A Design of an Automatic Web Page Classification System

Article *in* Current Journal of Applied Science and Technology · February 2017

DOI: 10.9734/BJAST/2016/30376

CITATIONS
4

READS
540

3 authors, including:

Tarek M Mahmoud
Minia University
63 PUBLICATIONS **689** CITATIONS
SEE PROFILE

Tarek Abd El-Hafeez
Deraya University
60 PUBLICATIONS **215** CITATIONS
SEE PROFILE

British Journal of Applied Science & Technology
18(6): 1-14, 2016; Article no.BJAST.30376
ISSN: 2231-0843, NLM ID: 101664541

SCIENCEDOMAIN *international*
www.sciencedomain.org

A Design of an Automatic Web Page Classification System

Tarek M. Mahmoud[1*], Tarek Abd-El-Hafeez[1] and Doha Taha Nour El-Deen[2]

[1]Department of Computer Science, Faculty of Science, Minia University, El-Minia, Egypt.
[2]MISR University for Science and Technology, 6th of October City, Egypt.

Authors' contributions

This work was carried out in collaboration between all authors. Author TMM designed the study, performed the statistical analysis and wrote the protocol and the first draft of the manuscript. Authors TAEH and DTNED managed the analyses of the study, managed the literature searches and conducted the experimental results. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2016/30376
Editor(s):
(1) Samir Kumar Bandyopadhyay, Department of Computer Science and Engineering, University of Calcutta, India.
(2) Xu Jianhua, Department of Geography, East China Normal University, China.
(3) Sunday Olusanya Olatunji, College of Computer Science and Engineering, King Fahd University of Petroleum & Minerals, Saudi Arabia.
(4) Harry E. Ruda, Stan Meek Chair Professor in Nanotechnology, University of Toronto, Director, Centre for Advanced Nanotechnology, University of Toronto, Canada.
Reviewers:
(1) Wael A. Awad, PortSaid University, Egypt.
(2) Ufuk Çelik, Bandirma Onyedi Eylul University, Turkey.
(3) Aleksandar Karadimce, University of Information Science and Technology "St. Paul the Apostle", Ohrid, R. Macedonia.
(4) Kuldeep Singh, Guru Nanank Dev University Regional Campus, Sultanpur Lodhi, Kapurthala, Punjab, India.
Complete Peer review History: http://www.sciencedomain.org/review-history/17784

Original Research Article

Received 5th November 2016
Accepted 1st February 2017
Published 10th February 2017

ABSTRACT

Web Page Classification is one of the common problems of the today's Internet. In this paper, an automatic Web page classification system is introduced. The proposed system tries to increase the accuracy of a web page classification via combine the well-known Naïve Bayesian algorithm, Support Vector Machine and K-Nearest Neighbor. The experimental results shows that the performance of classifying web page by hybrid Naïve Bayesian classifier, Support Vector Machine and K-Nearest Neighbor algorithm is better than using Naïve Bayesian alone as always used to get the highest and fastest classifier or using K-Nearest Neighbor alone or using Support Vector Machine alone to reduce the false positive rate and get highest accuracy. The experimental results, applied on 10.000 web pages (30% for training process and 70% for testing process), showed a high efficiency with the less number of false positive rate (on average) 0%, the true positive rate (on average) 1%, F-measure (on average) 1% and overall accuracy rate (on average) 99.98%.

**Corresponding author: E-mail: d.tarek@mu.edu.eg;*

Keywords: Web page classification; naïve bayesian algorithm; support vector machine; K-nearest neighbor; support vector machine.

1. INTRODUCTION

The World Wide Web (WWW) is the immense archive of data that has been becoming exponentially throughout the years. There are more than7 billion pages accessible on the World Wide Web with more than 3.424 million users Internet [1]. Hyper Text Mark-up Language (HTML) is used in design Web pages. The volume of information available on the Web is excessive, so it is infeasible to classify Web pages manually.Web pages classification is a process of assigning the Web page to one or more predefined category labels [2]. Classification is often described as a supervised learning problem in which the set of labelled data is used as the training set to train a classifier which later is applied to new data to determine their categories. It is an important ingredient as is apparent from the popularity of Web directories such as Yahoo! [3], the Open Directory Project [4] and Looksmart [5].

The Web provides a regularly and dynamically changing environment, which makes it difficult to build a classification model that can fit to classify many different web pages. Many Web pages classification techniques have been presented to classify the Web pages. Supervised machine-learning algorithms, such as Support Vector Machine [6], Naive Bayesian Classifier [7], K-Nearest Neighbor Classifier [8] and Decision Tree Algorithm [9], are considered the most common techniques used in Web classification. This paper proposes a hybrid Web page classifier based on the classical Naive Bayesian, K-Nearest Neighbor and Support Vector Machine. The proposed technique uses Web page title and body to extract the features used in the training process.

This paper is organized as follows: Section 2 represents related works of Web page classification.

Section 3 describes HTML structure. Section 4 describes classifiers: K-Nearest Neighbor, Naive Bayesian and Support Vector Machine. Section 5 explains the proposed classification technique; Section 6 discusses the results of using the proposed classification technique. Section 7 represents a conclusion and future works.

2. RELATED WORK

This paper identifies with automatic Web page classification. Many experiments have been done to enhance the efficiency of classification. For example, Materna [10] tested the performance of four classification algorithms with several term clustering, term representations and feature selection methods for Web page classification. The best (highest) Predictive performance among the examined algorithms was achieved with the Mixture of support vector machine classifier and mutual information based feature selection.

Zhang et al. [11] utilized fuzzy K-nearest neighbor classifier for Web document classification. In order to represent the dataset, a TF/IDF (term frequency/increase document frequency) measure was adopted. In addition, membership grade was used to enhance predictive performance. The experimental results indicated that the fuzzy K-nearest neighbor classifier obtains better performance compared with the support vector machine and K-nearest neighbor classifier for Web document classification.

Ozel [12] developed a genetic algorithm-based method for Web page classification. This method uses both HTML tags and tags of terms as features for classification. Genetic algorithms were used to determine appropriate weight values for each feature. The experimental results indicated that, in the case of there being enough negative documents available in the training set, high predictive accuracy rates can be obtained for Web page classification.

Zhong and Zou [13] presented an ensemble classification model that combines support vector machine classifiers, principal component analysis and independent component analysis methods for Web page classification. Principal component analysis was utilized for feature reduction and independent component analysis was utilized for feature selection.

Choudhary and Raikwal [14] developed a Web page classification model that consists of Naive Bayesians and K-nearest neighbor. After pre-processing, the feature vector of a Web page was classified by the Naive Bayesians classifier and the K-nearest neighbor algorithm was

utilized to measure the similarity between documents of training and test sets.

Gunal [15] presented a hybrid feature selection scheme for text classification. The feature selection scheme consists of a filter and wrapper-based feature selection methods. The scheme examined the effectiveness of varying sizes of feature subsets, different dataset characteristics and classification algorithms. Peng and Choi [16] have proposed single-path search technique to reduce the search complexity and increases the accuracy for text classification of Web pages.

Xin-She Yang and Xingshi He [17] offered nature-encouraged metaheuristic algorithms specifically those based on swarm intelligence. It describes the basics of firefly algorithm along with a determination of contemporary publications. The discussion is optimality related to balancing exploration and exploitation, which is principal for all methods algorithms. Via comparing with intermittent search procedure, the conclusion is that methods such as firefly algorithm are better than the choicest intermittent search method. Analysization of algorithms and their implications for greater-dimensional optimization problems is finished.

Pikakshi et. al. [18] proposed an utterly new dimension toward internet page classification utilising artificial Neural Networks (ANN). World Wide Web is growing at an uncontrollable fee. Enormous quantities of hundreds of web sites appear day-to-day with the delivered task of keeping the online directories up-to-date. The uncontrolled nature of Web presents difficulties for Web Page classification because the number of web users is growing, so there's a want for classification of web sites with bigger precision with a purpose to present the customers with web sites of their desired category. However, web page classification has been finished in most cases via utilizing textual categorization ways. They proposed a novel process for Web Page classification that uses the HTML knowledge to complete classification.

Basem et. al. [19] presented that Intelligent Water Drops (IWD) algorithm is tailored for feature selection with Rough Set (RS). Certainly, IWD is used to search for a subset of points based on RS dependency as an analysis operate. The resulting method, referred to as IWDRSFS (Intelligent Water Drops for Rough Set Feature Selection), is evaluated with six benchmark data sets. The efficiency of IWDRSFS are analysed and when compared with those from different methods within the literature. The outcomes indicate that IWDRSFS is equipped to provide aggressive and comparable outcome. In summary, this study shows that IWD is a priceless process for method for undertaking feature selection problems with RS.

3. HTML STRUCTURE

To extract HTML structure for representation of the Web page, we can choose how a term is representative of the page considering the HTML element [20,21]. For an example, Web page represents the words of the title, words of the body and the words of meta- description, but meta- tag doesn`t find in the most of Web pages. The words presented in the TITLE element are generally more representative of the page's content than words presented in the BODY element.

We tested three different text sources for Web page representation, namely:

- TITLE (T): The page's title.
- BODY (B): The content of the body tag.
- BT: Body, Title (BT) is combined between body and title content.

4. CLASSIFIERS

The classifier is a supervised function (machine learning tool) where a learned (target) attribute is categorical. It is used after the learning process to classify new data (Web page) by giving them the best target attributes (prediction). The target attribute can be one of k class membership. We use the K-Nearest Neighbor classifier because it is particularly well suited for multimodal Classes, Naïve Bayesian Classifier because it is short computational time and Support vector machine because it is Produce very accurate classifiers and it has Memory-intensive.

K-Nearest Neighbor classifier (KNN) is based on learning by analogy [22,8]. The input consists of the k closest training examples in the feature space. The output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. The

Euclidean distance between two points $A_1 = (a_{11}, a_{12}, \dots, a_{1n})$ and $A_2 = (a_{21}, a_{22}, \dots, a_{2n})$ will be as [23]:

$$\text{dist}(A_1, A_2) = \sqrt{\sum_{i=1}^{n} (a_{1i} - a_{2i})^2} \quad (1)$$

The unknown sample is defined the most common class among its k nearest neighbor. When k=1, the unknown sample stands for the category of the training sample which is nearest to it in pattern area [8].

The Naïve Bayesian Classifier (NB) is easy, effective and simple technique used for text classification algorithm [7,24,25]. The basic method of NB is using the probabilities of words and categories to rate the probabilities of the categories given a document. Thus the NB method applies Bayesian formula:

$$P(c_j|d_i;\hat{\theta}) = \frac{P(c_j|\hat{\theta}) \prod_{k=1}^{n} P(w_k|c_j;\hat{\theta})^{N(w_k,d_j)}}{\sum_{r=1}^{|c|} (c_r|\hat{\theta}) \prod_{k=1}^{n} P(w_k|c_r;\hat{\theta})^{N(w_k,d_j)}} \quad (2)$$

Where $P(c_j|\hat{\theta})$ is computed by counting the frequency of each class c_j occurring in the training data. $|c|$ is the number of classes. $P(w_i|c_j)$ assigned to the probability that word w_i found in class c_j which may be small in the training data, so the Laplace smoothing is used to rate it. $N(w_k, d_j)$ is assigned the number of occurrences of a word w_k in d_j, n is assigned to the number of words in the training data.

The Support vector machine (SVM)is the supervised machine learning algorithm used for either classification or regression challenges [26]. It is a good and powerful learning technique presented by Weimin et al [6]. There are two methods for SVM linear classifier and non-linear classifier [27]. SVM operates by finding the best hyper surface in the space of possible inputs. This is called a linear classifier because there are many hyper-planes that might classify the data. The hyper surface tries to split the positive examples from the negative examples of finding maximize the distance between the closest of the positive and negative examples to the hyper surface [27]. Axiomatic, this makes the classification corrects for data of testing, which is closer, but not identical to the data of training.

5. THE PROPOSED TECHNIQUE FOR WEB PAGE CLASSIFICATION

The steps of the proposed classification technique consist of four phases: Training phase, Classification phase, Improving phase and Testing phase. In the Training phase, a set of the Web page documents is used to train the system. In the classification phase, we use the three algorithms to classify the Web sites. In the Improving phase, we try to improve the performance of the system via combine the three (Naive Bayesians, Support Vector Machine and K-Nearest Neighbor) algorithms. In the Testing phase, a set of experiments is conducted to evaluate the efficiency of the proposed technique. The following subsections illustrate the details of each phase;

5.1 Training Phase

To extract the features used in classification phase, a training process is done. Feature extraction means transforming arbitrary data (Web page) into features that can be taken by the classifiers. The training process requires that a set of categories has been defined, and the training Web page contents need to be somehow labelled with their respective category. The steps of the training process can be written as:

1- Insert the Web page.
2- Extract the features (words exist either in the title or the body).
3- Remove stop words and lemmization.
4- Store the features.
5- Calculate the weight (probability) of each feature.

Stop-words refer to the words which are common like "the","an","are", "a", "is", "too", "also", "to", "two","as","off", which have high frequency but these words haven`t a value as an index word. These words show high frequencies in all documents on Web, so we must remove these words at the start of indexation and then we can obtain higher speeds of calculation and fewer words needing to be indexed [28].

Lemmatization (lemmization) is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma. For example, the words "classify"," classifying", "classifiers", "classified", "classification", "classifiable" exist in documents are transformed into the word "classify" [29].

To store the extracted features a SQL Server 2008 database table is used. Each record stores the following information:

- Category name.
- Feature name.
- Number of feature's occurrences (initialized to 0).

After all the features are extracted and filling the database table with all features, we start count the elements of the table to compute the probability for each feature. To compute the probability (P_f) for a feature, the following formula is used [30]:

$$P_F = \frac{S/T_s}{K_n/T_n + S/T_s} \quad (3)$$

Where S is the number of occurrences of feature F in a certain category,T_s is the total number of certain category in the training set, n is the number of occurrences of feature F in another category, T_n is thetotal number of another category in the training set, and K is a number that can be tuned to reducefalse positives by giving a higher weight to number of occurrences of other category.

5.2 Classification Phase

In the classification phase, a set of Web pages(which are different from the Web pages used in the training step) are used, the main steps of this phase can be described as follows;

1- Parse the Web page and extract both the title and the body.
2- Extract features from the title and body.
3- For each extracted feature retrieve the corresponding probability from the database DB.
4- Extract the most interesting features from the list of features (size strictly greater than 10. i.e according to the number of frequency/occurrence grater than 10).
5- Calculate the total page probability (P) by combining the probabilities of the most interesting features.

The closer the P is for 0, the more likely the page is not to be certain category and the closer P is for 1, the more likely the page is to be certain category. In our implementation, the threshold that determines whether the Web page belongs to certain category is set to 0.9.

5.3 Improving Phase

The purpose of the improving phase is increasing the accuracy rate of the used classifiers. In our implementation, Naïve Bayesian algorithm is used to extract the features and then the support vector machine is applied in the classification process. At the end of this process, the accuracy rate of the classifier is calculated. According, some of the Web pages are classified correctly and the other are classified incorrectly. To improve the accuracy of our system, we apply the K-Nearest Neighbor algorithm for these Web pages that are classified incorrectly. We can summarize the steps of the improving phase as follows;

Step1: Insert Web Pages.
Step2: Apply Naïve Bayesian for feature selection.
Step3: Apply support vector machine to classify these Web Pages and get the results.
Step4: Determine the Web pages that classified incorrectly.
Step5: Apply K-Nearest Neighbor on the Web pages obtained step 4.

Fig. 1, presents a flowchart of the proposed system in the improving phase.

5.2 Testing Phase

In this phase, we test our proposed system by five categories (Earrings, Merger, Money, Grain and Crude Oil) and record the efficiency of our technique. The flowchart of this phase can be seen in Figure 2, for Earrings category as an example.

The steps of this phase can be summarized as follows:

1- Insert a file (F) that contains Web pages with all categories.
2- When the classifier starts, five folders are created, one file for each category.
3- If the classifier judges a given Web page, as Earrings for example, the classifier copies it from the file (F) to its file (Earrings folder).
4- If the classifier does not judge a Web page as Earrings, then the classifier compare it with the rest categories.
5- When the classifier completes the classification process, it traverses all directories to count:

True positive (**TP**): The final result is adequately classified as positive.
True negative (**TN**): The final result is adequately classified as negative.
False positive (**FP**): The final result is incorrectly classified as positive but it is negative.
False negative (**FN**): The final result is incorrectly classified as negative but it is positive.

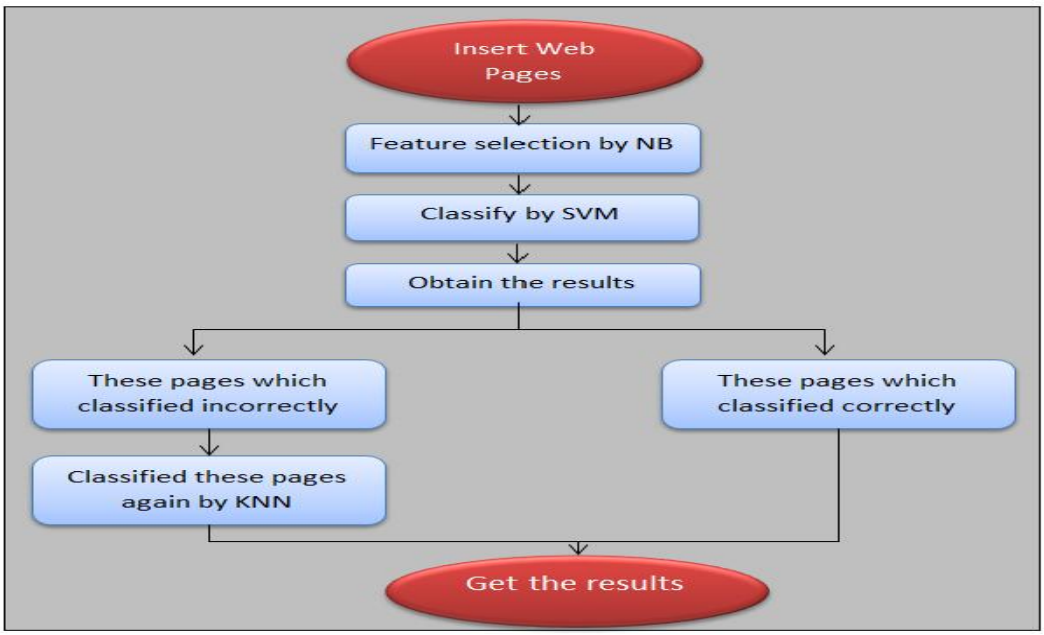

Fig. 1. Improving phase steps

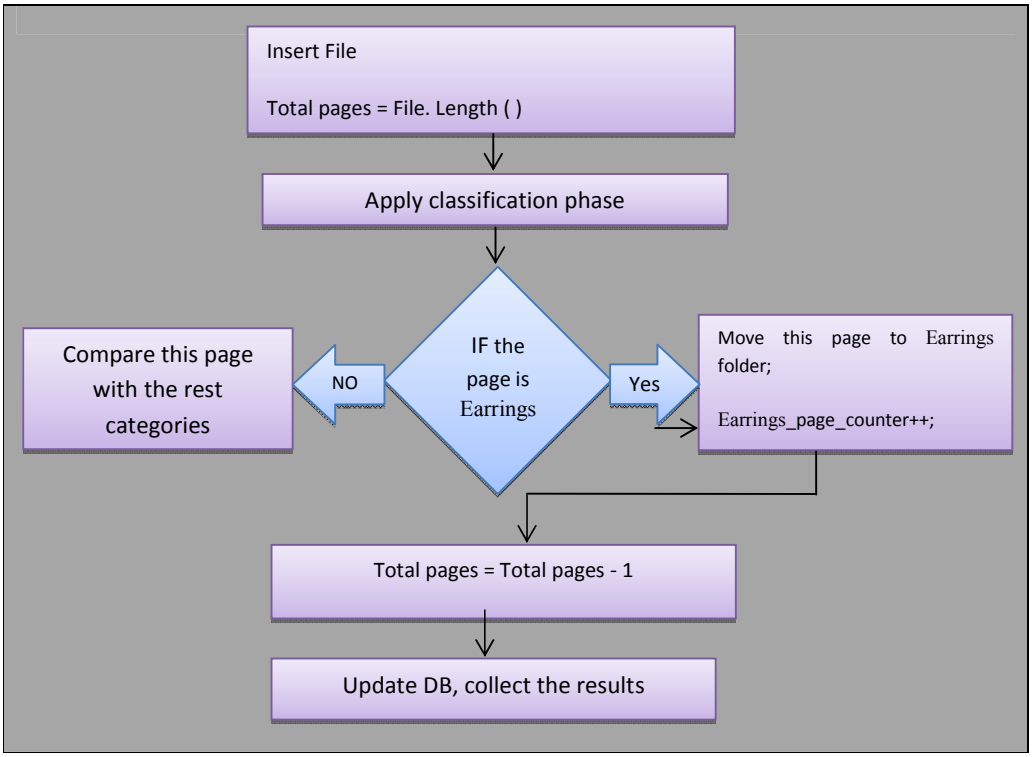

Fig. 2. Testing phase steps

There are typical measures to evaluate the efficiency of Web classification such as precision, recall, true positive rate, false positive rate, overall accuracy and F_1-measures [31]. We can compute these measures as follows [30]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$\text{True Positive (TP) Rate} = \frac{\text{TP}}{\text{TN} + \text{FP}} \quad (6)$$

$$\text{False Positive (FP) Rate} = \frac{\text{FP}}{\text{TN} + \text{FP}} \quad (7)$$

$$\text{Overall Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (8)$$

The F_1-measure is defined as equation (9):

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

6. EXPERIMENTAL RESULTS

In order to evaluate the efficiency of the classifiers discussed above, several experiments are conducted. Our experiments are done using C # programming languageon an Intel Core i5, RAM 4 GB laptop running Microsoft Windows 8 and use SQL server 2008.

6.1 Data Set

The Reuters-21578, which is considered as the standard Reuters for automatic document categorization systems [32], is used as a data set in the training and classification phases. The documents in Reuters-21578 have been collected from the Reuter's newswire in 1987.This data set consists of 21,578 Web pages which are distributed among 135 categories. We selected the categories that have largest repeated count (five category descriptions of Web pages), namely (Earrings, Merger, Money, Grain and Crude Oil) which contain 10.000 Web pages. Randomly, 30% of these Web pages are used in the training phase and the rest (70% of these Web pages) are used in the classification and testing phases.

6.2 Training Time

Fig. 3 and Table 1 illustrate the measured execution time (in second) of the training process using Naïve Bayesian, K-Nearest Neighbor and Support Vector Machine algorithms. In our implementation, the Web page title, body and both body and title (BT) are used in the training process. As can be seen in Figure 3 and Table 1, the execution time of the training process using Naïve Bayesian is smaller than both k-Nearest Neighbor and Support Vector Machine. The execution time of Support Vector Machine is greater than the K-Nearest Neighbor.

Table 1. Execution time of training process measured in seconds

Text source	Bayes	SVM	KNN
Title	0.054	18.649	1.304
Body	0.163	239.201	35.334
Body and title (BT)	0.362	466.646	71.246

6.3 Evaluation Experiments

6.3.1 Experiment 1: Testing using naïve bayesian classifier

In this section, we apply the Naive Bayesian technique (NB) using dataset selected randomly from testing data set. We have 3 test cases (title, body and, body and title (BT)). Table 2 contains the obtained true positive (TP) rate, false positive (FP) rate, precision, recall and F-measure after applying naïve Bayesian technique on each category. The Table 2 also contains the calculated overall accuracy (using equation 8) and execution time in the case of using title, body and BT.

As can be seen in Table 2, the overall accuracy of the classification process using Web Title (94.20%) is better than the classification using Web Body (93.27%) and BT (92.92%). The average F-measure values of the classification process in the case of using Title, Body and BT are 80.7%, 73.1% and 70.7% respectively.

6.3.2 Experiment 2. Testing using support vector machine classifier

In this section, the support vector machine (SVM) is used to classify the considered Web page categories.

As can be seen in Table 3, the overall accuracy of the considered classifier in the case of using Web Body (99.68%) is better than using Web title (97.22%) and BT (99.61%). The average F-measure values of the classification process in the case of using Title, Body and BT are 90.6%, 98.7% and 98.5% respectively. The experimental results of using hybrid naïve Bayesian and support vector machine classifiers are given in Table 4. In our implementation, the naïve

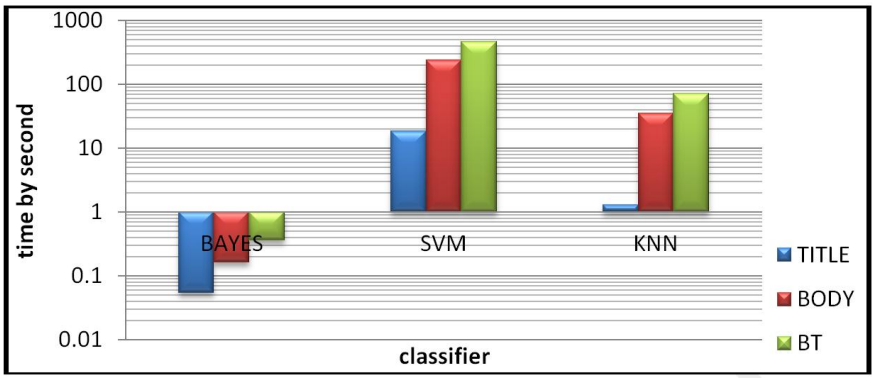

Fig. 3. The execution time of the training process

Table 2.Testing process using Naive Bayesian (NB)

		TP rate	FP rate	Precision	Recall	F-measure
Title	Earning	0.978	0.192	0.818	0.978	0.891
	Merger	0.794	0.022	0.941	0.794	0.861
	Money	0.729	0.022	0.733	0.729	0.721
	Grain	0.701	0	1	0.701	0.824
	Crude Oil	0.62	0.008	0.876	0.92	0.726
	Average	**0.765**	**0.049**	**0.874**	**0.765**	**0.807**
	Overall accuracy			**0.9420317**		
Body	Earning	0.89	0.001	0.999	0.89	0.941
	Merger	0.993	0.236	0.656	0.993	0.79
	Money	0.535	0	0.989	0.535	0.695
	Grain	0.511	0	1	0.511	0.676
	Crude Oil	0.403	0.005	0.869	0.403	0.551
	Average	**0.666**	**0.049**	**0.903**	**0.666**	**0.731**
	Overall accuracy			**0.9327386**		
BT	Earning	0.894	0.001	0.999	0.894	0.944
	Merger	0.995	0.25	0.938	0.995	0.778
	Money	0.5	0.001	0.987	0.5	0.664
	Grain	0.446	0	1	0.446	0.617
	Crude Oil	0.379	0.003	0.914	0.379	0.536
	Average	**0.643**	**0.051**	**0.908**	**0.643**	**0.707**
	Overall accuracy			**0.9291606**		

Bayesian classifier is used in the feature selection process then the support vector machine is used to classify the Web pages.

As can be seen in this Table, the overall accuracy of the considered hybrid classifier in the case of using BT (99.65%) is better than using Web Body (99.62%) and Web title (97.24%). The average F-measure of the considered hybrid classifier in the case of using title, body and BT are 90.4%, 98.5% and 98.7% respectively. Table 2 and 4 illustrate that there is an improvement in the overall accuracy and F-measure in the case of using hybrid naïve Bayesian and support vector machine compared with using naïve Bayesian alone.

6.3.3 Experiment 3. Testing using K-Nearest neighbor classifier

In this section, the K-Nearest Neighbor is used to classify the considered Web page categories.

As can be seen in Table 5, the overall accuracy of the considered classifier in the case of using Web Body (98.99%) is better than using Web title (93.61%) and BT (98.57%). The average F-measure values of the classification process in the case of using Title, Body and BT are 65%%, 77.6% and 76.9% respectively. The experimental results of using hybrid Naïve Bayesian and K-Nearest Neighbor Classifiers are given in Table 6. In our implementation, the Naïve Bayesian

classifier is used to classify the Web pages then the k-Nearest Neighbor is used to classify all Web pages that classified incorrectly again.

As can be seen in Table 6, the overall accuracy of the considered hybrid classifier in the case of using Body (99.33%) is better than using Web title (97.34%) and Web BT (99.23%). The average f-measure of the considered hybrid classifier in the case of using title, body and BT are 91.4%, 97.8% and 97.7% respectively.

Table 3. Testing process using support vector machine (SVM)

		TP rate	FP rate	Precision	Recall	F-measure
Title	Earning	0.976	0.065	0.929	0.976	0.952
	Merger	0.94	0.034	0.924	0.94	0.932
	Money	0.791	0.002	0.97	0.791	0.872
	Grain	0.872	0.003	0.961	0.872	0.914
	Crude Oil	0.817	0.007	0.911	0.817	0.861
	Average	**0.879**	**0.022**	**0.939**	**0.879**	**0.906**
	Overall accuracy			**0.9722772**		
Body	Earning	0.998	0.003	0.997	0.998	0.998
	Merger	0.988	0.004	0.991	0.988	0.99
	Money	1	0.003	0.961	1	0.98
	Grain	0.986	0	1	0.986	0.993
	Crude Oil	0.965	0.001	0.988	0.965	0.976
	Average	**0.987**	**0.002**	**0.987**	**0.987**	**0.987**
	Overall accuracy			**0.9967828**		
BT	Earning	0.999	0.006	0.993	0.999	0.996
	Merger	0.981	0.002	0.996	0.981	0.988
	Money	0.994	0.003	0.966	0.994	0.98
	Grain	0.986	0	0.993	0.986	0.989
	Crude Oil	0.972	0.002	0.972	0.972	0.972
	Average	**0.987**	**0.003**	**0.984**	**0.987**	**0.985**
	Overall accuracy			**0.9961045**		

Table 4. Testing process using hybrid naive bayesian and support vector machine (NB + SVM)

		TP rate	FP rate	Precision	Recall	F-measure
Title	Earning	0.977	0.065	0.928	0.977	0.952
	Merger	0.942	0.033	0.929	0.942	0.935
	Money	0.775	0.002	0.97	0.775	0.862
	Grain	0.867	0.002	0.959	0.867	0.911
	Crude Oil	0.824	0.008	0.901	0.824	0.861
	Average	**0.877**	**0.022**	**0.938**	**0.877**	**0.904**
	Overall accuracy			**0.972363**		
Body	Earning	0.998	0.004	0.995	0.998	0.997
	Merger	0.988	0.005	0.99	0.988	0.989
	Money	0.994	0.003	0.961	0.994	0.977
	Grain	0.986	0	0.993	0.986	0.989
	Crude Oil	0.956	0	0.994	0.956	0.975
	Average	**0.984**	**0.003**	**0.987**	**0.984**	**0.985**
	Overall accuracy			**0.996211**		
BT	Earning	0.999	0.006	0.994	0.999	0.996
	Merger	0.983	0.001	0.997	0.983	0.99
	Money	0.994	0.003	0.966	0.994	0.98
	Grain	0.986	0	0.993	0.986	0.989
	Crude Oil	0.978	0.002	0.978	0.978	0.978
	Average	**0.988**	**0.002**	**0.985**	**0.988**	**0.987**
	Overall accuracy			**0.9964586**		

Table 2, 5 and 6 illustrate that the overall accuracy and F-measure of hybrid the Naïve Bayesian classifier andK-Nearest Neighbor is better than the overall accuracy and F-measure of both the Naïve Bayesian alone and K-Nearest Neighbor alone.

6.3.4 Experiment 4. Testing using hybrid Naïve Bayesian, K-nearest neighbor and support vector machine

In this section, the experimental results of using hybrid Naïve Bayesian, K-Nearest Neighbor and

Table 5. Testing process using K-nearest neighbor (KNN)

		TP rate	FP rate	Precision	Recall	F-measure
Title	Earning	0.983	0.29	0.806	0.983	0.885
	Merger	0.684	0.023	0.928	0.684	0.787
	Money	0.647	0.01	0.833	0.647	0.729
	Grain	0.787	0.004	0.924	0.787	0.85
	Crude Oil	0	0	0	0	0
	Average	**0.62**	**0.065**	**0.698**	**0.62**	**0.65**
	Overall accuracy			**0.9361362**		
Body	Earning	0.998	0.035	0.968	0.9988	0.983
	Merger	0.957	0.005	0.989	0.957	0.973
	Money	0.97	0.004	0.953	0.97	0.962
	Grain	0.939	0.001	0.984	0.939	0.961
	Crude Oil	0	0	0	0	0
	Average	**0.773**	**0.009**	**0.779**	**0.773**	**0.776**
	Overall accuracy			**0.989906**		
BT	Earning	0.996	0.054	0.952	0.996	0.974
	Merger	0.931	0.007	0.985	0.931	0.957
	Money	0.955	0.004	0.949	0.955	0.952
	Grain	0.945	0.001	0.984	0.945	0.964
	Crude Oil	0	0	0	0	0
	Average	**0.765**	**0.013**	**0.774**	**0.765**	**0.769**
	Overall accuracy			**0.9857434**		

Table 6. Testing process using hybrid Naive Bayesians and k-nearest neighbor (NB + KNN)

		TP rate	FP rate	Precision	Recall	F-measure
Title	Earning	0.994	0.104	0.903	0.904	0.946
	Merger	0.904	0.011	0.974	0.904	0.937
	Money	0.841	0.007	0.907	0.841	0.873
	Grain	0.849	0	1	0.849	0.918
	Crude Oil	0.807	0	1	0.807	0.893
	Average	**0.879**	**0.024**	**0.957**	**0.879**	**0.914**
	Overall accuracy			**0.973359**		
Body	Earning	1	0.022	0.979	1	0.989
	Merger	0.964	0.001	0.997	0.964	0.98
	Money	0.994	0.005	0.947	0.994	0.97
	Grain	0.95	0.001	0.993	0.95	0.971
	Crude Oil	0.96	0	1	0.96	0.98
	Average	**0.974**	**0.006**	**0.983**	**0.974**	**0.978**
	Overall accuracy			**0.9933333**		
BT	Earning	1	0.03	0.971	1	0.985
	Merger	0.956	0.001	0.998	0.956	0.977
	Money	0.982	0.004	0.959	0.982	0.97
	Grain	0.961	0.001	0.992	0.961	0.976
	Crude Oil	0.958	0	1	0.958	0.978
	Average	**0.971**	**0.007**	**0.984**	**0.971**	**0.977**
	Overall accuracy			**0.9922854**		

Support Vector Machine classifiers are given in Table 7.

The experimental results of using hybrid Naïve Bayesian, Support Vector Machine and K-Nearest Neighbor classifiers are given in Table 7. In this implementation, the naïve Bayesian classifier is used in the feature selection process then the support vector machine is used to classify the Web pages. At the end of this step, the accuracy rate of the classifier is calculated. According, some of the Web pages are classified correctly and the other are classified incorrectly. The Web pages that classified incorrectly are classified again by K-Nearest Neighbor algorithm.

As can be seen in Table 7 and Table 8, the overall accuracy of the considered hybrid classifier in the case of using BT (both body and title) (99.98%) is better than using Web Body (99.94%) and Web title (98.78%). The average f-measure of the considered hybrid classifier in the case of using title, body and BT are 95.9%, 99.6% and 1% respectively. Comparing Table 7 with the rest of the tables, it is clear that there is an improvement in the overall accuracy and F-measure from using the naïve Bayesian classifier or support vector machine classifier or K-Nearest Neighbor classifier.

Finally, in Table 9, we introduce a summary for the CPU Time taken by the classification phase (the consumed time is measured by seconds).

Table 7. Testing process using hybrid Naive Bayesian, k-nearest neighbor and Support Vector Machine (NB + KNN + SVM)

		TP rate	FP rate	Precision	Recall	F-measure
Title	Earning	0.995	0.048	0.949	0.995	0.972
	Merger	0.972	0.006	0.987	0.972	0.979
	Money	0.881	0.001	0.979	0.881	0.927
	Grain	0.949	0	1	0.494	0.974
	Crude Oil	0.897	0	1	0.897	0.946
	Average	**0.939**	**0.011**	**0.983**	**0.939**	**0.959**
	Overall accuracy			**0.987695**		
Body	Earning	1	0	1	1	1
	Merger	1	0	1	1	1
	Money	1	0.002	0.982	1	0.991
	Grain	0.986	0	1	0.986	0.993
	Crude Oil	0.994	0	1	0.994	0.997
	Average	**0.996**	**0**	**0.996**	**0.996**	**0.996**
	Overall accuracy			**0.9994294**		
BT	Earning	1	0.001	0.999	1	1
	Merger	0.998	0	1	0.998	0.999
	Money	1	0	1	1	1
	Grain	1	0	1	1	1
	Crude Oil	1	0	1	1	1
	Average	**1**	**0**	**1**	**1**	**1**
	Overall accuracy			**0.9998134**		

Table 8. The overall accuracy of each technique

Technique	Overall accuracy of title	Overall accuracy of body	Overall accuracy of BT
Bayes	**0.942032**	0.93274	0.92913
SVM	0.972277	**0.99678**	0.9961
KNN	0.936136	**0.98991**	0.98574
NB & SVM	0.972363	0.99621	**0.99646**
NB & KNN	0.973359	**0.99333**	0.99229
NB & SVM& KNN	0.987695	0.99943	**0.99981**

Table 9. The CPU Time taken by the classification phase

	Execution time of title	Execution time of body	Execution time of BT
Bayes	338.25	304.62	318.54
SVM	314.65	351.69	275.28
KNN	300.1700	2184.34	2609.93
NB & SVM	281.83	342	274.32
NB & KNN	297.23	1992.84	2007.80
NB & SVM& KNN	280.31000	576.19	442.51

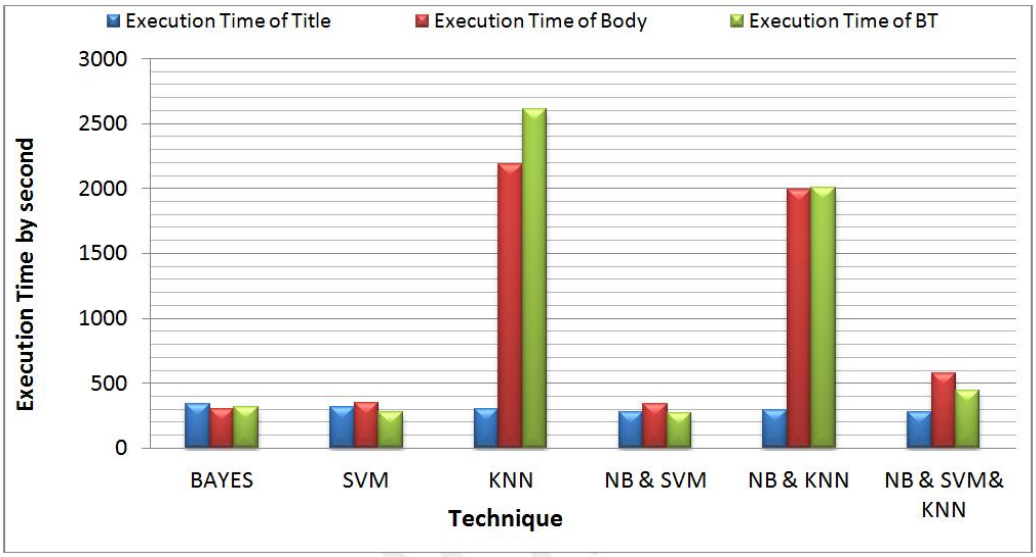

Fig. 4. A summary for the CPU Time taken by applying the different techniques

Based on Table 9 and Figure 4, the time taken (measured by second) by applying a hybrid Naïve Bayesian and Support Vector Machine, hybrid Naïve Bayesian and K- Nearest Neighbor and, hybrid Naïve Bayesian, Support Vector Machine and K- Nearest Neighbor are 274.32, 2007.80 and 442.51 respectively. Although, the time taken by the proposed hybrid technique (Naïve Bayesian, Support Vector Machine and K-Nearest Neighbor) is greater than hybrid Naïve Bayesian and Support Vector Machine and, hybrid Naïve Bayesian and K- Nearest Neighbor. But we recommend using our system because the performance accuracy rate is 99.98% and false positive rate is zero.

6. CONCLUSION

In this paper, an automatic Web page classification system is introduced. The proposed system tries to increase the accuracy of a web page classification via combine the well-known Naïve Bayesian algorithm, Support Vector Machine and K-Nearest Neighbor. The experimental results shows that the performance of classifying web page by hybrid Naïve Bayesian classifier, Support Vector Machine and K-Nearest Neighbor algorithm is better than using Naïve Bayesian alone as always used to get the highest and fastest classifier or using K-Nearest Neighbor alone or using Support Vector Machine alone to reduce the false positive rate and get highest accuracy. The experimental results, applied on 10.000 web pages (30% for training process and 70% for testing process), showed a high efficiency with the less number of false positive rate (on average) 0%, the true positive rate (on average) 1%, F-measure (on average) 1% and overall accuracy rate (on average) 99.98%. There are still many issues to be considered in the future work. Some instances of these issues are: using parallel programming to decrease the time taken, using unsupervised machine learning, using other machine learning methods such as neural network compare with our work, also we will study how we use our system to classify images or videos or audios.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Available:http://www.internetlivestats.com/internet-users/
(Last visited 1 may 2016).
2. Pratiksha Y Pawar, Gawande SH. A comparative study on different types of approaches to text categorization. International Journal of Machine Learning and Computing. 2012;2(4):423-426.
3. Yahoo!.
Available:http://www.yahoo.com/
(last visited 1 June 2016).
4. Looksmart.
Available:http://www.looksmart.com/
(last visited 15June 2016).
5. Open Directory Project.
Available:athttp://www.dmoz.org/
(last visited 1 July 2016).
6. Weimin Xue, Hong Bao, Weitong Huan, Yuchang Lu. Web page classification based on SVM, 6th World Congress on Intelligent Control and Automation, Dalian, China. 2006;6111-6114.
7. Fan Yan, Zheng C, Wang QY, Cai QS, Liu Jo. Web page classification based on naive bayes method, (In Chinese). Journal of Software. 2001;1386-1392.
8. Shengyi Jiang, Guansong Pang, Meiling Wu, LiminKuang, An improved K-nearest neighbor algorithm for text categorization Elsevier Ltd. 2011;39:1503–1509.
9. Pant B, Pant K, Pardasani KR, Decision tree classifier for classification of plant and animal micro RNA's", computational intelligence and intelligent systems. Springer Berlin Heidelberg. 2009;443–451.
Available:http://link.springer.com/chapter/10.1007/978-3-642-04962-0_51
10. Materna J. Automated web page classification, In: Proceedings of recent advances in Slavonic natural language processing, Masaryk, Czech Republic. Masaryk: University Press. 2008:84–93.
11. Zhang J, Niu Y, Nie H, Web document classification based on fuzzy k-nn algorithm, In: Proceedings of international conference on computational intelligence and security, Beijing. New York: IEEE. 2009;193–196.
12. Ozel SA. A web page classification system based on a genetic algorithm using tagged-terms as features, Expert Systems with Applications. 2011;28:3407–3415.
13. Zhong S, Zou D. Web page classification using ensemble of support vector machine classifiers. Journal of Networks. 2011;76: 1625–1630.
14. Choudhary R, Raikwal J. An ensemble approach to enhance performance of webpage classification. International Journal of Computer Science and Information Technologies. 2014;5:5614–5619.
15. Gunal S. Hybrid feature selection for text classification. Turkish Journal of Electrical Engineering and Computer Sciences. 2012;20:1296–1311.
16. Xiaogang Peng, Choi B, Automatic web page classification in a dynamic and hierarchical way, data Mining, 2002. ICDM 2003. Proceedings. IEEE International Conference. 2002;386-393.
DOI: 10.1109/ICDM.2002.1183930
17. Xin-She Yang, Xingshi He. Firefly algorithm: Recent advances and applications. Int. J. Swarm Intelligence. 2013;1(1).
18. Pikakshi Manchanda, Sonali Gupta, Komal Kumar Bhatia. On the automated classification of web pages using artificial neural network. IOSR Journal of Computer Engineering (IOSRJCE). 2012;4(1):20-25.
19. Basem O. Alijla, Lim Chee Peng, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar. Intelligent water drops algorithm for rough set feature selection. Springer Verlag Berlin Heidelberg. 2013;356–365.
20. Johnwiley and sons, HTML and CSS: Design and Build Websites manufactured in the united states of America; 2011.
21. The World Wide Web Consortium (W3C) HTML 4.01 Specification.
Available:http://www.w3.org/TR/html4/
(Last visited 1 June 2016).
22. Fang Lu Qingyuan Bai. A refined weighted K-nearest neighbors algorithm for text categorization, Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference; 2010.
23. Abdo Y Alfakih, Amir Khandani, Henry Wolkowicz. Solving euclidean distance matrix completion problems via semidefinite programming, Computational Optimiziation and Applications. 1999;12: 13–30.
24. Vidhya KA, Aghila GA. survey of naïve bayes machine learning approach in text document classification, (IJCSIS)

International Journal of Computer Science and Information Security. 2010;7.
25. McCallum A, Nigam KA comparison of event models for naive bayes text classification, In AAAI-98 Workshop on Learning for Text Categorization; 1998.
26. Joachims T. Text categorization with support vector machines: Learning with many relevant features. In proceedings of ECML-98, 10th European Conference on Machine Learning (Chemnitz, DE). 1998:137–142.
27. Hanif Mohaddes Deylami, Yashwant Prasad Singh. Cybercrime detection techniques based on support vector machines. Artificial Intelligence Research. 2013;2:1-12.
28. Xiaoguang Qi, Brian D. Davison. Web page classification: Features and algorithms. ACM Comput.Survey. 2009;41.
29. Christopher D. Manning, Prabhakar Ragha vanand Hinrich Schütze. Introduction to Information Retrieval. Cambridge University Press. 2008;32-35.
30. Tarek M. Mahmoud, Alaa Ismail El-Nashar, Tarek Abd-El-Hafeez, Marwa Khairy Mohamed. An efficient three-phase email spam filtering technique. British Journal of Mathematics & Computer Science. 2014; 4(9):1184-1201.
31. Powers David MW. Evaluation: From precision, recall and F-measure to Roc, informedness, markedness & Correlation. Journal of Machine Learning Technologies, Bioinfo Publications. 2011;2:37-63.
32. Lewis DD. Reuters-21578 document corpus; 1.0. Available:http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html (Last visited 11, 2016).

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/17784