J Biochem Tech (2019) Special Issue (2): 174-182
ISSN: 0974-2328

Document Classification Base on Ensemble Classifiers Support Vector Machine, Multi layer Perceptron and k-Nearest Neighbors

Somaye Esmaeili Rad* and Amir Rajabi Behjat

Received: 20 September 2018 / Received in revised form: 10 March 2019, Accepted: 21 March 2019, Published online: 25 April 2019

Abstract

Text Categorization (TC), also known as Text Classification, is the task of automatically classifying a set of text documents into different categories from a predefined set. TC uses several tools from Information Retrieval (IR) and Machine Learning (ML) and has received much attention in the last years from both researchers in the academia and industry developers. In this paper, we first categorize the documents using of three algorithm KNN, MLP, SVM based machine learning approach and two Data set: "Routers" and "Hamshahri" that the idea of combining multiple classifiers has been reviewed well and by combining Support vector machine (SVM), k-Nearest Neighbors (KNN) and Multi layer perceptron (MLP) algorithms,text and configuration of the "Hamshahri" for Persian documents and "Reuters" for English documents is classified. The used criteria for assessing and accuracy, along with experimental results on the Configure of the Hamshahri and Reuters by using of SVM, MLP and KNN algorithms indicated that the combination of algorithms and feature selection methods, while reducing the number of features, improves the efficiency and accuracy in the combining classifiers system.

Key words: Text Mining, k-Nearest Neighbors, Multi Layer Perceptron, Support vector machine, Classifier ensembles, Machine Learning.

Introduction

Dietterich (2002) Ensemble learning is a new direction of machine learning, which trains a number of specific classifiers and selects some of them for ensemble. It has been shown that the combination of multiple classifiers could be more effective compared to any individual ones and a popular method for creating an accurate classifier from a set of training data is to train several classifiers, and then to combine their predictions.

Ekbal and Saha (2011) from a technical point of view, ensemble learning is mainly implemented as two steps: training base classifiers and selectively combining the member classifiers by a stronger classifier. Usually the members of an ensemble are constructed in two ways. One is to apply a single learning algorithm, and the other is to use different learning algorithms over a dataset. Zhou and Tang (2002) Then, the base classifiers are combined to form a decision classifier. Generally, to get a good ensemble, the base learners should be as more accurate as possible and as more diverse as possible. So how to choose an ensemble of some accurate and diverse base learners is a focus of concern of many researchers.

Kumari and Jain and Bhatia (2016) among the most popular combination schemes, majority voting and weighted voting for classification are widely used. Simple majority voting is a decision rule that selects one of many alternatives, based on the predicted classes with the most votes. Majority voting does not require any parameter tuning once the individual classifiers have been trained.

Yong and Zhang and Cai and Yang (2014) In case of weighted voting, weights of votes should vary among the different output classes in each classifier. The weight should be high for that particular output class for which the classifier performs well. So, it is a crucial issue to select the appropriate weights of votes for all the classes per classifier. Weighting problem can be viewed as an optimization problem.

Somaye Esmaeili Rad*
Master of artificial intelligence, Department of Computer, Rafsanjan Branch, Islamic Azad University, Rafsanjan, Iran.

Amir Rajabi Behjat
Faculty member, Department of Computer, Rafsanjan Branch, Islamic Azad University, Rafsanjan, Iran.

***Email:** somaye_esmaeili_rad@yahoo.com

Therefore, it can be solved by taking advantage of classification techniques such as MLP, KNN,MLP.

In this paper we present a technique for building ensembles MLP, KNN, SVM classifiers in random subspaces. We used Tf IDF, which improves accuracy and diversity of the base classifiers. We conduct a number of experiments on a collection of Reuters and Hamshahri data sets.

The rest of this paper is structured as follows. The proposed ensemble approach is based on MLP, SVM and KNN. Section 2 first introduces KNN, SVM and MLP. Then, Section 3 introduces the proposed ensemble approach for classifying the texts. Experimental results approaches in Section 4. Conclusions are finally drawn in Section 5.

Classification Methods

The proposed ensemble approach is based on MLP, SVM, and KNN. In this section, the brief descriptions of KNN, SVM and MLP are introduced.

The brief description of k-Nearest Neighbors

Vishwanath and Kumari and Pascual and Semwal (2014) The initial application of k-Nearest Neighbors (KNN) to text categorization was reported in The basic idea is to determine the category of a given query based not only on the document that is nearest to it in the document space, but on the categories of the k documents that are nearest to it. Having this in mind, the Vector method can be viewed as an instance on the KNN method, where k=1. This work uses a vector-based, distance-weighted matching function, as did Yang, by calculating document's similarity like the Vector method .Then, it uses a voting strategy to find the query's class: each retrieved document contributes a vote for its class, weighted by its similarity to the query. The query's possible classifications will be ranked according to the votes they got in the previous step.

Algorithm:

1) Make vector for every document in the test set.
2) Make centroid vector for each class.
3) Calculate similarity between each document vector and class vector.
4) Document belongs to the class for which the similarity is maximum.

The brief description of Support vector machines

Vladimir (2013) In today's machine learning applications, support vector machines (SVM) are considered a must try it offers one of the most robust and accurate methods among all well-known algorithms. It has a sound theoretical foundation, requires only a dozen examples for training, and is insensitive to the number of dimensions. In addition, efficient methods for training SVM are also being developed at a fast pace.

Cunhe and Liu and Wang (2011) In a two-class learning task, the aim of SVM is to find the best classification function to distinguish between members of the two classes in the training data. The metric for the concept of the "best" classification function can be realized geometrically. For a linearly separable dataset, a linear classification function corresponds to a separating hyper plane f (x)that passes through the middle of the two classes, separating the two. Once this function is determined, new data instance x_n can be classified by simply testing the sign of the function f (x_n); x_n belongs to the positive class if f (x_n)> 0.

Because there are many such linear hyper planes, what SVM additionally guarantee is that the best such function is found by maximizing the margin between the two classes. Intuitively,

the margin is defined as the amount of space, or separation between the two classes as defined by the hyper plane. Geometrically, the margin corresponds to the shortest distance between the closest data points to a point on the hyper plane. Having this geometric definition allows us to explore how to maximize the margin, so that even though there are an infinite number of hyper planes, only a few qualify as the solution to SVM.

Ludmila and Rodríguez (2014) The reason why SVM insists on finding the maximum margin hyper planes is that it offers the best generalization ability. It allows not only the best classification performance (e.g., accuracy) on the training data, but also leaves much room for the correct classification of the future data. To ensure that the maximum margin hyper planes are actually found, an SVM classifier attempts to maximize the following function with respect to w and b

$$L_P = \frac{1}{2}\|\vec{w}\| - \sum_{i=1}^{t} \alpha_i y_i(\vec{w}.\vec{x}_i + b) + \sum_{i=1}^{t} \alpha_i \quad (1)$$

where t is the number of training examples, and α_i , i = 1,. . , t, are non-negative numbers such that the derivatives of L_P with respect to α_i are zero. α_i are the Lagrange multipliers and L_P is called the Lagrangian. In this equation, the vectors w and constant b define the hyper plane.

The brief description of Multi Layer Perceptron

Hagan and Demuth and Beale and De Jesús (1996) In 1958 Frank Rosenblatt proposed the Perceptron Model which he named as brain model. Perceptron model was used for solving pattern classification problems. Perceptron was the first model which was making use of supervised learning for training the network. The model mostly comprise of single neuron with adjustable weights and bias. Rosenblatt in his study proved that if the vectors used to train the network are taken from linearly separable classes then the algorithm which he named perceptron convergence algorithm will converge and position the decision surface in the form of hyper plane between the two classes. The proof of convergence is known as Perceptron Convergence Theorem. The Activation function used in the neuron is signum (sgn). The definition is:

$$sgn\ (v) = \begin{cases} +1\ if\ v > 0 \\ -1\ if\ v < 0 \end{cases}$$ here,v is the include local filed value (2)

The Equations for Update weight vector w (n+1), desired response d(n) and the actual response y(n)is given below:

Y(n)=sgn(wt(n) x(n)), where y(n) is actual response, x(n) is the input vector and wt(n) is weight matrix W(n+1)=w(n)+Ϭ[d(n)-y(n)]x(n), where Ϭ is learning constant and n denotes the iteration step

$$d(n) = \begin{cases} +1\ if\ x(n)\ belong\ to\ class\ €1 \\ +1\ if\ x(n)\ belong\ to\ class\ €2 \end{cases}$$ Where d(n)represents desired response (3)

Cunhe (2011) The perceptron model could solve only the problems which are linearly separable. Since there are various problems which are inherently non-linearly separable the solution was not possible through the perceptron model. The perceptron model was of single layer although there could be more than one neuron in the layer.

Hagan andDemuth and Beale and Jesús (1996) Fig 3 represents MLP. MLP exhibits three basic characteristics which are (a) The model of neuron in the network includes non linear activation function. However a special case is the minimum configuration MLP, (b) The hidden layers present in the network have inherent capacity to learn complex patterns present in the input pattern and (c) The network connectivity present in the MLP is of high degree. MLP uses supervised learning algorithm called Back propagation to train the network.

The ensemble approach

In ensemble learning, a collection of single classification or regression models is trained, and the output of the ensemble is obtained by aggregating the outputs of the single models, e.g. by majority voting in the case of classification, or averaging in the case of regression. shows that the result of the ensemble might outperform the single models when weak (unstable) models are combined, mainly because of three reasons:

a) several different but equally optimal hypotheses can exist and the ensemble reduces the risk of choosing a wrong hypothesis.
b) learning algorithms may end up in different local optima, and the ensemble may give a better approximation of the true function, and c) the true function cannot be represented by any of the hypotheses in the hypothesis space of the learner and by aggregating the outputs of the single models, the hypothesis space may be expanded.

In this paper, the ensemble approach consists of MLP, SVM, and KNN. The flow chart of the ensemble approach is shown in Fig 5.In this study, two test datasets are used to test the performance of the proposed ensemble approach. Firstly, Reuters were collected Reuters[1] Secondly, collect Hamshahri[2] as the second test dataset after data pre-processing, the ensemble approach, based on KNN, MLP and SVM, is applied to classify text.

For classifying the documents in Reuter-21578 and Hamshahri we initially pre-processed the data by performing techniques:

A. TF-IDF

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[2] http://ece.ut.ac.ir/dbrg/Hamshahri/faindex.html

B. Stop word removal

A. Term frequency

Inverse Document Frequency (TF.IDF) Murugesan and Keerthiram and Zhang (2011) is the most widely used and considered as one of the most appropriate term weighting schemes. This TF.IDF is employed to get rid of terms with lower weights from documents and helps to increase the retrieval effectiveness. Term frequency–inverse document frequency, is a numerical statistic

that tells us how important a word is to a document in a collection or corpus. It is mostly used as a weighting factor in various processes used for information retrieval and text mining. The increase in the TF.IDF (4) value of a word is directly proportional to the number of times that word occurs in the document, but is neutralized by the frequency of the word in the corpus, which helps to balance off those words which appear more frequently in general.

TF.IDF = (Term Frequency ∗ Inverse Document Frequency) (4)

Term Frequency (TF) measures how many times a term occurs in a document. Since documents have different lengths, so it can happen that the longer document contains a term more times than the documents which are shorter in length. Thus, to normalize it, the term frequency (2) is mostly.

TF = Total number of items in a document/Number of times a term appears in a document. (5)

Inverse Document Frequency (IDF), helps in determining the importance of a term. When we compute term frequency, we give equal importance to all the terms. But certain terms, such as “the”, “that”, and “is”, may appear very frequently which are not important. So, we need to bring down the weights of frequent terms and increase the weights of the rare terms, by calculating the following:

IDF = log2 (Number of document with term t in them)/Total number of documents. (6)

Learning process

Trstenjak and Mikac and Donko (2014) Learning process starts with parsing the basic text which searches the words in documents and forms a vector. Parsing process removes all control characters, spaces between words, dots, commas, and similar characters. The formed vector represents a fundamental object that will be used for classification of tested documents. Example 1 below illustrates the procedure for preprocessing search text and the formation of the main vector.

Example 1:
Search text: “Text classification, KNN,MLP,Svm method.”
Preprocessing: Text classification KNN ,MLP,Svm method
Base vector: mainVec = [Text classification KNN ,MLP,Svm method]
mainVec[0]= Text
mainVec[1]= classification
mainVec[2]= k-NN, MLP, Svm
mainVec[3]= method

Determination of the weight matrix

To provide text classification and searching the documents it is necessary to establish the weight matrix. The matrix contains the values of relations between each unique words and documents. It is the initial object in the algorithm to calculate the individual importance (weight) of each searched document. Each document is represented as a vector in n-dimensional vector space. We can imagine a matrix A with dimensions NxM, where N dimension is defined by a number of unique words in a sample of all documents. M represents the number of documents to be classified. Weight matrix can be characterized as a relational matrix of word - document. Dimension of the matrix is equal to product, the number of different unique words and the total number of documents. Each matrix element a_{ij} represents weight value of word i in the document j. Weight matrix is shown in Fig. 4 In determining the weight values in the matrix, we can use different metrics and methods of calculation.

B. Stopping

Kumari and Jain and Bhatia (2016) Stopping is the process of removing common words like “if”, “than”, “or”, “in”, “and”, “the”. It helps in increasing the efficiency and effectiveness in the information retrieval process. Some common words which are very less important in selecting documents according to the user need are removed. These words are called stop words.

Stop words are usually determined by sorting the terms by their frequency in the document collection and then the most frequent terms are taken as stop words, often exceptions are made for the words semantically related to the domain of the documents under consideration. The stop words from the stop list are then not included for the further processes such as stemming, indexing etc.

Then after pre-processing, we applied KNN, MLP, and SVM algorithms to classify the documents in the training set into seven categories. We further applied our classifier model on the test documents and calculated the accuracy by comparing it with the default answers given for the test documents. To compare the above mentioned algorithms, we used the following metric: Accuracy, which is defined as the percentage of correctly classified documents, is generally used to evaluate single-label TC tasks.

$$\text{Accuracy} = \frac{\#\text{Correctly classified documents}}{\#\text{Total documents}} \quad (7)$$

The last test was focused on measuring the quality of classification depending on the documents category.It was previously indicated that the quality of classification depends on the preprocessing documents, removing undesired characters and words that have no significant information. Table 1 shows the results of successful classification over the documents from various categories. The results show that the worst classification was performed in the category Daily News. The analysis of document contents in this category showed that documents contained a lot of "unusable words", the words that are often repeated and do not have important weight but have adverse impact on KNN, MLP, SVM classification.

Experimental Results

Dataset Reuters

Reuters[3] The data set used for this paper is in the form of sgml files. We have used Reuters-21578 dataset which is available at. There are 21578 documents; according to the „ModApte‟ split: 9603 training docs, 3299 test docs and 8676 unused docs. They were labeled manually by Reuters personnel. Labels belong to 7 different category classes, such as people, places, Daily News, Organization, Political, Sport and topics‟. The total number of categories is 672, but many of them occur only very rarely. The dataset is divided in 22 files of 1000 documents delimited by SGML tags.

Dataset Hamshahri

Hamshahri[4] Body of Hmshahri is collection of news articles, also it is the first online newspaper that more than 20 years to be released in Iran. It’s archives are available to the public. This configuration include 345MB of text with appropriate structures is labeled news .As regards very limited efforts in the field of application of classification and data mining has been done on Persian literature .It was decided to use the Hamshahri to test the classification algorithm. Hamshahri test series is one of the most reliable sources in Persian .The first version of the Hamshahri contains more Than 160000 documents and 65 related judgments and requests. That from 1375 to 1381 written by different people with different topics. the second version of the Hamshahri is large and more comprehensive Then the previous version, it also involves picture of article. Hamshahri authors manually split their article in to different categories, and placed them on Hamshahri sit. all documents are sorted in This collection of 82 different categories based on news set available on the website of the newspaper.

In this paper we present a framework for text classification based on KNN, SVM, MLP algorithm and the TF-IDF method .The main motivation for the research was to develop concept of frameworks with emphasis on KNN SVM, MLP & TF-IDF module. The framework with embedded methods gave good results, confirmed our concept and initial expectations. Evaluation of framework was performed on several categories of documents in online environment. Tests are supposed to provide answers about the quality of classification and to determine which factors have an impact on performance of classification. The framework work was very stable and reliable. During testing the quality of classification we have achieved good results regardless of the K factor value in the KNN algorithm. Performed tests have detected a sensitivity of the implemented algorithm. Tests have shown that the embedded algorithm is sensitive to the type of documents. The analysis of documents contents showed that the amount of unusable words in documents has a significant impact on the final quality of classification. Because of this, it is necessary improve the preprocessing of data for achieving better results.

[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[4] http://ece.ut.ac.ir/dbrg/Hamshahri/faindex.html

The combination of KNN algorithm and TF-IDF method has been shown as a good choice with minor modifications in their implementation. The framework provides the ability to upgrade and improve the present embedded classification algorithm.

The test accuracy of the proposed ensemble approach for the first dataset is 98.21%. The results of the first dataset are shown in Table 3.From Table 2, the proposed ensemble approach has the best performance among these compared approaches. For the second dataset. The test accuracy for the second dataset is shown in Table 2. It is easy to see that the proposed ensemble approach outperforms MLP, SVM and KNN, individually. The proposed ensemble approach has the best accuracy among those approaches for both datasets.

Figure 8 shows the ROC curves of the ensemble and the 3 individual models: SVM, KNN and MLP in the ensemble. It clearly shows that the ensemble considerably improved the performance in classifying the texts.

Conclusions

In this paper, an ensemble approach, based on K-nearest neighbor, support vector machine and Multi Layer Perceptron, is applied to classify text. The text are first divided into 14 features and the ensemble approach is used to classify them. From simulation results, the proposed ensemble approach has the best accuracy of classification among these compared approaches for two test datasets. In Table 1, the test accuracy for the first dataset is Routers In Table 2, the test accuracy for the second dataset is Hamshahri It indeed shows that the proposed ensemble approach outperforms other approaches.

References

Bijalwan, V., Kumari, P., Pascual, J., & Semwal, V. B. (2014). Machine learning approach for text and document mining. arXiv preprint arXiv:1406.1580.

Cunhe L., Liu K., Wang H. (2011). The incremental learning algorithm with support vector machine based on hyperplane-distance. Applied Intelligence 34, no. 1, 19-27.

Dietterich T., (2002). Ensemble learning in The Handbook of Brain Theory and Neural Networks, 2nd edition.

Ekbal A., Saha S., (2011). Weighted vote-based classifier ensemble for named entity recognition: a genetic algorithm-based approach, ACM Transactions on Asian Language Information Processing, vol. 10, no. 2, article 9, 37 pages.

Hagan, M.T., Demuth, H.B., Beale, M.H. and De Jesús, O.. (1996). Neural network design (Vol. 20). Boston: PWS publishing company.

Kumari, M., Jain, A., & Bhatia, A. (2016). Synonyms based term weighting scheme: an extension to TF. IDF. Procedia Computer Science, 89, 555-561.

Ludmila K., Rodríguez J.J. (2014). A weighted voting framework for classifiers ensembles. Knowledge and Information Systems 38, no. 2, 259-275.

Murugesan, A. Keerthiram, B.J. Zhang, (2011). A New Term Weighting Scheme for Document Clustering. In 7th Int. Conf. Data Min. (DMIN 2011-WORLDCOMP 2011), Las Vegas, Nevada, USA.

Trstenjak, B., Mikac, S., & Donko, D. (2014). KNN with TF-IDF based Framework for Text Categorization. Procedia Engineering, 69, 1356-1364.

Vapnik, V. (2013). The nature of statistical learning theory. Springer science & business media.

Zhang, Y., Zhang, H., Cai, J., & Yang, B. (2014). A weighted voting classifier based on differential evolution. In Abstract and Applied Analysis (Vol. 2014). Hindawi.

Zhou, Z. H., Wu, J., & Tang, W. (2002). Ensembling neural networks: many could be better than all. Artificial intelligence, 137(1-2), 239-263.

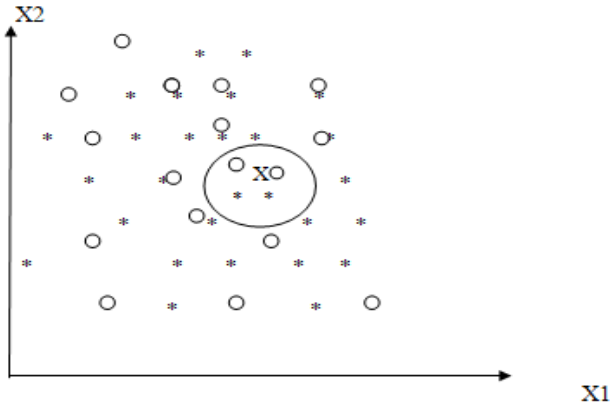

Figure 1: K - Nearest neighbor classifier

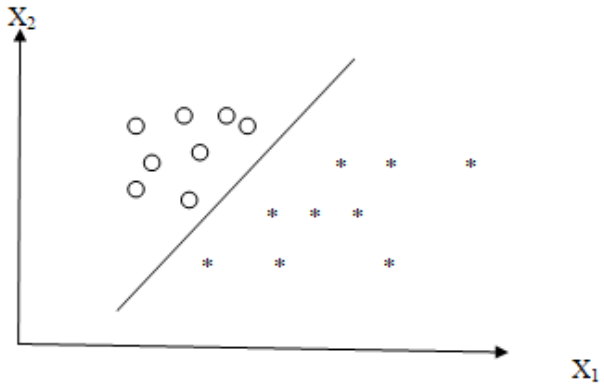

Figure 2: Architectural Support Vector Machine.

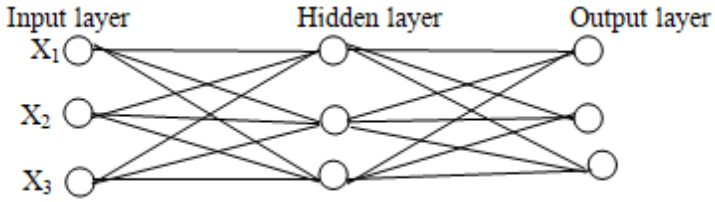

Figure 3: Architectural Graph representing Multilayer Perceptron.

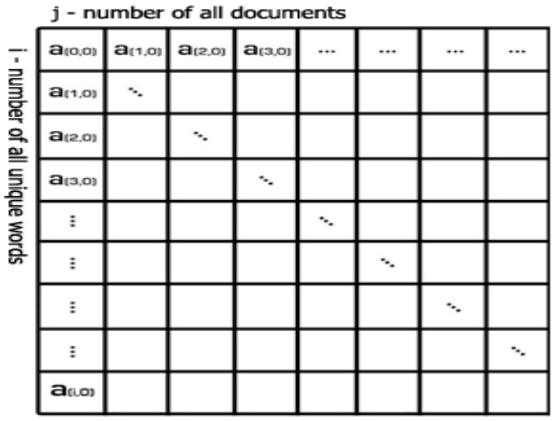

Figure 4. Weight matrix.

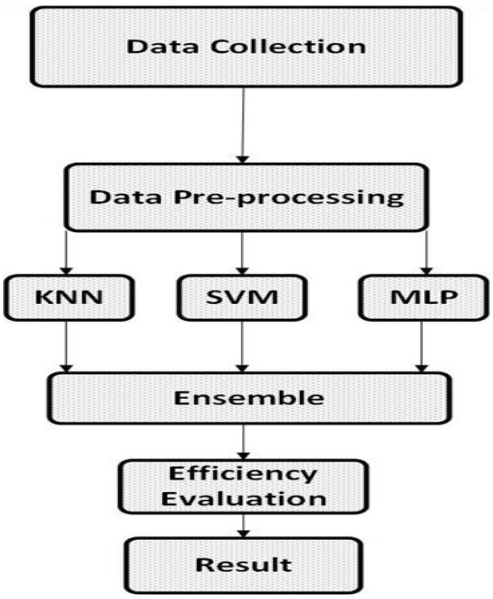

Figure 5. The flow chart of ensemble approach.

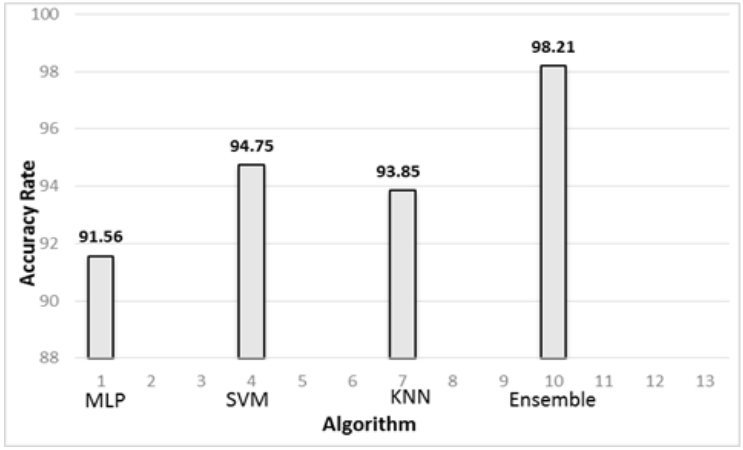

Figure 6. Compare the result of combining classifiers on the database Hamshahri.

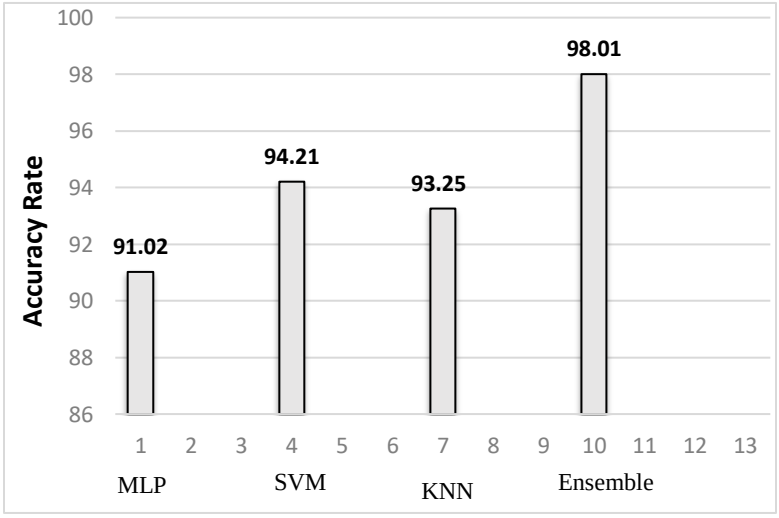

Figure 7. Compare the result of combining classifiers on the database Reuters

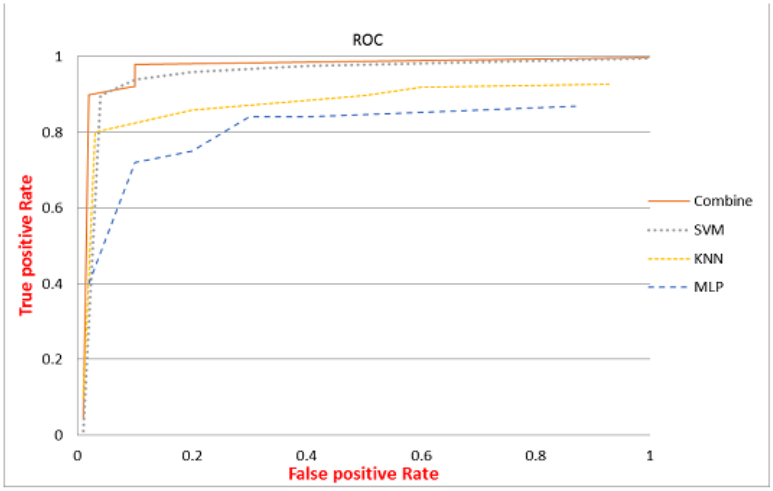

Figure 8. Curve ROC combination of algorithms SVM,KNN,MLP

Table 1: Category classification.

Category of documents	KNN Classification	SVM Classification	MLP Classification
People	90.01%	90.06	89.45%
Daily News	65.06%	71.06%	64.3%
Places	89.01%	89.03%	88.81%
Organization	86.03%	87.08%	86.01%
Political	87.02%	88.6%	87.01%
Sport	88.01%	89.02%	87.98%
Topics	82.21%	84.05%	81.89%

Table 2: The test accuracy for the second dataset (Reuters).

Approach	Accuracy (%)
MLP	91.02
SVM	94.21
KNN	93.25
The propose ensemble approach	98.01

Table 3: The test accuracy for the first dataset (Hamshahri).

Approach	Accuracy (%)
MLP	91.56
SVM	94.75
KNN	93.85
The propose ensemble approach	98.21