction method.

Table 3
Classification accuracies obtained by different algorithms and keyword extraction methods.

Algorithm	CSI	EB	MF	TF-ISF	TR
NB	***78.00***	90.29	92.34	88.79	91.24
SVM	71.60	87.79	91.26	86.21	89.85
LR	71.90	89.01	91.74	87.13	90.90
RF	76.29	***90.88***	92.93	***89.83***	***92.66***
AdaBoost + NB	76.38	87.49	89.65	86.27	88.46
Bagging + NB	77.29	90.09	92.18	88.57	91.20
Dagging + NB	72.82	87.32	90.29	86.25	88.98
Random Subspace + NB	77.04	89.48	91.82	88.00	90.52
Adaboost + SVM	69.24	86.68	90.29	84.90	88.91
Bagging + SVM	71.16	87.79	91.09	86.04	89.80
Dagging + SVM	66.12	85.28	88.26	82.86	87.01
Random Subspace + SVM	74.06	88.52	91.38	87.17	90.56
Adaboost + LR	71.08	88.73	91.77	87.06	90.74
Bagging + LR	72.27	89.41	92.34	87.72	91.40
Dagging + LR	58.77	83.95	88.65	80.75	86.61
Random Subspace + LR	74.36	89.71	92.30	88.41	91.55
Adaboost + RF	67.11	83.28	86.50	79.97	85.38
Bagging + RF	**79.64**	**92.23**	**93.80**	**90.97**	**93.70**
Dagging + RF	59.30	84.61	88.48	78.99	87.82
Random Subspace + RF	75.63	90.54	92.73	89.35	92.35
Majority Voting	76.33	90.65	***93.17***	89.25	92.32

CSI: co-occurrence statistical information, EB: eccentricity-based keyword extraction, MF: most frequent, TF-ISF: term frequency-inverse sentence frequency, and TR: TextRank algorithm.

The second concern is the effectiveness of the classification algorithms and ensemble methods with the different keyword extraction methods. The empirical analysis aims to determine whether the ensemble methods can enhance the predictive performance of text classifiers when keywords are utilized as the features. Hence, we have analysed the results obtained by the classification algorithms and the ensemble methods. Taking these research concerns into account, the results indicate that Bagging ensemble of Random Forest algorithm yields the best (highest) classification accuracies in each of the keyword extraction configurations. For three keyword extraction methods (eccentricity-based keyword extraction, term frequency-inverse sentence frequency based keyword extraction and TextRank algorithm), the second highest predictive performance is obtained by Random Forest algorithm. For co-occurrence statistical information based keyword extraction, the second highest classification accuracy is obtained by Naïve Bayes algorithm. Among the all configurations compared

Table 4
Classification accuracy comparison of keyword extraction methods' performance.

Number of keywords	CSI	EB	MF	TS-ISF	TR
5	40.05	55.2	76.18	62.73	69.15
10	48.41	71.38	83.62	72.11	80.45
15	53.97	79.32	86.5	75.94	84.48
20	58.08	83.48	88.07	80.07	86.83
25	62.01	85.87	88.94	82.88	87.86
30	65.23	87.41	89.58	84.19	88.77
35	68.84	88.88	90.67	86.17	90.32
40	71.5	89.56	90.94	87.19	90.7
45	74.06	89.97	91.04	87.92	91.03
50	76.09	90.47	91.32	88.34	91.27
55	77.74	90.75	91.57	89.17	91.5
60	79.14	90.96	91.82	89.5	91.52
65	80.34	92.16	93.02	90.7	92.72
70	82.64	94.46	94.32	92	94.02
75	84.34	96.16	96.02	93.7	95.72
80	85.14	***96.91***	***97.02***	***94.2***	***96.22***
85	**86.04**	**97.06**	**97.77**	**94.7**	**96.72**
90	***85.29***	96.52	96.27	93.95	95.97
95	83.19	95.01	94.17	91.85	93.87
100	82.09	93.91	93.07	90.75	92.77

Table 5
F-measure values obtained by different algorithms and keyword extraction methods.

Algorithm	CSI	EB	MF	TF-ISF	TR
NB	**0.74**	***0.87***	***0.90***	***0.86***	0.88
SVM	0.68	0.85	0.89	0.84	0.88
LR	0.68	0.85	0.89	0.83	0.88
RF	0.51	0.74	0.77	0.68	0.77
AdaBoost + NB	***0.73***	0.85	0.87	0.83	0.86
Bagging + NB	**0.74**	***0.87***	***0.90***	0.85	0.88
Dagging + NB	0.68	0.84	0.88	0.84	0.86
Random Subspace + NB	***0.73***	***0.87***	0.89	0.85	0.88
Adaboost + SVM	0.66	0.84	0.88	0.82	0.87
Bagging + SVM	0.68	0.85	0.89	0.83	0.88
Dagging + SVM	0.63	0.83	0.87	0.81	0.85
Random Subspace + SVM	0.71	0.86	0.89	0.85	***0.89***
Adaboost + LR	0.67	0.85	0.89	0.83	0.87
Bagging + LR	0.68	0.86	***0.90***	0.84	0.88
Dagging + LR	0.64	0.84	0.87	0.82	0.85
Random Subspace + LR	0.71	0.86	0.89	0.85	0.88
Adaboost + RF	0.62	0.80	0.83	0.77	0.82
Bagging + RF	0.68	0.88	**0.91**	**0.87**	**0.90**
Dagging + RF	0.35	0.67	0.73	0.57	0.73
Random Subspace + RF	0.66	*0.87*	*0.90*	***0.86***	***0.89***
Majority Voting	0.72	*0.87*	**0.91**	***0.86***	***0.89***

CSI: co-occurrence statistical information, EB: eccentricity-based keyword extraction, MF: most frequent, TF-ISF: term frequency-inverse sentence frequency, and TR: TextRank algorithm.

in the study, the highest (best) predictive performance (93.80%) is obtained by the combination of the most frequent based keyword extraction method with Bagging ensemble of Random Forest algorithm. The other concern of the study is the optimal number of keyword to be used as the features in classification model construction. Table 4 presents the average classification performance of different number of keywords with different keyword extraction methods. As it can be observed from Table 4, the highest (best) predictive performance in each keyword extraction method is obtained for 85 keywords and the second highest predictive performance in each method is generally obtained for 80 keywords.

We have also utilized *F*-measure and area under curve (AUC) metrics as the evaluation metrics. The results obtained by *F*-measure and AUC metric are presented in Tables 5 and 6, respectively. Among all results presented in Table 5, the most frequent measure based keyword extraction method yields the highest *F*-measure values. The second highest *F*-measure values are obtained

Table 6
AUC values obtained by different algorithms and keyword extraction methods.

Algorithm	CSI	EB	MF	TF-ISF	TR
NB	***0.91***	0.96	0.97	0.95	0.96
SVM	0.80	0.91	0.93	0.90	0.92
LR	0.80	0.91	0.93	0.90	0.92
RF	0.79	0.92	0.94	0.90	0.94
AdaBoost + NB	0.90	0.96	0.97	0.95	0.96
Bagging + NB	**0.92**	***0.97***	***0.98***	***0.97***	***0.98***
Dagging + NB	0.90	***0.97***	***0.98***	***0.97***	***0.98***
Random Subspace + NB	**0.92**	***0.97***	***0.98***	***0.97***	***0.98***
Adaboost + SVM	0.88	0.96	***0.98***	0.96	0.97
Bagging + SVM	0.87	0.96	0.97	0.95	0.97
Dagging + SVM	0.87	*0.97*	*0.98*	0.96	***0.98***
Random Subspace + SVM	0.90	*0.97*	*0.98*	*0.97*	***0.98***
Adaboost + LR	0.88	0.96	*0.98*	0.96	0.97
Bagging + LR	0.87	0.96	0.97	0.95	0.97
Dagging + LR	0.88	***0.97***	*0.98*	0.96	*0.98*
Random Subspace + LR	0.90	***0.97***	*0.98*	*0.97*	*0.98*
Adaboost + RF	0.87	0.95	0.97	0.94	0.96
Bagging + RF	***0.91***	**0.98**	**0.99**	0.98	**0.99**
Dagging + RF	0.81	0.95	0.96	0.92	0.96
Random Subspace + RF	**0.92**	**0.98**	**0.99**	0.98	**0.99**
Majority Voting	0.82	0.92	0.94	0.91	0.94

CSI: co-occurrence statistical information, EB: eccentricity-based keyword extraction, MF: most frequent, TF-ISF: term frequency-inverse sentence frequency, and TR: TextRank algorithm.

by TextRank algorithm and the third highest *F*-measure values are obtained by eccentricity-based keyword extraction. The highest average *F*-measure values are generally obtained by Bagging ensemble of Random Forest algorithm. *F*-measure values obtained by Random Subspace ensemble of Random Forest algorithm and majority voting are also relatively high. Regarding the *F*-measure values obtained by different algorithms and keyword extraction methods, the best (the highest) *F*-measure value (0.91) is obtained by the most frequent measure based keyword extraction (MF). Bagging ensemble of Random Forest and majority voting classifiers achieve this *F*-measure value.

Regarding the area under curve (AUC) values presented in Table 6, the best (the highest) AUC values are obtained by the most frequent measure based keyword extraction (MF) and the second best AUC values are obtained by TextRank algorithm. Regarding the performance of classifiers and ensemble methods in terms of AUC values, the highest values are generally obtained by Random Subspace ensemble of Random Forest and Bagging ensemble of Random Forest, respectively. Besides, the highest AUC value (0.99) is obtained by Bagging ensemble of Random Forest and Random Subspace ensemble of Random Forest.

To further evaluate the results obtained in the empirical analysis, we applied two-way ANOVA test in Minitab statistical analysis software. The results for ANOVA test of overall results obtained by keyword extraction methods and classifiers are summarized in Table 7, where DF, SS, MS, *F* and *p* denote degrees of freedom, adjusted sum of squares, adjusted mean square, *F*-statistics and probability value, respectively. As it can be observed from the results presented in Table 7, there are statistically meaningful differences between the results of compared keyword extraction methods and the results of compared classifiers at 99% confidence level. Besides, the *p*-Values ($p < 0.001$) indicate that the keyword extraction methods and classifiers have statistically significant effect on the evaluation metric values. It can also be seen from Table 7 that the interaction between keyword extraction methods and classifiers is also important at least ($p < 0.05$) significance level.

To summarize the main results from the empirical analysis, we have presented main effects plots for accuracy, *F*-measure and AUC values in Figs. 4, 5 and 6, respectively. The main effect plots

Table 7
Two-way ANOVA test results.

Source (accuracy values)	DF	SS	MS	*F*	*p*
Keyword extraction methods	4	603,341	150,835	2730.02	0.000
Classifiers	20	4903	245	4.44	0.000
Keyword extraction methods∗classifiers	80	6311	79	1.43	0.009
Error	1995	110,225	55		
Total	2099	724,779			
Source (*F*-measure values)					
Keyword extraction methods	4	12.4085	3.10213	429.23	0.000
Classifiers	20	6.6671	0.33335	46.13	0.000
Keyword extraction methods∗classifiers	80	0.7683	0.00960	1.33	0.029
Error	1995	14.4181	0.00723		
Total	2099	34.2620			
Source (AUC values)					
Keyword extraction methods	4	2.6402	0.660050	359.25	0.000
Classifiers	20	1.2709	0.063545	34.59	0.000
Keyword extraction methods∗classifiers	80	0.2281	0.002851	1.55	0.002
Error	1995	3.6654	0.001837		
Total	2099	7.8046			

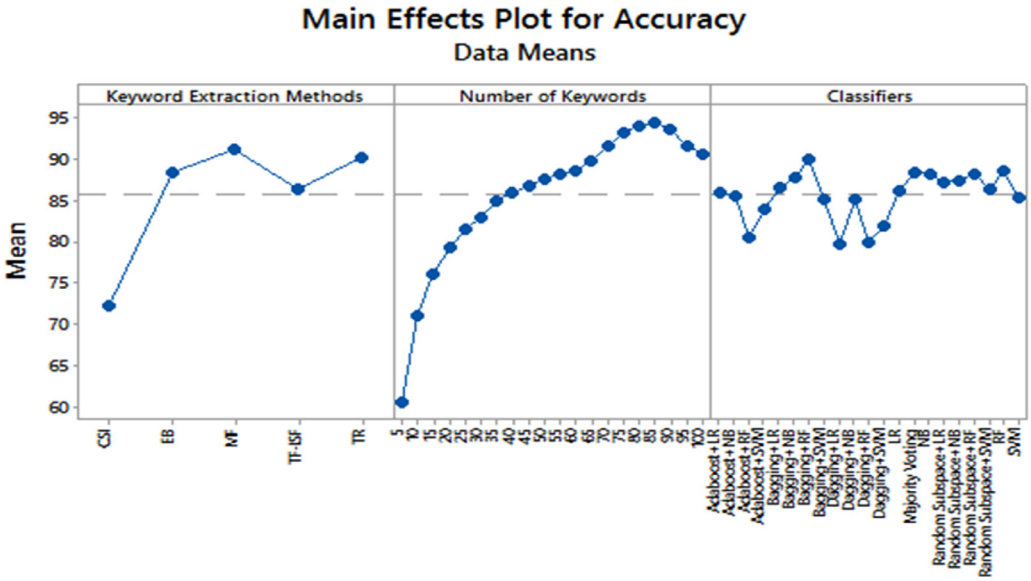

Fig. 4. Main effects plot for accuracy results.

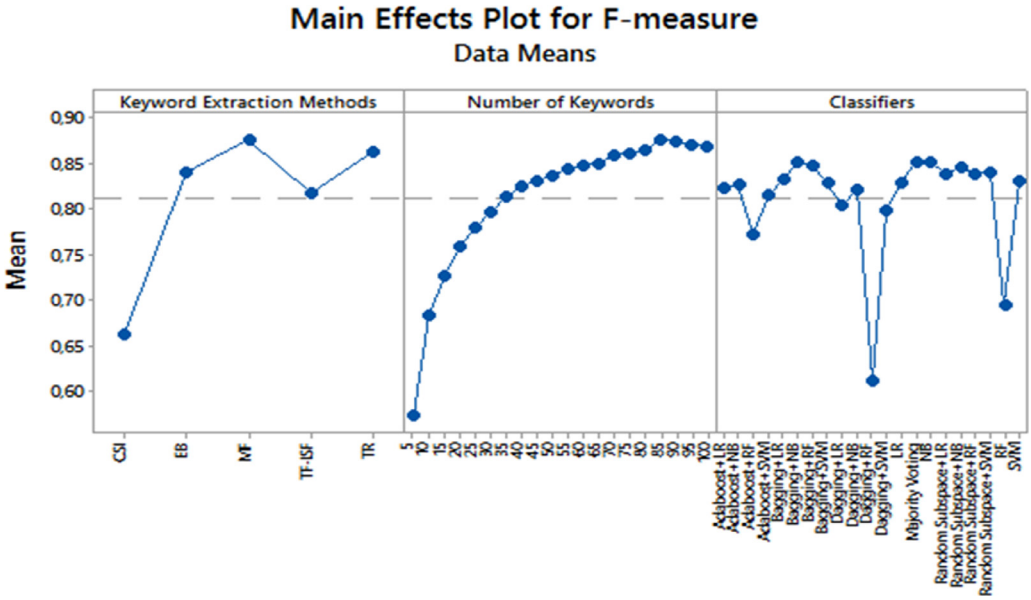

Fig. 5. Main effects plot for *F*-measure results.

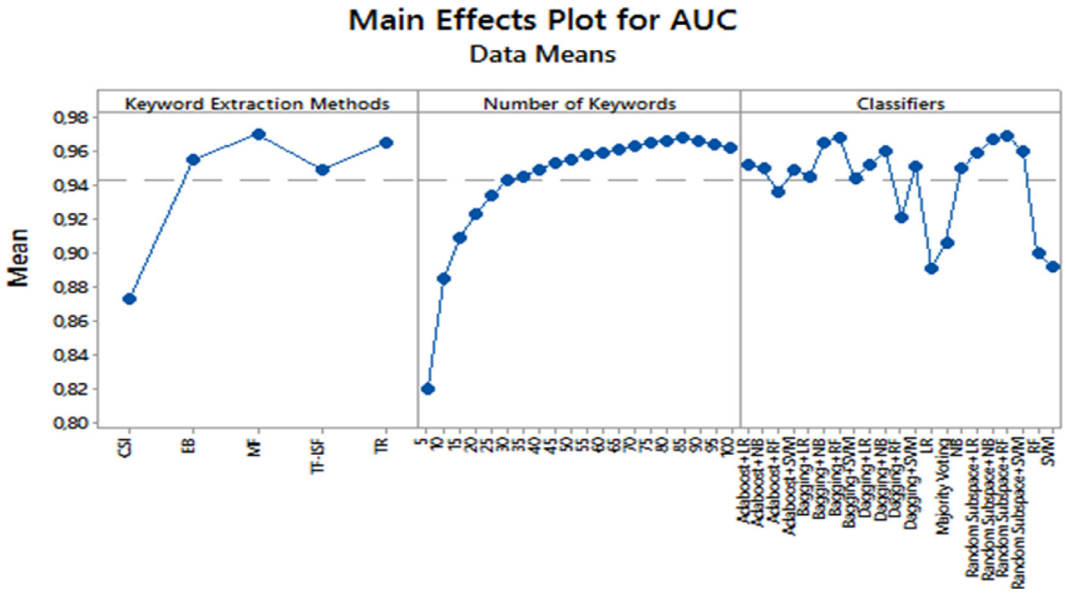

Fig. 6. Main effects plot for AUC results.

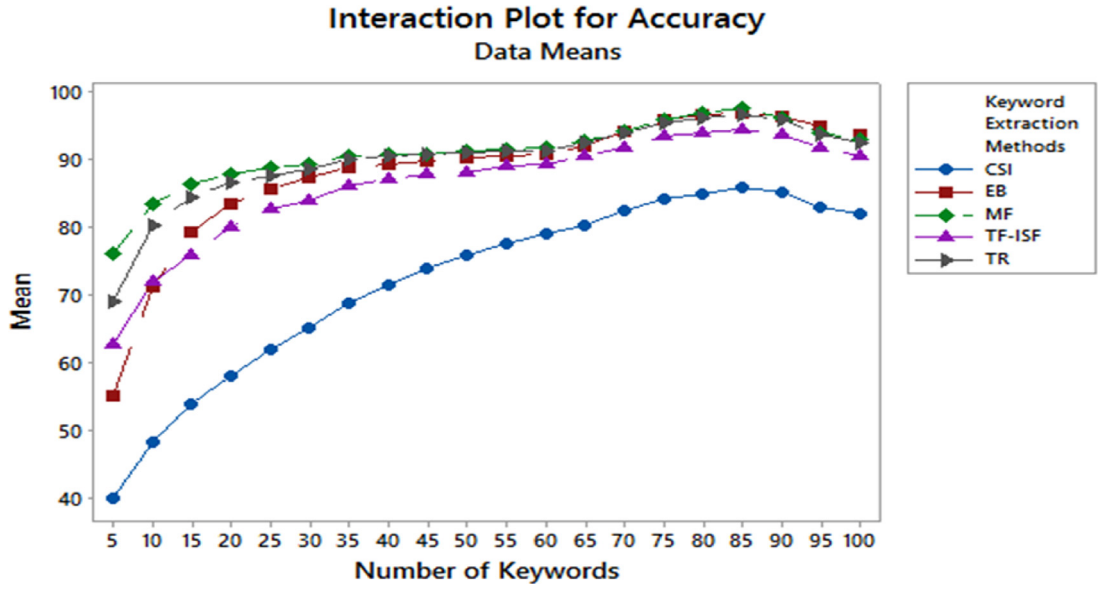

Fig. 7. Interaction plot for accuracy (keyword extraction methods–number of keywords interaction).

summarize the change of metrics' value based on the particular factor. Keyword extraction methods, the number of keywords and classifiers are the three main effects of the empirical analysis. Hence, keyword extraction based comparison presents the change of particular metric's value (classification accuracy, *F*-measure or AUC value) for different keyword extraction methods (CSI, EB, MF, TF-ISF and TR). The number of keywords based comparison presents the change of particular metric's value based on the number of keywords. Finally, classifier based comparison presents the change of particular metric's value based on the classifiers utilized in the empirical analysis. As it can be observed from Figs. 4–6, the most frequent measure based keyword extraction achieves the highest performance in terms of all compared metrics. For the keyword extraction methods, the same pattern is achieved for three compared metrics. The second highest performance is achieved by TextRank algorithm, the third highest performance is achieved by the eccentricity-based keyword extraction method, and the fourth highest performance is obtained by TF-ISF method. The worst results for each metric is obtained by co-occurrence statistical information based keyword extraction method. Besides, the figures about the number of keyword based comparisons clearly depict that the predictive performance generally enhance as the number of keywords kept in the dataset increase. The highest predictive performance in terms of all compared metrics is obtained by the number of keywords value of 85. However, there is a subtle trend of decrease for the number of keywords after the value of 85. Regarding the performance of classifiers, the highest accuracy results are obtained by Bagging ensemble of Random Forest, the highest *F*-measure and AUC values are obtained by the majority voting and Random Subspace ensemble of Random Forest, respectively. In addition, we have presented the interactions between keyword extraction methods and the number of keywords and the interactions between keyword extraction methods and classifiers in Figs. 7 and 8, respectively. As it can be observed from Fig. 7, the predictive performance enhances as the number of keywords increases up to 85 keywords. Besides, the performance

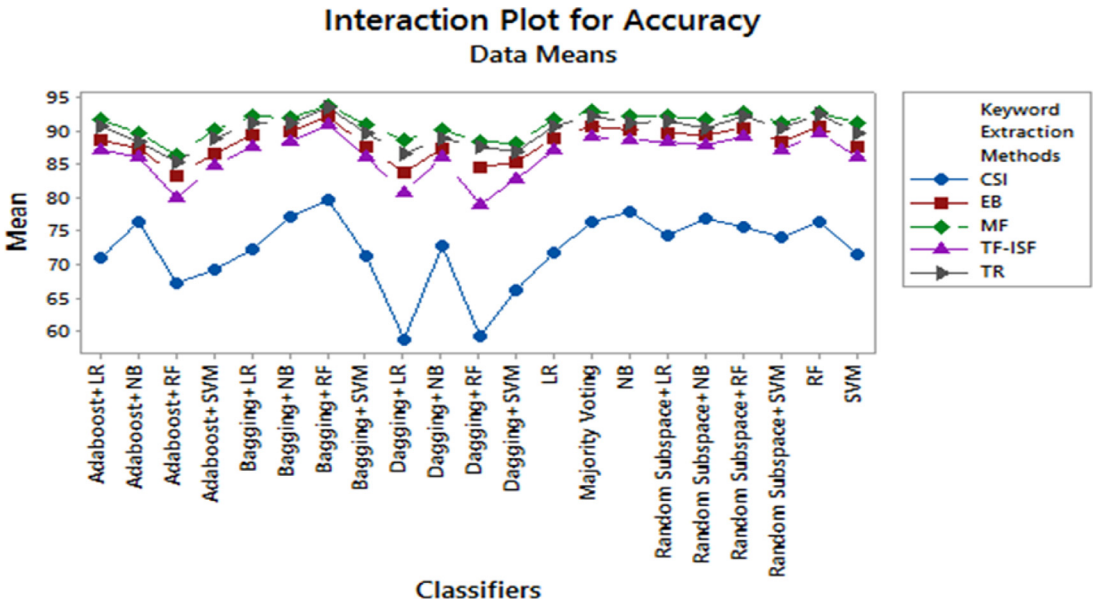

Fig. 8. Interaction plot for accuracy (keyword extraction methods–classifiers interaction).

of co-occurrence statistical information based keyword extraction method is very low compared to the other methods. For lower number of keywords (up to 10 keywords), the performance of the eccentricity-based keyword extraction method is worse than the performance of term frequency-inverse sentence frequency based keyword extraction method. However, for higher number of keywords, the eccentricity-based keyword extraction method performs better in terms of accuracy. Regarding the performance of classifiers based on the keyword extraction methods, similar performance patterns are observed on the classifiers for the most frequent measure based keyword extraction, term frequency-inverse sentence frequency based keyword extraction, eccentricity-based keyword extraction and TextRank algorithm.

Several managerial insights from the results of experimental analysis presented in Tables 3–7 can be summarized as follows:

1. For the classification performance of statistical keyword extraction methods on 21 different classifiers and classifier ensembles, most-frequent based keyword extraction method obtained the highest predictive performance in all compared cases. The second successful keyword extraction method is TextRank algorithm. Success ranking of the keyword extraction methods in terms of classification accuracy is most-frequent based keyword extraction method, TextRank algorithm, eccentricity-based keyword extraction method, term frequency-inverse sentence frequency based keyword extraction and co-occurrence statistical information in respective order.
2. Ensemble learning algorithms can generate more successful results from a single learning algorithm. Bagging ensemble of Random Forest, in general, obtains more accurate results than the base learning algorithms. For most-frequent based keyword extraction method, majority voting scheme also yields higher predictive performance than the base learning algorithms. Random Subspace ensemble works well with support vector machines and logistic regression classifier. For all of the compared cases, Random Subspace ensemble of support vector machines outperforms support vector machines and Random Subspace ensemble of logistic regression outperforms logistic regression. Bagging ensemble works well with logistic regression and Random Forest classifiers.
3. Regarding the predictive performance of ensemble learning algorithms, Random Subspace and Bagging methods works well with the base learning algorithms. In contrast, the worst predictive performance is obtained by Dagging ensemble.
4. For the predictive performance of base learning algorithms, Random Forest algorithm, in general, performs better than the other base learning algorithms. For occurrence statistical information based keyword extraction method, Naïve Bayes algorithm, however, outperforms all the other base learning algorithms.
5. For different number of keywords, there are similarities in the performance of statistical keyword extraction methods. As the number of keywords utilized in the empirical analysis increase up to 85 keywords, the classification accuracies obtained by keyword-based text documents increase. In contrast, there is a subtle trend of decrease for the number of keywords of 90, 95 and 100.
6. The base learning algorithm-ensemble learning method pair which obtains the best results in terms of classification accuracy, *F*-measure and area under curve metrics is Bagging ensemble of Random Forest algorithm.
7. For lower number of keywords (10 keywords or lower), the term frequency-inverse sentence frequency based keyword extraction method outperforms the eccentricity-based keyword extraction method.
8. The keyword extraction methods, the number of keywords and classification schemes (the base learning algorithms and/or ensemble learning methods) have significant impact on classification results.

7. Experimental results on Reuters-21578 document collection

To better understand the performance of the ensemble learning methods in keyword-based text classification, we divide the experimental analysis into two sections. In the first section (Section 6), the predictive performance of five statistical keyword extraction methods, classification algorithms and ensemble learning methods are extensively analysed on ACM document collection. Text classification is characterized by high dimensionality of the feature space. In the second section (Section 7), the predictive performance of

Table 8
Basic descriptive information about Reuters-21578 dataset (Uysal, 2016).

Class label	Training samples	Testing samples
Earn	2877	1087
Acq	1650	719
Money-fx	538	179
Grain	433	149
Crude	389	189
Trade	369	117
Interest	347	131
Ship	197	89
Wheat	212	71
Corn	181	56

ensemble classification schemes on a larger text document collection is evaluated. The ensemble classification schemes evaluated in Section 7 are selected among classification schemes explained in Section 6 based on their high predictive performance. The main reason for splitting our experiments into two sections is that the first section empirically evaluates the predictive performance of keyword extraction methods, the number of keywords, different classifiers and classification schemes and the second section empirically evaluates the performance of best techniques on a larger collection to identify the suitability of the techniques on a larger collection. To evaluate the predictive performance of ensemble learning algorithms on a larger collection, the evaluation measures mentioned in Section 6.2 are utilized. Regarding the experimental procedure, the principles outlined in Section 6.3 are still valid for the experimental analysis presented in Section 7.

7.1. Reuters-21578 document collection

To evaluate the predictive performance of ensemble classification schemes on a larger document collection, Reuters-21578 document collection is used. The Reuters-21578 dataset utilized in the experimental analysis consists of 10 classes of Reuters-21578 ModApte Split (Asuncion & Newman, 2007; Uysal, 2016). The basic descriptive information regarding the number of training and testing samples of Reuters-21578 dataset is summarized in Table 8.

Similar to the empirical procedure listed in Section 6, five different statistical keyword extraction methods with twenty different numbers of keywords (5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95 and 100) per document are considered for Reuters-21578 document collection. In this way, 100 different configurations are generated for the Reuters-21578 document collection and the experimental results presented in Section 7.2 list the average results for these configurations. The average number of distinct terms extracted by the statistical keyword extraction methods are 3420, 5914, 7850, 9462, 10,836, 12,086, 12,672, 13,716, 14,738, 15,708, 16,630, 17,570, 18,340, 19,026, 19,778, 20,468, 21,118, 21,746, 22,408 and 23,074 for 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95 and 100 keywords per document, respectively.

7.2. Results and discussion on Reuters-21578 document collection

This section presents the classification accuracy, *F*-measure and the area under curve (AUC) measure values of the base learning algorithms and the ensemble learning methods on Reuters-21578 document collection. In Table 9, the best (the highest) results obtained by a particular keyword extraction method are indicated as only boldface, the second best results obtained by a particular keyword extraction method are indicated as both boldface and italics. The highest predictive performance among all compared results is indicated as both boldface and underline. Based on the empirical

Table 9
The evaluation results of base learners and ensemble methods and keyword extraction methods on Reuters-21578 dataset.

Algorithm	CSI	EB	MF	TS-ISF	TR
Accuracy values					
NB	71.08	82.38	83.70	76.19	81.62
SVM	64.70	74.30	78.53	70.78	77.98
LR	66.70	76.94	76.24	72.71	78.79
RF	70.57	79.94	81.73	75.42	79.78
Bagging + RF	**73.34**	**87.37**	**91.49**	**82.39**	**88.96**
Random Subspace + RF	72.67	***85.77***	***91.42***	***82.19***	***88.63***
Majority Voting	***72.94***	83.18	86.83	76.90	82.64
***F*-measure values**					
NB	0.68	0.83	0.87	0.76	0.81
SVM	0.65	0.79	0.85	0.71	0.76
LR	0.65	0.79	0.84	0.73	0.79
RF	0.67	0.81	0.87	0.75	0.80
Bagging + RF	**0.71**	**0.87**	**0.92**	**0.82**	**0.89**
Random Subspace + RF	***0.70***	***0.85***	***0.91***	***0.81***	***0.88***
Majority Voting	***0.70***	0.83	0.87	0.75	0.82
AUC values					
NB	***0.79***	0.87	*0.90*	*0.83*	0.85
SVM	0.75	0.81	0.88	0.78	0.80
LR	0.78	0.82	0.87	0.80	0.81
RF	***0.79***	0.86	0.89	0.82	0.85
Bagging + RF	**0.95**	**0.98**	**0.99**	**0.98**	**0.99**
Random Subspace + RF	**0.95**	***0.97***	**0.99**	**0.98**	**0.99**
Majority Voting	***0.79***	0.87	***0.90***	***0.83***	***0.87***

CSI: co-occurrence statistical information, EB: eccentricity-based keyword extraction, MF: most frequent, TF-ISF: term frequency-inverse sentence frequency, and TR: TextRank algorithm.

analysis presented in Section 6.4, the ensemble learning algorithms are evaluated in terms of their predictive performance (classification accuracy). As listed in Table 3, the highest predictive performance on ACM document collection is obtained by Bagging ensemble of Random Forest algorithm. In the empirical analysis on Reuters-21578 document collection, we intend to examine the performance of ensemble learning method that performs well. Hence, the root mean square error for the classification accuracy of each ensemble learning scheme is computed by taking Bagging ensemble of Random Forest as the base model. Based on this computation, a ranking of the ensemble learners on ACM document collection is obtained. Therefore, Bagging ensemble of Random Forest, Random Subspace ensemble of Random Forest and the majority voting schemes are selected in the experimental analysis listed in Section 7.

Regarding the performance of ensemble learning methods on Reuters-21578 document collection, the highest (the best) classification accuracy (91.49%) among all the configurations listed in Table 9 is obtained by the combination of most frequent based keyword extraction method in conjunction with Bagging ensemble of Random Forest. As summarized in the managerial insights of the empirical analysis presented in Section 6.4, the same ranking list of keyword extraction methods is still valid on Reuters-21578 collection. In other words, the most-frequent based keyword extraction method is the best keyword extraction method and the co-occurrence statistical information based keyword extraction method obtains the worst predictive performance. In Table 9, the highest accuracy values are obtained by Bagging ensemble of Random Forest for all the compared keyword extraction methods. The second highest accuracy values are generally achieved by Random Subspace ensemble of Random Forest. Regarding the performance of ensemble learning methods in terms of *F*-measure values, the highest *F*-measure value is also obtained by Bagging ensemble of Random Forest with most-frequent based keyword extraction method and the highest *F*-measure values for other compared keyword extraction methods are also obtained by this method.

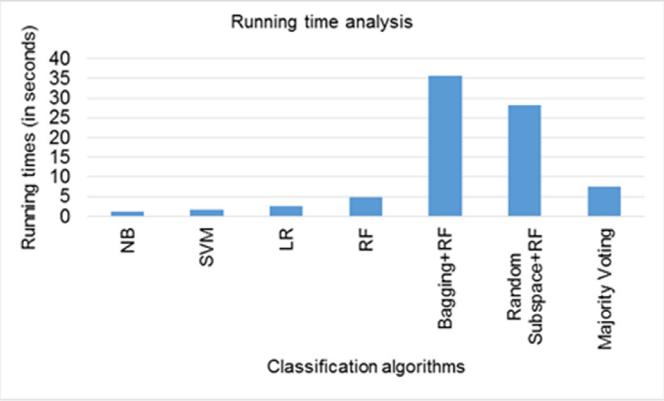

Fig. 9. Average running time analysis for the compared classification algorithms.

Regarding the performance of ensemble learning methods in terms of area under curve values, the highest values are obtained by Bagging ensemble of Random Forest and Random Subspace ensemble of Random Forest when most-frequent based keyword extraction or TextRank algorithm is utilized.

As stated in advance, high dimensional feature space is one of the challenges of text classification and the keyword-based representation of text documents can be a good alternative to deal with this problem. The comprehensive empirical analysis on ACM collection indicates that keyword-based representation can yield promising results. In addition, the evaluation of ensemble learning on a larger text document collection (Reuters-21578) supports the findings obtained on the ACM collection and indicates that this classification scheme can be utilized on larger collections, as well.

The empirical analysis on Reuters-21578 document collection indicates that the ensemble learning methods generally yield better predictive performance in terms of accuracy, *F*-measure and area under curve values. In Fig. 9, the average running times of classification algorithms and ensemble methods on Reuters-21578 document collection is presented. Regarding the computational analysis of classification algorithms and ensemble methods, there is a trade-off between the running time and the predictive performance of classification models. The running times of base learning algorithms are relatively low (1.3 s for Naïve Bayes algorithm), whereas the running times of the ensemble learning methods are relatively high (about 35 s for Bagging ensemble of Random Forest algorithm). In contrast, the predictive performances of the ensemble learning methods are higher than the predictive performances of the base learning algorithms; such as 83.70% for Naïve Bayes algorithm and 91.49% for Bagging ensemble of Random Forest algorithm. Regarding the running times of the base learning algorithms, the best performance is obtained by Naïve Bayes algorithm and the worst performance is obtained by Random Forest algorithm. Since the ensemble learning methods presented in Table 9 utilize Random Forest algorithm as the base learning algorithm, their running times are relatively high.

8. Conclusion

This paper presents an empirical analysis for five statistical keyword extraction methods (the most frequent measure based keyword extraction, the term frequency-inverse sentence frequency based keyword extraction, the co-occurrence statistical information based keyword extraction, the eccentricity-based keyword extraction and the TextRank algorithm) in conjunction with classification algorithms and ensemble learning methods.

The main contributions of this study can be summarized as follows. First, it presents an extensive empirical analysis on the predictive performance of classification algorithms and ensemble methods when keyword-based representation is utilized to represent scientific text documents. To the best of our knowledge, this is the first comprehensive analysis on this field. The representation of text documents with keywords in a compact way can be extremely useful, since text documents are characterized by the high dimensionality of feature space. Secondly, text classification finds applications in a variety of fields, including document organization and opinion mining. Hence, the development of efficient text classification schemes can be useful in the application fields of text classification. In the context of expert and intelligent systems, text classification serves as an important tool to provide business intelligence. With the advances in information technology, vast majority of information is stored in the form of text documents. This huge amount of unstructured data can only provide valuable insights to decision makers with the use of text classification tools. Hence, the development of efficient and robust text classification schemes is of great importance for business intelligence. Based on the experimental analysis, the best (highest) average predictive performance on ACM document collection is obtained by the combination of the most frequent based keyword extraction method with Bagging ensemble of Random Forest algorithm. Besides, the experimental results indicate that as the number of keywords increases the predictive performance of classifiers generally enhances. Regarding the evaluation of classifiers in terms of accuracy, *F*-measure and area under curve value, the highest results are generally obtained by Bagging and Random Subspace ensembles of Random Forest algorithm. In addition, the *p*-Values ($p < 0.05$) indicate that keyword extraction methods and classifiers have statistically significant effect on the performance metrics. In order to evaluate the performance of ensemble classifiers on a larger text document collection, results of the best techniques on Reuters-21578 Document Collection is also presented. The results on this collection also support the findings obtained on ACM document collection. Hence, the results presented in the study can be used as a valid future reference for the future works on the keyword-based representation for text classification.

As pointed out in the previous sections, the scope of the paper is confined to the analysis of statistical keyword extraction methods. There are also linguistic, machine learning based and other approaches for keyword extraction. In addition, twenty different numbers of keywords are examined in the empirical analysis. Besides, the ensemble learning methods are evaluated on ACM document collection and Reuters-21578 collection.

There are a number of aspects that should be beneficial to extend in the future. First, the analysis of the keyword extraction methods from different models and the comparison of these models can be a good research direction. By taking the keyword extraction methods and classification performances reported in this paper as a basis, it should be beneficial to propose a more robust and scalable keyword extraction based representation scheme for text classification. In addition, the performance of keyword extraction-based text representation should be evaluated on text benchmarks from several domains with different characteristics, such as sentiment analysis, medical documents and news articles.

References

Abellan, J., & Mantas, C. J. (2014). Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 41*(8), 3825–3830.

Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing, 38*, 360–372.

Aggarwal, C. C., & Zhai, C. X. (2012). A survey of text classification algorithms. In C. C. Aggarwal, & C. X. Zhai (Eds.), *Mining text data* (pp. 77–128). Berlin Heidelberg: Springer-Verlag.

Amancio, D. R., Comin, C. H., Casanova, D., Travieso, G., Bruno, O. M., Rodrigues, F. A., et al. (2014). A systematic comparison of supervised classifiers. *PLoS One, 9*(4), 1–14.

Asuncion, A., & Neman, D. J. (2007). *UCI machine learning repository*. Irvine, CA: University of California, Department of Information and Computer Science.

Beliga, S., Mestrovic, A., & Martincic-Ipsic, S. (2015). An overview of graph-based keyword extraction methods and approaches. *Journal of Information and Organizational Sciences, 39*(1), 1–20.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123–140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Catal, C., Tufekci, S., Pirmit, E., & Kocabag, G. (2015). On the use of ensemble of classifiers for accelerometer-based activity recognition. *Applied Soft Computing, 37*, 1018–1022.

De Silva, N. F. F., Hruschka, E. R., & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems, 66*, 170–179.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proceedings of the first international workshop on multiple classifier systems* (pp. 1–15).

Fersini, E., Messina, E., & Pozzi, F. A. (2014). Sentiment analysis: Bayesian ensemble learning. *Decision Support Systems, 68*, 26–38.

Fiori, A. (2014). *Innovative document summarization techniques: Revolutionizing knowledge understanding*. Pennsylvania: IGI Global.

Grineva, M., Grinev, M., & Lizorkin, D. (2009). Extracting key terms from noisy and multi-theme documents. In *Proceedings of the 18th international conference on world wide web* (pp. 661–670).

HaCohen-Kerner, Y. (2003). Automatic extraction of keyword from abstracts. *Lecture Notes in Computer Science, 2773*, 843–849.

HaCohen-Kerner, Y., Gross, Z., & Masa, A. (2005). Automatic extraction and learning of keyphrases from scientific articles. *Lecture Notes in Computer Science, 3406*, 657–669.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufmann Publishers.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(8), 832–844.

Huan, C., Tian, Y., Zhou, Z., Ling, C. X., & Huang, T. (2006). Keyphrase extraction using semantic network structure analysis. In *Proceedings of the sixth international conference on data mining* (pp. 275–284).

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 216–223).

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS Transactions on Computers, 8*(4), 966–974.

Jain, A., Raghuvanshi, A., & Shrivastava, G. (2012). Analysis of query based text classification approach. *International Journal of Advanced Research in Computer Science and Software Engineering, 8*(2), 362–366.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the tenth European conference on machine learning* (pp. 137–142).

Joachims, T. (2002). *Learning to classify text using support vector machines*. New York: Springer.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the eleventh conference on uncertainty in artificial intelligence* (pp. 338–345).

Kantardziz, M. (2011). *Data mining: Concepts, models, methods and algorithms*. New York: Wiley-IEEE Press.

Kim, M. J., & Kang, D. K. (2012). Classifier selection in ensemble using genetic algorithms for bankruptcy prediction. *Expert Systems with Applications, 39*(10), 9308–9314.

Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intellience, 20*(3), 226–239.

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: A review of classification and combination techniques. *Artificial Intelligence Review, 2006*(3), 159–190.

Krapivin, M., Autayeu, A., Marchese, M., Blanzieri, E., & Segata, N. (2010). Keyphrase extraction from scientific documents: Improving machine learning approaches with natural language processing. *Lecture Notes in Computer Science, 6102*, 102–111.

Kuncheva, L. (2014). *Combining pattern classifiers: Methods and algorithms*. New York: John Wiley & Sons Publishers.

Litvak, M., & Last, M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on multi-source multilingual information extraction and summarization* (pp. 17–24).

Liu, J., & Wang, J. (2007). Keyword extraction using language network. In *Proceedings of the international conference on natural language processing and knowledge engineering* (pp. 129–134).

Matsuo, Y., & Ishizuka, M. (2003). Keyword extraction from a single document using word co-occurrence statistical information. In *Proceedings of the sixteenth international Florida artificial intelligence research society conference* pp. 392–296.

Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools, 13*(1), 1–13.

Medelyan, O., & Witten, I. H. (2006). Thesaurus based automatic keyphrase indexing. In *Proceedings of the Sixth ACM/IEEE-CS JCD* (pp. 296–297).

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404–411).

Najafi, E., & Darooneh, A. H. (2015). The fractal patterns of words in a text: A method for automatic keyword extraction. *PLoS One, 10*(6), e0130617.

Neto, L. J., Santos, A. D., Kaestner, C. A., & Freitas, A. A. (2000). Document clustering and text summarization. In *Proceedings of the 4th international conference on practical applications of knowledge discovery and data mining* (pp. 41–55).

Nguyen, T. D., & Kan, M. Y. (2007). Keyphrase extraction in scientific publications. In *Proceedings of the 10th international conference on Asian digital libraries* (pp. 317–326).

Onan, A. (2016). Classifier and feature set ensembles for web page classification. *Journal of Information Science, 42*(2), 150–165.

Onan, A., & Korukoğlu, S. (2015). A feature selection model based on genetic rank aggregation for text sentiment classification. *Journal of Information Science*. doi:10.1177/0165551515613226.

Palshikar, G. K. (2007). Keyword extraction from a single document using centrality measures. *Lecture Notes in Computer Science, 4815*, 503–510.

Panov, P., & Dzeroski, S. (2007). Combining bagging and random subspaces to create better ensembles. *Lecture Notes in Computer Science, 4723*, 118–129.

Porter, M. F. (1989). An algorithm for suffix stripping. *Readings in Information Retrieval, 14*(3), 130–137.

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics, 3*(2), 143–157.

Ranawana, R., & Palade, V. (2006). Multi-classifier systems: Review and a roadmap for developers. *International Journal of Hybrid Intelligent Systems, 3*(1), 35–61.

Reboiro-Jato, M., Diaz, F., Glez-Penaa, D., & Fdez-Riverola, F. (2014). A novel ensemble of classifiers that use biological relevant gene sets for microarray classification. *Applied Soft Computing, 17*, 117–126.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review, 33*(1), 1–39.

Rossi, R. G., Maracini, R. M., & Rezende, S. O. (2014). Analysis of domain independent statistical keyword extraction methods for incremental clustering. *Learning and Nonlinear Models, 12*(1), 17–37.

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

Seifert, G., Ulbrich, E., & Granitzer, M. (2011). Word clouds for efficient document labelling. *Lecture Notes in Computer Science, 6926*, 292–306.

Siddiqi, S., & Sharan, A. (2015). Keyword and keyphrase extraction techniques: A literature review. *International Journal of Computer Applications, 109*(2), 18–23.

Ting, K. M., & Witten, I. H. (1997). Stacking bagged and dagged models. In *Proceedings of the 14th international conference on machine learning* (pp. 367–375).

Tsai, C. F., Lin, Y. C., Yen, D. C., & Chen, Y. M. (2011). Predicting stock returns by classifier ensemble. *Applied Soft Computing, 11*(2), 2452–2459.

Turney, P. D. (2003). Coherent keyphrase extraction via web mining. In *Proceedings of the 18th international joint conference on artificial intelligence* (pp. 434–439).

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

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Wang, G., Sun, J., Ma, J., Xue, K., & Gu, J. (2014). Sentiment classification: The contribution of ensemble learning. *Decision Support Systems, 57*, 77–93.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques*. Burlington: Morgan Kaufmann Publishers.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on digital libraries* (pp. 254–255).

Xia, R., Zong, C., & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences, 181*(6), 1138–1152.

Yang, B., Zhang, Y., & Li, X. (2011). Classifying text streams by keywords using classifier ensemble. *Data & Knowledge Engineering, 70*(9), 775–793.

Zhang, C., Wang, H., Liu, Y., Wu, D., Liao, Y., & Wang, B. (2008). Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems, 4*(3), 1169–1180.

Zhang, P., & He, Z. (2015). Using data-driven feature enrichment of text representation and ensemble technique for sentence-level polarity classification. *Journal of Information Science, 41*(4), 531–549.