Artificial Intelligence Review (2019) 52:273–292
https://doi.org/10.1007/s10462-018-09677-1

Survey on supervised machine learning techniques for automatic text classification

Ammar Ismael Kadhim[1]

Published online: 19 January 2019

Abstract
Supervised machine learning studies are gaining more significant recently because of the availability of the increasing number of the electronic documents from different resources. Text classification can be defined that the task was automatically categorized a group documents into one or more predefined classes according to their subjects. Thereby, the major objective of text classification is to enable users for extracting information from textual resource and deals with process such as retrieval, classification, and machine learning techniques together in order to classify different pattern. In text classification technique, term weighting methods design suitable weights to the specific terms to enhance the text classification performance. This paper surveys of text classification, process of different term weighing methods and comparison between different classification techniques.

Keywords Supervised machine learning · Text classification · Term weighting · Classification techniques

1 Introduction

Supervised machine learning is one of the well-studied problems in text classification and information retrieval. Classification is the task in which ideas and objects are determined, differentiated and understood. Classification defines that objects are collected into categories, usually for some specific purpose. A category explores a relationship between the words and words of knowledge as well as the words and the categories. The text classification involves short and long text. Text classification becomes a key technology to deal with and organize large numbers of documents. Text classification is an important task that it used for information management applications by automatically allocating a specified document to one or more predefined classes. Automatic text classification is treated as a supervised machine learning technique. The goal of this technique is to determine whether a given document belongs to the given category or not by looking at the words or terms of that category.

✉ Ammar Ismael Kadhim
ammarusm70@gmail.com

[1] Department of Computer Science, College of Medicine, University of Baghdad, Baghdad, Iraq

The present research of text classification objects to enhance the quality of text representation and develop high quality classifiers (Jothi and Thenmozhi 2015). Furthermore, text classification (TC) aids users' hold their fields of attention, specify them to be easy separate out texts that are not related to their attention by automatically grouping the texts according to their subjects. Thereby, these groups may then be utilized to both enhance confident tasks like getting search outcomes and avail as a means of enhancing user knowledge in searching the essential text dataset (Bindra 2012).

This paper discusses the different supervised machine learning techniques such as Naïve Bayes classifiers (NB), Support vector machine (SVM) and k-Nearest Neighbors (kNN) that are used in the present research work and analyzes the effect of each technique on text classification using machine learning algorithms.

2 Text classification

Text classification can be defined that the task was automatically categorized a group documents into one or more predefined classes according to their subjects (Joseph and Ramakrishnan 2015). Thus, the major objective of TC is to stem methods for the classification of natural language processing document (Horecki and Mazurkiewicz 2015). A specified a group of learning documents D = {d1,…, dn} with predefined classes C = {c1,…,cq} and a new document q, which is generally indicated as the query, will forecast the query's class, which goes to one or more of the classes in C (Allahyari et al. 2017).

Text classification methods are utilized in several variant tasks like looking for related documents, categorizing subject by documents from appropriate documents, establishing documents in variant subjects, etc. Therefore, the objective of the categorization is to automatically tag the suitable categorization to each document that wants to be categorized. In addition, TC can be used in several NLP applications. While the traditional methods of TC based on several human-designed structures like dictionaries, knowledge rules and distinct tree kernels (Lai et al. 2015).

2.1 Comparison between single-label and multi-label

Normally, text classification challenge is classified as two significant partitions: Specifying only one predefined class to each "unknown" natural language processing document as in Moreno and Redondo (2016) and often indicated as the single-label TC task; where closely one class $c_k \in C$ have to determine to each document $d_j \in D$. While specifying more than one predefined class to an "unknown" document as in Feng et al. (2005). While it is often indicated as the multi-label TC task, where any number $0 < nj \leq |C|$ of classes may be determined to each document $d_j \in D$. "TC is binary TC reflected a special case of single-label" (Allahyari et al. 2017), which in individual identifies neither a predefined class nor its supplement to an "unknown" document (Sebastiani 2006). In the past, several researches were studied.

Disadvantages of classical single-label TC as follows:

(i) The big size of word-based data is so huge and it keeps problems in expressions of testing.
(ii) Several different predefined classes included.
(iii) Lacking terms number and documents number for learning tasks.
(iv) A few texts that were classified to a single class and the other have multiple classes.

Thereby, the original shape for textual datasets were very difficult to achieve TC.

2.2 Automatic text categorization

The common challenge of text categorization can be further classified into multiple sub-challenges like sentiment categorization, functional categorization, subject categorization, and other kinds of categorization (Bijalwan et al. 2014). Nevertheless, this paper is focused on text classification, or basically, the categorization of text into variant classes.

Automatic text classification is considered a supervised machine learning challenge in which a set of categorized texts is utilized for learning technique. This technique is used to allocate one or more predefined class tags to subjects. These subjects were categorized as different fields through variant methods.

A supervised classification method is used to utilize a group of predefined class texts produced as a learning model. TC considers a vital part in several fields like word sense disambiguation, information retrieval, web page categorization and in different areas that needs text arrangement (Yan and Xu 2010). Text classification can be used to detect documents on the similar subject. Nevertheless, the main challenges in utilizing this type of area is the especially hard categorization and limitation of documents for the similar subject (Chen et al. 2011).

2.3 Text classification process

Text classification process can be divided into six main stages which involve collection of data documents, Text preprocessing, feature extraction, dimensionality reduction, different classification techniques and performance evaluation as shown in Fig. 1.

2.4 Data collection

Data collection can be defined as the first stage and includes the acquisition of different datasets, which include different kind of format such as HTML, PDF, DOC etc.

2.5 Text preprocessing

Text preprocessing includes four steps: Tokenization, Removing stop words, Stemming and vector space model. Tokenization is used to remove the white space and special characters. Removing stop words are very common words, which is used to remove the informative data that carry little meaning, they serve only syntactic meaning but do not show subject matter

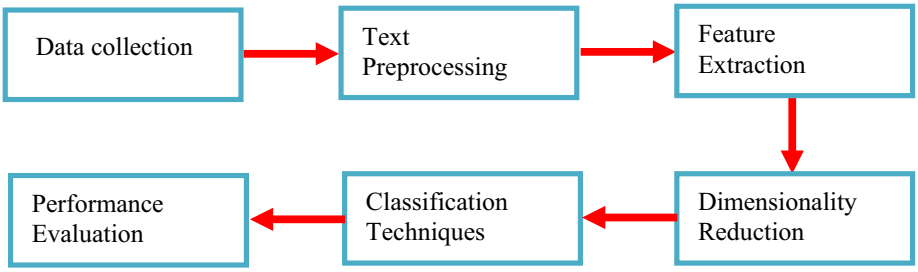

Fig. 1 Stages of text classification

it is well known among the conformation retrieval experts that a set of functional English words (e.g. “the”, “a”, “and”, “that”) is useless as weighting terms. These words have very low discrimination value, since they appear in every English text document (Sharma 2012). Thereafter, the set of words generated by word extraction is then checked so that every word occurring in the stop list is removed.

Stemming is used to remove the suffixes and prefixes from keywords that is the procedure stemmed by decreasing modified terms to their term stems. Therefore, the procedure is utilized to decrease the keywords number in the keyword space as well as the enhancement of the performance evaluation for the text classification when the variant shapes of keywords are derived as a single keyword. Furthermore, it is utilized commonly in information retrieval tasks to increase the recall rate (Sharma 2012) and get most related results like sky- > ski.

Vector space model: Naturally, each term in the learning group can be defined as a vector in the shape (x, td), where x ∈ Rn and R considers a vector of dimensions, n considers the number of terms, and the class tag is d. A single keyword of the vector space takes one dimension that it is computed by the frequency of appearance for each keyword in that document, that is, the text classification, information retrieval vector space model is repeatedly utilized as the data demonstration of documents.

2.6 Feature extraction

The major objective of feature extraction (FE) is used to convert a text from any setup into keyword schedule which it may be easy to process by supervised learning. Furthermore, FE presents knowledge concerning the texts like the maximum term frequency for each text. Choosing the related keywords and identifying the method is used to encode these keywords in a supervised machine learning. These keywords may have a huge impact on the classification techniques capability to extract the best pattern.

2.7 Feature weighting

TF-IDF technique is commonly utilized for FE and to identify terms those occur repeatedly in a text. The TF-IDF do not appear repeatedly in the whole for each document collection. Several studies showed the TF-IDF considers so active in extracting feature was proposed by Hu et al. (2018). The most common term weighting is TF-IDF. Let w be this term weighting, then the weight of each term ∈ d is calculated is shown in Eq. 1 as follows:

$$w(t) = f_d(w) * \log \frac{|D|}{f_D(w)} \tag{1}$$

where |D| can be defined as the number of documents in the collection D.

In TF-IDF the term frequency is normalized by inverse document frequency, IDF. This normalization declines the weight of the terms occurring more frequently in the document collection. Supposing that the matching of documents be more effected by characteristic terms which have relatively low frequencies in the collection.is shown in Eq. 2 as follows:

$$(TF - IDF)_{ih} = (TF)_{ij} * (IDF)_j \tag{2}$$

The improvement of the present implementation of TF-IDF can be solved by using TF-IDF with logarithm, as shown in Eq. 3.

$$\log(TF - IDF)_{ij} = \log(TF)_{ij} * \log(IDF)_j \tag{3}$$

Nevertheless, Eq. 3 is utilized only when $\log(\mathrm{TF})_{ij} \geq 1$. Otherwise, TF-IDF = 0 [see Eq. (4)] is given by (Kadhim et al. 2017).

$$\log(TF - IDF)_{ij} = \begin{cases} \log(TF - IDF)_{ij} & if\ \log(TF)_{ij} \geq 1 \\ 0 & otherwise \end{cases} \tag{4}$$

The traditional TF-IDF technique is utilized for FE. First, features that represent classes for documents were achieved. Thereafter, this document will be converted into arithmetical keywords using calculating the occurrences number that it's named term frequency (TF) in the document.

The keyword can be nominated according to the thresholding group for each documents collection. A significant step can be denoted as selecting proposed features which bring the document meaning and to refuse rest (Debole and Sebastiani 2004). Subsequent this step, the keywords from the individual documents are joined.

Several text categorization techniques utilized the traditional TF-IDF technique to weight keyword to enhance k-NN findings that it is used as an input for the keyword can attempt to realize a relation among techniques (Agarwal and Mittal 2012). Keywords can be divided into positive or negative cases according to the likely terms utilizing features set. Keywords cannot basically defined as the best individual or particular terms of a document such as keywords. While the other keywords also the frequencies number in a dataset of document can show an important in feature selection. If FE can be used to process as a supervised machine technique problem the combination of keywords variant types is direct method (Hira and Gillies 2015). According to Hira and Gillies (2015), a word co-occurrence distribution technique using a clustering method for extracting keywords for a single document without based on a huge dataset, and established hopeful findings. TF-IDF considers calculatingly effective and produced rationally well. FE can be treated as a supervised machine-learning issue. A term weighting procedure precisely considered for IR areas connecting supervised machine learning like document cleaning and text categorization that it is called as supervised term weighting (STW). The supervised word indexing influences on the learning data using weighting a word with respect to how variant its scattering may be in the negative and positive learning cases was proposed by Debole and Sebastiani (2004). Automatic learning and extraction of point expressions from technical papers that was printed in English. J48, an improved different of C4.5 can be resulted by Matsuo and Ishizuka (2004). A new method (ConfWeight) is used to weight keywords in the vector space model for TC using leveraging the categorization mission. In addition, TF-IDF is generally utilized to display the text keyword weight was presented by HaCohen-Kerner et al. (2005). Nevertheless, TF-IDF cannot denote the entire of words in document as well as it may be not to denote the significance degree and variance between classes. A new keyword weighting technique TF-IDF-C using adding a novel weight to display the variances between classes (Kuang and Xiaoming 2011). The importance of a word in a text can be calculated by its weight in text. Terms number weighting techniques can be conducted in works by Sharmila et al. (2014). The two different methods was presented the traditional models like bag of words, n-grams as well as the TF-IDF variants, and deep training models like word-based ConvNets and recurrent neural networks was proposed by Zhang et al. (2015).

2.8 Dimensionality reduction

Meanwhile, text categorization has been dealing with the large dataset. Therefore, dimension reduction is a vital in text categorization that it decreases the calculation budget as well as

it is not simply provides the highest dimensional document labeled. Dimension reduction may be given users through a stronger photo and visual investigation of the data of attention (Saeys et al. 2007).

Dimension reduction (DR) methods is divided as three central classes. The techniques group that they provide advantage of class-association information where the computing the lowest dimensional space. Instances of some methods contain a variant of feature selection procedure that they decrease the DR through selecting a subgroup of the original keywords (Li et al. 2003). The techniques that they derive new keywords using categorizing the word is called the first class (Tatu et al. 2009).

The aim of these DR methods can be used to decrease the knowledge cost related to the original data or to maintain the relationship space achieved in the corpus. The calculation methods based on statistical analysis. PCA, latent semantic indexing and multidimensional scaling (MDS) which were going to the dimension reduction techniques is called second class. This class is suitable to use in the linear relations between the dimensions (Marlow et al. 2006). The final class can be used a neuro calculation method is called self-organizing maps (SOMs) (the third class).

2.9 Categorization techniques

The categorization technique considers an efficient method that it used to construct the categorization pattern from an income group of data. This technique requests a supervised machine learning to recognize a pattern which recognizes the relation among the keywords group and class tag of the income data. This supervised machine learning technique must match the income data so fully as well as forecast the class tags of earlier unidentified registers.

Several variant techniques is utilized to categorize the documents into one or more areas according to their subjects like Naïve Bayes, support vector machine and k-nearest neighbor KNN.

2.10 Naïve Bayes

Naïve Bayes classifier considers a kind of pattern classifier that drop in variant pattern classifier of primly likelihood and class restricted likelihood (Lausch et al. 2015). This simplest likelihood based on providing the famous Bayes' formula. This formula based on robust (Naïve) self-determining hypothesis. This hypothesis is clearly disturbed in natural language processing: variant types of requirements among terms made by the conversational framework of document, semantic, syntactic and pragmatic. Consequently, the conditional independence assumption is utilized the likelihoods p(fi/c) are independence identified the category and thus the right category forecast that is computed by utilizing Eq. 5 as follows:

$$p(f_1, f_2, \ldots f_n/c) = p(f_1/c) \cdot p(f_2/c) \cdot_{\ldots} \cdot p(f_n/c) \tag{5}$$

The final equation that is used for the class chosen by a Naïve Bayes classifier by using Eq. 6 as follows:

$$c_{NB} = \arg\max_{c \in C} p(c) \prod_{f \in F} p(f/c) \tag{6}$$

In order to apply the Naïve Bayes classifier to document that it needs to consider word positions by simply keeps walking an index via every word position in the text document by using Eq. 7 as follows:

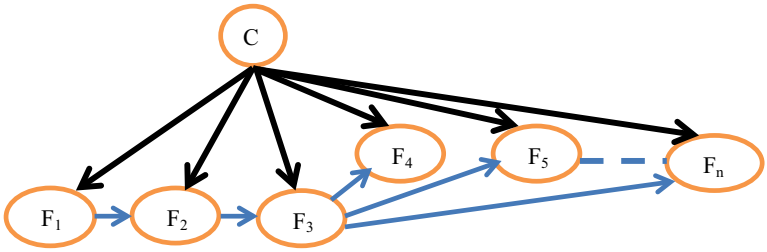

Fig. 2 The general idea of Naïve Bayes classifier

Positions ← all word positions in test text document.

$$c_{NB} = \arg\max p(c) \prod_{i \in position} p(w_i/c) \tag{7}$$

Naive Bayes computations, such as computations for text classification that were done in log space, to solve the underflow problem and increase speed, which was calculated by using Eq. 8. This kind of classifier needs a fewer quantity of learning data to accept the factors like means and variances of the variables that considered significant for categorization. Through studying and looking for the dependence between variant attributes, NB is so relaxed to achieve and calculate (Kunchala 2015).

$$c_{NB} = \arg\max_{c \in C} \log(x) + \sum_{i \in positions} \log(w_i/c) \tag{8}$$

By considering features in log space by using Eq. 8 calculates the predicted class as a linear function of input features. The classifier that applies in a linear combination of the inputs to implement a classification decision such as Naïve Bayes is called linear classifier. Figure 2 shows the general idea of Naïve Bayes.

2.11 Support vector machine

Support Vector Machine (SVM) classifier technique, introduced by Khamar (2013) to procedure two-class difficulties that based on looking a split among hyperplanes indicated by classes of data (Sahami et al. 1998). The training sample $\{(x_i, c_i)\}_{i=1}^{N}$, where the input topic for the ith example is and is the matched desired response (target output). In order to start with, it needs to assume that the topic (categories) indicated by the subset and the topic indicated by the subset are "linearly separable". In order to apply the SVM classifier for topic separation between hyperplanes that is used Eq. 9. This implies that the SVM classifier may procedure even in huge keyword datasets as the aim is to scale the border of split of the data, which utilized rather than fits on features. The SVM is learned utilizing predefined categorized texts.

$$w^T x + b = 0 \tag{9}$$

where x is an input of vector machine, w is an adjustable weight vector machine and b is a bias thus we can rewrite the Eq. 5 to become as follows:

$$\begin{aligned} w^T x_i + b \geq 0 \ \ for \ c_i = +1 \\ w^T x_i + b < 0 \ \ for \ c_i = -1 \end{aligned} \tag{10}$$

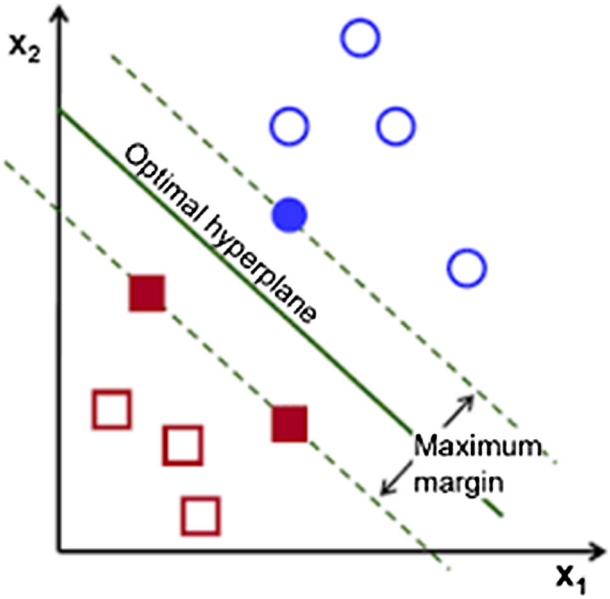

Fig. 3 The geometric building of an optimal hyperplane for two-dimensional input space

For a specified weight vector w and bias b, the Eq. 10 is used to separate between the hyperplane and the closest data point is represented by ρ that it is denoted margin of separation. The objective of a support vector machine is to obtain the highest of particular hyperplane for margin of separation ρ. In this case, the decision surface is indicated to as the optimal hyperplane. Figure 3 shows the geometric building of an optimal hyperplane for two-dimensional input space.

Let w0 and b0 define the optimum values of the weight vector and bias, respectively. Consequently, the optimal hyperplane, denoting a multidimensional linear decision surface in the input space is denoted by Eq. 11 as follows:

$$w_0^T x + b_0 = 0 \tag{11}$$

We may rewrite the Eq. 11; the discriminant function is calculated using Eq. 12 as follows:

$$g(x) = w_0^T x + b_0 \tag{12}$$

The algebraic measure of the distance takes from x to the optimal hyperplane (Vapnik 2000). We can rewrite Eq. 13 to be easy for using as follows:

$$x = x_\rho + r\frac{w_o}{\|w_o\|} \tag{13}$$

In Burges (1996), the researcher has displayed the measures sound by a worthy performance evaluation on large datasets. Both NB and SVM classifiers considered as linear, scalable and efficient to huge text corpus (Chen 2018).

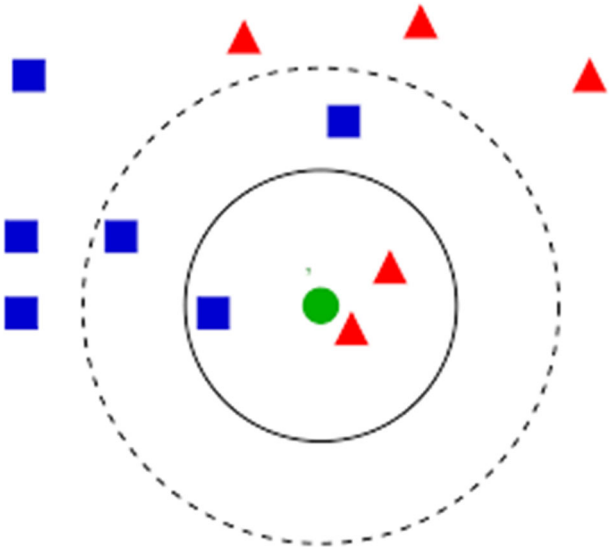

Fig. 4 The common knowledge of k-NN classifier

2.12 K-nearest neighbors

k-Nearest Neighbors (k-NN) is used as a classifiers. The central basic knowledge of the identification is used the class of a specified query with respect not only on the text which is closest to it in the text space. As well as classes of the k texts which can be closed to it (Kwok 1998). The k-NN technique is a similarity-based learning method which can be displayed to be so active for variant of issue areas containing text categorization (Rennie and Rifkin 2001). Specified a test text to identify the class is go to whereas the k-NN classifier explores the closest neighbors among the learning texts then uses the classes of the k neighbors to give weighting to the class candidates. Figure 4 displays the common knowledge of k-NN classifier.

The traditional k-NN classifier utilizes the Euclidean distance formula to categorize the texts into one or more predefined classes according to their subjects. Therefore, the classifier utilizes the keywords of texts that have be compared with the keywords of new texts. Namely, unless anyone has a classified corpus {xi}, and it needs to categorize particular new element y, the k-NN classifier looking for the k keywords in the corpus which may be neighboring to element y, afterward be about their classes to become the class of element y. Afterward, the classifier tests for possible classes for the text by determining learning texts that have keywords closet like to it. The classifier supposes the likely texts like topics in the Euclidean space. The space between two topics in the Euclidean space is used the Euclidean distance formula. The space between two topics in the level with assortment $p =$ (x1, x2) and q = (a, b) can be calculated by Eq. 14, is given by (Fix and Hodges 1951)

$$y = d(p, q) = d(q, p) = \sqrt{(x_1 - a)^2 + (x_2 - b)^2} \tag{14}$$

The similarity formula score of each neighbor text to the test text that used to give weighting of the classes of the neighbor text. Unless several k-nearest neighbors are existed to contribute a class, thereafter the neighbor weights of that class are added together as well as the finding weighted amount that is used as the likelihood score of likely classes. An ordered schedule

Table 1 The functions of similarity measurements

Manhattan distance	$\left\|(x_{i1} - x_{j1}) + (x_{i2} - x_{j2}) + \cdots (x_{in} - x_{jn})\right\|$
Minkowski distance	where i $= x_{i1}, x_{i2},\ldots x_{in}$ and j $= x_{j1}, x_{j2},\ldots x_{jn}$ $\left\|(x_{i1} - x_{j1})^p + (x_{i2} - x_{j2})^p + \cdots (x_{in} - x_{jn})^{\frac{1}{p}}\right\|$ where i $= x_{i1}, x_{i2},\ldots x_{in}$, j $= x_{j1}, x_{j2},\ldots x_{jn}$ and p can be defined as the positive integer
Jaccard coefficient	J(A,B) = (A∩B)/(AUB) where A and B are documents

is obtained for the test text. Therefore, text categorization is based on the thresholding for these scores and double class is found (Masand et al. 2012) as shown in Eq. 15.

$$s(x, y) = \frac{x^t}{\|x\|\|y\|} \tag{15}$$

where x^t can be defined as the transposition vector x, $\|x\|$ can be defined as the Euclidean norm of vector x, $\|y\|$ can be defined as the Euclidean norm of vector y.

In addition, there are different functions that were used to find the similarity between the documents is normally calculated based on the distance between document pairs was given by Trstenjak et al. (2014) as shown in Table 1.

2.13 Supervised machine learning techniques for automatic text classification

Actually, several supervised machine learning techniques is used to text categorization. Text categorization considers an important study field of text mining that the texts can be categorized with unsupervised, semi-supervised and supervised knowledge. Conventionally, this procedure may be resolved manually, however some manual categorization may be very expensive to measure as well as required to lengthy time for categorization and the labour exhaustive also. Consequently, investigators discovered the habit of supervised machine learning techniques to automatic texts categorization (Pitigala and Li 2015). The supervised machine learning that underlying input–output comparative is learned utilizing little number of learning data and then output values for unknown input texts have been forecasted (Han et al. 2011).

Text categorization is the allocating a predefined class according to their subjects. Therefore, categorization in machine learning issue considers a competence of supervised machine learning because of the training process was "supervised" utilizing the information of the classes as well as of the learning subjects which is related to them (Allahyari et al. 2017). Automatic text classification considered as supervised machine learning whereas a group of labeled texts is utilized for learning a classifier thereafter the classifier is used to allocate one or more predefined class labels to new texts (Ikonomakis et al. 2005).

According to Sugiyama and Kawanabe (2012) enhanced k-NN classifier for text categorization, which builds the categorization pattern utilizing mixing, limited to one badge gathering process and k-NN categorization. The adjusted k-NN technique using converting the bias on larger class in the traditional k-NN technique. The technique used stratified on Chinese text simply thus, the technique might be generally applicable to resolve categorization issues for data at variant languages was presented by Saeys et al. (2007). The efficiency of classification by eliminating those instances which are too far away from query instance using 8-bin hashing technique thus reducing the computational time was enhanced by Qi and

Davison (2009). Moreover, the accuracy of classification by weighting the features using positive instances based feature weighting algorithm was improved. Furthermore, the limitation text classification speed was enhanced and does not improve the accuracy of the k-NN classification. In the improvement of k-NN classification performance, further research is needed. The weight modified k-NN categorization technique which based on the k-NN categorization pattern. Several real life text data sets show the promise of WIND, as it is outperformed the state of the art categorization such as C4.5, VSM, RIPPER, Rainbow, and PEBBLES was proposed by Jiang et al. (2012).

Similar to this study Rane et al. (2014) enhanced the performance evaluation of the k-NN technique with a term weighting and a feature selection technique using maintain and markup tags its text–text similarity score. Long web pages are required long time to categorize of the entire Website. K-NN technique utilizing mixing k-NN and genetic algorithm in order to enhance the categorization performance. This enhanced in the accuracy value of categorization by decreasing the difficulty of the k-NN. Furthermore, the result performance is matched with the normal k-NN, CART and SVM classifiers was improved by Vapnik (2000).

In the same context, text categorization by utilizing AkNN text classifier and kMdd gathering. The local weighting TF method for feature selection and the cosine similarity score in AkNN classifier technique are used. The traditional k-NN based on Reuters-21578 and compared with other classifiers. They found that the traditional k-NN outperformed the results in both categorization and clustering was studied by Sharmila et al. (2014).

KNC algorithm for combining KNN algorithm and other three classifiers (C4.5 algorithm, Naive Bayes classifier and SVM) based on their classification capabilities on different types of instances. They found that KNC algorithm can enhance accuracies of KNN algorithm, algorithm, Naive Bayes classifier, C4.5 and SVM. KNC algorithm finds comparative performances with AdaBoost.M1 algorithm for both Naive Bayes and SVM classifier was utilized by Bijalwan et al. (2014). They stratified four machine learning techniques such as Language Model (LM), TF-IDF, k-NN and NB. Categorizations used through trending topics training corpus. A new supervised machine like k-NN that is used to train the pattern as well as to return the related texts was presented by Han et al. (2001). The likelihood of using the mixing of k-NN classifier, TF-IDF and framework for text categorization. Whereas the framework allows categorization according to variant factors, scores and analysis of findings was proposed by Kwon and Lee (2003). Four classifiers (decision DT, support vector machine SVM, Naïve Bayes NB, and k nearest neighbors k-NN) to ontology-based multi-label for text classification. They found the best classifier is Naïve Bayes coupled with the binary relevance transformation approach for single-label, while the best value is HOMER classifier for multi-label with respect to the evaluation metrics was used by Suguna and Thanushkodi (2010).

The Naïve Bayes (NB) classifier is a private of simple probabilistic classifiers depended on a public assumption that all features are independent of each other, specified the class variable. Moreover, they proposes three Bayesian counterparts, where it turns out that classical NB classifier with Bernoulli event model is corresponding to Bayesian counterpart based on 20 newsgroups and WebKB dataset was studied by Kurada and Pavan (2013).

Text categorization used a new method which needs less texts for learning. Naïve Bayes classifier depend on stemmed keywords and added genetic algorithm for the last categorization are used. They showed that the increased both data for learning and the computation time that impact on the accuracy was presented by (Bijalwan et al. 2014).

An effective and efficient technique like Naïve Bayes algorithm for classifying text classification to provide feasible information retrieval. They found the combinational term "Weight matrix" provides an extra weight and balance weight where needed. Moreover, they found

that the NB with weight matrix seldom reduces accuracy compared to standard Naïve Bayes instead of improving accuracy dramatically was introduced by Vogrinčič and Bosnić (2011).

Several studies in text classification have only executed in a small number of fields. The experiment results between SVM and Naive Bayes classifier under text enrichment via Wikitology was compared by Xu (2018). They found that the Naive Bayes classifier outperformed when external enriching is utilized via any external knowledge base. A method based on Naive Bayes learning support vector machine. The Naive Bayes algorithm been used to train the support vector machines, SVM is utilized to new text classification. They found that the method proposed are not only more dependable, but also further enhance the precision classification according to compare with traditional support vector machines algorithm was proposed by Kamruzzaman and Haider (2010).

Similar studies have been conducted Shathi et al. (2016) applied three different text classifier models like the vector space model, the Naive Bayes Classifier model (NB) and newly implemented based on two different datasets namely 20 Newsgroup and New dataset consisting of comparatively less data. They found that the NB classifier worked significantly better than remaining two classifiers. The SVM and Naïve Bayes classifiers to identify to each article based on ANT corpus its accurate predefined class. They found that the SVM technique outperformed the results of classification Naïve Bayes of both title and text parts was applied by Hassan et al. (2011).

On the other hand, utilizing of meta-features in automatic document classification has allowed significant enhancements in the efficiency of classification algorithms. Meta-feature methods is commonly used depend on intensive utilize of the k-NN algorithm to exploit local information concerning the neighborhood of learning documents was proposed by Canuto et al. (2014). A new method to hierarchical document classification like HDLTex that was combined multiple deep learning method to generate hierarchical classifications was introduced by (Kowsari et al. 2017). Detecting fake reviews through sentiment analysis using machine learning techniques was introduced by Elmurngi and Gherbi (2017). They found that the SVM algorithm outperformed the other algorithms and K* and KNN–IBK were taken the best time algorithms using movie reviews dataset V2.0 and V1.0, respectively.

For short text categorization, the sentiment and text categorization problem that displayed in online posts by discovering the latent relationship between tweet sentiments and texts was proposed by Huang et al. (2013). While opinion mining, it can be defined as the transition of automatically inforamtion finding from the text document that nation opinion. Thereby, opinion mining indicates to a large area of Text Mining, Natural Language Processing and Computational Linguistics was presented by Aytekin (2013).

3 Performance evaluation

Several different metrics that are used to evaluate the efficiency for any technique. Accuracy, recall, precision, and F1-measure that are commonly used (Sadiq and Abdullah 2012). Accuracy (Acci) can be defined as the ratio among the number of texts that were properly categorized, and the entire texts. Following that, recall (Ri) is defined as the ratio of properly categorized texts among all texts belonging to that class. While precision (Pi) is indicated as the ratio of properly categorized texts between all texts that can be identified to the class. Finally, F1-measure can be indicated to the symmetrical average for the precision and recall. These metrics are used to measure the final measurement of performance for any technique. Moreover, these values may be predicted in terms of the contingency table for category ci on

Table 2 The confusion matrix of the performance evaluation

Actual class	Predicted class	
	Categorized positive	Categorized negative
Actual positive	TP	FN
Actual negative	FP	TN

TP True positive, *FP* false positive, *FN* false negative, *TN* true negative

a given test set (see Table 2). Table 2 presents a confusion matrix is utilized to calculate the metrics.

True positives (TPi) can be denoted to the number of texts properly recognized in the particular class, true negatives (TNi) can be indicated to the number of texts properly which it's not recognized to a particular class. On the same context, false positives (FPi) can be indicated to the number of texts improperly recognized for a particular class, and false negatives (FNi) can be indicated to the number of texts improperly which it's not recognized to a particular class.

$$Acc_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{15}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{16}$$

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{17}$$

$$F1 - measure = \frac{2 \times P_i \times R_i}{(P_i + R_i)} \tag{18}$$

where i is the size of the dataset.

4 Comparison among classification techniques

The classification techniques were used to classify the documents into one or more class according to their contents. A huge amount of researchers is being presented to different classifiers in order to obtain a higher performance evaluation through developing the traditional technique that it is already existed. The most of the comparison among classification techniques can be summarized in Table 3.

According to Hao et al. (2009) the researcher proposed KNC algorithm that they used the combination of KNN and three and other three classifiers (C4.5 algorithm, Naive Bayes classifier and SVM) based on their classification capabilities on different types of instances. They found that the KNC algorithm performed better than C4.5 and KNN algorithm, but is performed better than AdaBoost.M1 with 100 decision trees produced by C4.5 algorithm (see Table 4).

The KNC algorithm performed better than Naive Bayes classifier, KNN algorithm and AdaBoost.M1 algorithm with 100 Naive Bayes classifiers (see Table 5).

KNC algorithm performed better than SVM and KNN algorithm, and found comparative performance with AdaBoost.M1 algorithm with 100 SVM classifiers (see Table 6).

Table 3 Summary of classification techniques for text classification

Nos.	Author name and year	Technique	Domain
1.	Yan and Xu (2010)	KNC & KNN SVM, NB and C4.5	UCI repository includes S-, DS- and D- instances
2.	Jiang et al. (2012)	KNN NB and SVM	Reuters-21578 and Fudan Univ. corpus
3.	Li et al. (2003)	Modified KNN Traditional KNN	Online web pages (Chinese language)
4.	Rane et al. (2014)	Improved KNN Standard KNN	Thyroid and Wine
5.	Han et al. (2001)	WAKNN KNN and VSM	TREC-5
6.	Kwon and Lee (2003)	KNN	Pages (18,445 pages) from hosts of 100 sites (online)
7.	Suguna and Thanushkodi (2010)	Genetic with KNN CARD and SVM	UCI repository
8.	Kurada and Pavan (2013)	Augmented (AKNN) K-Mediods (KMdd)	Reuters-21578
9.	Bijalwan et al. (2014)	KNN	Reuters-21578
10.	Vogrinčič and Zoran (2011)	KNN SVM and NB	1015 economic paper abstracts and semi-automatically constructing an ontology
11.	Shathi et al. (2016)	Weighted NB Standard NB	BBC news article
12.	Hassan et al. (2011)	NB Improvement NB and SVM	Reuters-21578
13.	Hao et al. (2009)	Traditional SVM SVM with NB	Corpus Sogou
14.	Tilve and Jain (2017)	SVM NB and a new algorithm (POSC)	20 Newsgroups and New news dataset for five categories

Table 4 The average accuracy of KNC algorithm based on C4.5

Avg.	C4.5 (%)	KNN (%)	AdaBoost (%)	KNC (%)
1.	84.43	84.87	87.17	88.21

Table 5 The average accuracy of KNC algorithm based on NB

Avg.	NB (%)	KNN (%)	AdaBoost (%)	KNC (%)
1.	81.67	84.82	84.94	85.62

Table 6 The average accuracy of KNC algorithm based on SVM

Avg.	SVM (%)	KNN (%)	AdaBoost (%)	KNC (%)
1.	85.65	84.86	86.22	86.86

Overall, the combination of KNN algorithm enhanced the classification capabilities of these three classifiers. By using KNC algorithm found worse average accuracies than AdaBoost.M1 algorithm for C4.5 algorithm, but it used the combination only two classifiers.

Table 7 The average of F1-measure for Reuter-21578

K = 10	Technique		
	KNN	NB	SVM
Avg.	0.8597	0.8843	0.8628

Table 8 The average of F1-measure for Fudan university corpus

K = 5	Technique		
	KNN	NB	SVM
Avg.	0.7512	0.4590	0.7011

Table 9 The average of performance and standard deviation for traditional KNN and modified KNN

K = 5–60	The traditional KNN				The modified KNN			
	Micro-avg.	Micro-avg.			Micro-avg.	Micro-avg.		
	Pre = Rec = F1	Pre	Rec	F1	Pre = Rec = F1	Pre	Rec	F1
Avg.	68.64	74.03	60.56	63.03	70.05	70.58	64.92	65.51
STDev.	2.38	1.27	4.08	3.45	1.46	1.64	1.94	1.26

Table 10 The comparison of accuracy for classification algorithms

Classification algorithms	Accuracy
NB	79.7
KNN–IBK (K = 3)	70.85
K*	71.15
SVM	81.35
DT-J48	71.60

Comparison among three different classifiers such KNN, NB and SVM based on Reuters-21578 and Fudan University corpus dataset was introduced by Tilve and Jain (2017). They found that the KNN improved better than NB and SVM according to F1-measure (see Tables 7, 8 respectively). From the Tables, the average of F1-measure for Rueuters-21578 increased better than Fudan University corpus due to the number of training text documents.

The traditional KNN and the modified KNN were presented by Chouigui et al. (2017). They found that the modified KNN outperformed the traditional KNN with respect to the average of performance expect the precision. Moreover, the standard deviations of modified KNN was reduced compared with the traditional KNN due to the modified KNN behaves quite stably with different values of k (see Table 9).

Five different algorithms (NB, KNN, K*, SVM and DT-J48) were used to compare between two different dataset (Movie review dataset V2.0 and Movie review dataset V1.0). Table 10 shows the experimental results for comparison of accuracy among algorithms using movie review dataset V2.0. Table 11 shows the time that was taken for each algorithm to generate prediction model using movie review dataset V2.0.

Table 12 shows the experimental results for comparison of accuracy among algorithms using movie review dataset V1.0. Table 13 shows the time that was taken for each algorithm to generate prediction model using movie review dataset V1.0.

Table 11 Time taken to generate prediction model

Classification algorithms	Time taken (ms)
NB	110
KNN–IBK (K = 3)	10
K*	0
SVM	14,840
DT-J48	340

Table 12 The comparison of accuracy for classification algorithms

Classification algorithms	Accuracy
NB	70.9
KNN-IBK (K = 3)	70.5
K*	69.4
SVM	76
DT-J48	69.9

Table 13 Time taken to generate prediction model

Classification algorithms	Time taken (ms)
NB	90
KNN-IBK (K = 3)	0
K*	10
SVM	4240
DT-J48	330

In terms of accuracy, SVM is the best algorithm for all tests (81.35%) using movie reviews in dataset V2.0 and (76%) using movie reviews in dataset V1.0. SVM tends to be more accurate than other methods.

While in time that was taken to generate the prediction model, the K* is the best algorithm for all tests (0) milliseconds using movie reviews in dataset V2.0 and the KNN-IBK is the best algorithm for all tests (0) milliseconds using movie reviews in dataset V1.0.

From the tables, they found that the SVM algorithm outperformed the other algorithms according to the accuracy. While, the time required for classification algorithms is performed better in K* algorithm as compared with other algorithms using movie reviews in dataset V2.0 and KNN algorithm is performed better than other algorithms using movie reviews in dataset V10. Moreover, it observed that well-trained machine learning classification algorithms is performed very useful classifications on the sentiment movie reviews was proposed by Elmurngi and Gherbi (2017).

The comparison between two different classifiers SVM and NB based on the experimental results was introduced by Mudgal and Munjal (2015). They found that the enhancement from 0.868 to 0.919 with respect to the average of F1-measure for micro and 0.865 to 0.920 for macro applied by SVM. While they found that the enhancement from 0.693 to 0.881 with respect to the average of F1-measure for micro and 0.681 to 0.877 for macro based the same experimental. It is clear that the improvement of 6.36% and 28.78% is performed on SVM and NB classifier respectively, by mixing information extracted from Wikitology.

Two different techniques that were used to text classification based on Corpus Sogou dataset was suggested by Masand et al. (2012). SVM based on NB performed better than

Table 14 The performance evaluation and speed classification for Corpus Sogou dataset

Technique	Pre	Rec	F score (%)	Speed classification
Traditional SVM	0.84	0.82	83.33	6.3467/s
SVM based on NB	0.86	0.86	86.89	0.3172/s

traditional SVM according to performance evaluation. Moreover, the speed of classification was reduced by using SVM based on NB (see Table 14).

The overall of the comparison, several different datasets with different classifiers have been studied by authors. These classifiers were developed based on the traditional technique with processed some modification like combination two classifier (SVM with NB). These improvements of any technique must be compared with other techniques to prove it's better than others. Moreover, the number of training dataset is a significant factor that it's impact of the experimental results for texts classification.

5 Discussions

There exist several issues that are evident in the text classification process. It was observed that the data collection process invariably affects the subsequent steps of preprocessing, feature extraction, and ultimately text classification.

Text preprocessing The type of data formed that are collected in different methods of preprocessing with different analysis results. The dataset must be prepared via tokenization, stop words removal, stemming and vector space document in order to be easy to use it.

Feature extraction Carrying out functions affecting the model extraction in the operations like the word frequency choice made at the level of text preprocessing, feature extraction and dimensional reduction, which can improve the text classification with respect to the performance evaluation.

Text classification From the related work, one can observe that the factors affecting the text classification can be summarized:

The number of learning texts The number of learning texts: The increasing of the learning texts, the evaluation is increased also. This attributed to increase the terms number that it is manually composed to a class generated a good categorization of sample texts.

Insufficient number of samples of testing text documents,
Features for the dataset,
Differences in techniques,
Incompatibility between the techniques and the problems,
Class ambiguity.

Class ambiguity represents the cases in which no distinction can be obtained with the features specified within the text classification problem using any classification technique.

Another factor which makes text classification more difficult is the scarcity of data. Classifying cases with insufficient number of examples would limit the generalization capability the classifiers and would likely result in a classifier to be randomly chosen. Naïve Bayes and SVM are linear classifiers with the assumption that the dataset is normally distributed, while k-NN is based on the similarity between two text documents, and can function as a non-linear classifier when kernel intensity predictors are used (Mudgal and Munjal 2015).

6 Conclusions

The increasing use of the textual data has resulted in the need for automatic text classification. This paper is presented the standardization stages of text classification like data collection, text preprocessing, feature extraction, text classification algorithms and performance evaluation. Consequently, this can be achieved using supervised machine learning techniques to organize, extract feature from the text documents. The stages of text classification literature focusing on supervised machine learning techniques like NB, SVM and k-NN are surveyed. The experimental results showed that the combination of relation information into the classification process can meaningfully enhance the quality of the underlying results. In addition, the existing classification algorithms are compared and contrasted depend on different parameters namely criteria utilized for classification, algorithms adopted. Finally, the performance metrics was discussed in order to evaluate each technique. From the above discussion it is understood that different techniques perform differently depending on the dataset (short and long) text. However, it was observed that k-NN with TF-IDF term weighting representation scheme performs well in several text classification algorithms.

References

Agarwal B, Mittal N (2012) Text classification using machine learning methods–a survey. In: Proceedings of the second international conference on soft computing for problem solving (SocProS 2012), December 28–30. Springer, New Delh, pp 701–709

Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut KA (2017) Brief survey of text mining: classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919

Aytekin Ç (2013) An opinion mining task in Turkish language: a model for assigning opinions in Turkish blogs to the polarities. J Mass Commun 3(3):179–198

Bijalwan V, Kumar V, Kumari P, Pascual J (2014) KNN based machine learning approach for text and document mining. Int J Database Theory Appl 7(1):61–70

Bindra A (2012) "SocialLDA: scalable topic modeling in social networks". Dissertation University of Washington

Burges CJC (1996) Simplified support vector decision rules. In: ICML, Vol. 96, pp 71–77

Canuto S, Salles T, Gonçalves MA, Rocha L, Ramos G, Gonçalves L, Martins W (2014) On efficient meta-level features for effective text classification. In: Proceedings of the 23rd ACM international conference on conference on information and knowledge management. ACM, pp 1709–1718

Chen S (2018) K-nearest neighbor algorithm optimization in text categorization. In: IOP conference series: earth and environmental science. IOP Publishing, Vol. 108, No. 5, p 052074

Chen M, Jin X, Shen D (2011) Short text classification improved by learning multi-granularity topics. In: IJCAI, pp 1776–1781

Chouigui A, Khiroun OB, Elayeb B (2017) ANT Corpus: An Arabic news text collection for textual classification. In: IEEE/ACS 14th international conference on computer systems and applications (AICCSA). IEEE, pp 135–142

Debole F, Sebastiani F (2004) Supervised term weighting for automated text categorization. Text mining and its applications. Springer, Berlin, pp 81–97

Elmurngi E, Gherbi A (2017) Detecting fake reviews through sentiment analysis using machine learning techniques. In: IARIA/data analytics, pp 65–72

Feng Y, Zhaohui W, Zhou Z (2005) Multi-label text categorization using k-nearest neighbor approach with m-similarity. String Processing and Information Retrieval. Springer, Berlin

Fix E, Hodges JL Jr (1951) Discriminatory analysis-nonparametric discrimination: consistency properties. California University, Berkeley

HaCohen-Kerner Y, Gross Z, Masa A (2005) Automatic extraction and learning of keyphrases from scientific articles. In: Computational linguistics and intelligent text processing. Springer Berlin, pp 657–669

Han EHS, Karypis G, Kumar V (2001) Text categorization using weight adjusted k-nearest neighbor classification. Springer, Berlin, pp 53–65

Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier, Amsterdam

Hao P, Ying D, Longyuan T (2009) Application for web text categorization based on support vector machine. In: International forum on computer science-technology and applications, IFCSTA'09, Vol. 2. IEEE, pp 42–45

Hassan S, Rafi M, Shaikh MS (2011) Comparing SVM and Naive Bayes classifiers for text categorization with wikitology as knowledge enrichment. In: 14th international multitopic conference (INMIC). IEEE, pp 31–34

Hira ZM, Gillies DF (2015) A review of feature selection and feature extraction methods applied on microarray data. Adv Bioinf 2015:198363

Horecki K, Mazurkiewicz J (2015) Natural language processing methods used for automatic prediction mechanism of related phenomenon. In: Artificial intelligence and soft computing. Springer, pp 13–24

Hu J, Li S, Yao Y, Yu L, Yang G, Hu J (2018) Patent keyword extraction algorithm based on distributed representation for patent classification. Entropy 20(2):104

Huang S, Peng W, Li J, Lee D (2013) Sentiment and topic analysis on social media: a multi-task multi-label classification approach. In: Proceedings of the 5th annual ACM web science conference. ACM, pp 172–181

Ikonomakis M, Kotsiantis S, Tampakas V (2005) Text classification using machine learning techniques. WSEAS Trans Comput 4(8):966–974

Jiang S, Pang G, Wu M, Kuang L (2012) An improved K-nearest-neighbor algorithm for text categorization. Expert Syst Appl 39(1):1503–1509

Joseph F, Ramakrishnan N (2015) Text categorization using improved K nearest neighbor algorithm. Int J Trends Eng Technol 4:65–68

Jothi CS, Thenmozhi D (2015) Machine learning approach to document classification using concept based features. Int J Comput Appl 118(20):33–36

Kadhim AI, Cheah Y-N, Hieder IA, Ali RA (2017) Improving TF-IDF with singular value decomposition (SVD) for feature extraction on Twitter. In: 3rd international engineering conference on developments in civil and computer engineering applications 2017 (ISSN 2409-6997)

Kamruzzaman SM, Haider F (2010) A hybrid learning algorithm for text classification. arXiv preprint arXiv:1009-4574

Khamar K (2013) Short text classification using kNN based on distance function. In: IJARCCE International Journal of Advanced Research in Computer and Communication Engineering. Government Engineering College, Modasa (ISSN Print: 2319-5940 ISSN Online, pp 2278–1021

Kowsari K, Brown DE, Heidarysafa M, Meimandi KJ, Gerber MS, Barnes LE (2017) Hdltex: hierarchical deep learning for text classification. In: 2017 16th IEEE international conference on machine learning and applications (ICMLA). IEEE, pp 364–371

Kuang Q, Xiaoming X (2011) An improved feature weighting method for text classification. Adv Inf Sci Service Sci 3(7):340–346

Kunchala DR (2015) Applying data mining techniques to social media data for analyzing the student's learning experience. Ph.D. Dissertation, Texas A&M University-Corpus Christi

Kurada RR, Pavan DKK (2013) Novel text categorization by amalgamation of augmented k-nearest neighborhood classification and k-medoids clustering. arXiv preprint arXiv:1312.2375

Kwok JT-Y (1998) Automated text categorization using support vector machine. In: Proceedings of the international conference on neural information processing (ICONIP 1998)

Kwon O-W, Lee J-H (2003) Text categorization based on k-nearest neighbor approach for web site classification. Inf Process Manag 39(1):25–44

Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. AAAI 333:2267–2273

Lausch A, Schmidt A, Tischendorf L (2015) Data mining and linked open data—new perspectives for data analysis in environmental research. Ecol Model 295:5–17

Li B, Yu S, Lu Q (2003) An improved k-nearest neighbor algorithm for text categorization. arXiv preprint arXiv:cs/0306099

Marlow C, Naaman M, Boyd D, Davis M (2006) HT06, tagging paper, taxonomy, Flickr, academic article, to read. In: Proceedings of the seventeenth conference on hypertext and hypermedia. ACM, pp 31–40

Masand VH, Mahajan DT, Patil KN, Chinchkhede KD, Jawarkar RD, Hadda TB, Alafeefy AA, Shibi IG (2012) k-NN, quantum mechanical and field similarity based analysis of xanthone derivatives as α-glucosidase inhibitors. Med Chem Res 21(12):4523–4534

Matsuo Y, Ishizuka M (2004) Keyword extraction from a single document using word co-occurrence statistical information. Int J Artif Intell Tools 13(01):157–169

Moreno A, Redondo T (2016) Text analytics: the convergence of big data and artificial intelligence. IJIMAI 3(6):57–64

Mudgal A, Munjal R (2015) Role of support vector machine, fuzzy K-means and Naive Bayes classification in intrusion detection system. Int J Recent and Innov Trends Comput Commun 3:1106–1110
Pitigala S, Li C (2015) Classification based filtering for personalized information retrieval. In: Proceedings of the international conference on information and knowledge engineering (IKE). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), pp 125–131
Qi X, Davison BD (2009) Web page classification: features and algorithms. ACM Comput Surv (CSUR) 41(2):12
Rane A, Naik N, Laxminarayana JA (2014) Performance enhancement of K nearest neighbor classification algorithm using 8-bin hashing and feature weighting. In: Proceedings of the 2014 international conference on interdisciplinary advances in applied computing. ACM, p 8
Rennie JDM, Rifkin R (2001) Improving multiclass text classification with the support vector machine
Sadiq AT, Abdullah SM (2012) Hybrid intelligent technique for text categorization. In: International conference on advanced computer science applications and technologies (ACSAT). IEEE, pp 238–245
Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517
Sahami M, Dumais S, Heckerman D, Horvitz E (1998) A Bayesian approach to filtering junk e-mail. Learn Text Categ 62:98–105
Sebastiani F (2006) Classification of text, automatic. Encycl Lang Linguist 14:457–462
Sharma D (2012) Stemming algorithms: a comparative study and their analysis. Int J Appl Inf Syst 4(3):7–12
Sharmila V, Vasudevan I, Arasu GT (2014) Pattern based classification for text mining using fuzzy similarity algorithm. J Theor Appl Inf Technol 63(1):92–103
Shathi SP, Hossain MD, Nadim M, Riayadh SGR, Sultana T (2016) Enhancing performance of Naïve Bayes in text classification by introducing an extra weight using less number of training examples. In: International workshop on computational intelligence (IWCI). IEEE, pp 142–147
Sugiyama M, Kawanabe M (2012) Machine learning in non-stationary environments: introduction to covariate shift adaptation. MIT Press, Cambridge
Suguna N, Thanushkodi K (2010) An improved K-nearest neighbor classification using Genetic Algorithm. Int J Comput Sci Issues 7(2):18–21
Tatu A, Albuquerque G, Eisemann M, Schneidewind J, Theisel H, Magnork M, Keim D (2009) Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In: IEEE symposium on visual analytics science and technology, 2009, VAST 2009, pp 59–66
Tilve AKS, Jain SN (2017) A survey on machine learning techniques for text classification. Int J Eng Sci Res Technol 6:513–520
Trstenjak B, Mikac S, Donko D (2014) KNN with TF-IDF based framework for text categorization. Proc Eng 69:1356–1364
Vapnik V (2000) The nature of statistical learning theory. Springer, New York
Vogrinčič S, Bosnić Z (2011) Ontology-based multi-label classification of economic articles. Comput Sci Inf Syst 8(1):101–119
Xu S (2018) Bayesian Naïve Bayes classifiers to text classification. J Inf Sci 44(1):48–59
Yan Z, Xu C (2010) Combining KNN algorithm and other classifiers. In: 2010 9th IEEE international conference on cognitive informatics (ICCI). IEEE, pp 800–805
Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. In: Advances in neural information processing systems, pp 649–657