www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 6 June 2017, Page No. 21854-21860
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i6.51

An efficient method for web page classification based on text

Dr.Tamer Anwar Ahmed Alzohairy, Assistant Professor of Computer Science, Faculty of Science ,Al-Azhar University, Dr. Mohamed Taha Abu-Kresha, Lecture of Computer Science, Faculty of Science ,Al-Azhar University and Ahmed Nagy Ramadan Bakry, Teaching Assistant of Computer Science, Faculty of Science ,Al-Azhar University

***Abstract*—According to Google index , the number of web pages now exceeds 50 billion, and is increasing by millions per day. The global population of internet users is also growing rapidly and then a web page classification problem arises. According to that, automatic classification method is required to deal with this problem of the World Wide Web (WWW). The traditional methods that use text determine the class of the document, but usually retrieve unrelated web pages. In order to effectively classify web pages, we apply different feature extraction techniques with different web page classification methods to find an efficient method for web page classification. The three feature extraction methods used in the study are Term Occurrence (TO), Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF) and the three classifiers used in the study are K-nearest neighbor (K-NN), Naive Bayes (NB) and Decision Tree (DT). Each web page is represented by the three feature extraction methods. The principal component analysis (PCA) is used to select the most relevant features for the classification as the number of unique words in the collection set is big. The final output of the PCA is sent to the three different classifiers to find the best method for web page classification. The experimental evaluation used demonstrates that the combination of Naive Bayes (NB) and Term Frequency-Inverse Document Frequency provides an efficient classification accuracy compared to other methods**

Keywords*—Decision Tree (DT), K-nearest neighbor (KNN), Naive Bayes (NB), Principal component analysis (PCA), Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), Term Occurrence (TO)

I. Introduction

According to Google index[10], web pages number now Exceeds 50 billion, and is increasing. The global population of internet users is also growing rapidly. To find specified web page, many search engines are available to users, including Google[11], Yahoo[12], and Bing[13]. Typical search engines work through keyword inputs. However, pages retrieved in this manner usually include invalid links and irrelevant web pages. A good web page classification method is thus an urgent need in facilitating user searches and increasing percentage of related web pages.

There are many classification methods for web pages [1]. By following given advantage and disadvantage of some of them.

A decision tree Apte[3] is a general data classification method. It has two major advantages which are: 1) It's fast. 2) The classification result can be transformed into an IF-THEN relation that the user can easily understand.

Mccallum and Nigam [4] transform the frequency of keywords to condition probabilities in which Bayesian probability is used to calculate the probability value between every document and category. Under this system, the category with highest probability is the one the document belongs to. The advantage is that the correlation between two documents can be represented by a probability.

k-nearest neighbor method by Tan[5] is often used in text document classification. Woog and Lee[6] use a k-NN approach to calculate the likelihood of a category and relevant web page. In order to improve performance, they add a feature selection, HTML tags, a new similarity measure and evaluation. Selamat and Omatu[7] use a training sample to do the stemming process and remove stop words, then the feature vector dimensions for a portion are reduced, while another portion is used for each category extraction of the keyword and to assign the weight value. The two types of feature vectors are then combined and inputted to a neural network for training. The system can then classify the web pages into the desired categorization. Unfortunately, a long time is required for training, and the convergence speed is low.

In this paper, we apply different feature extraction techniques with different web page classification methods to find an efficient method for web page classification. The three feature extraction methods used in the study are Term Occurrence, Term Frequency, Term Frequency-Inverse Document Frequency and the three classifiers used in the study are K-nearest neighbor (KNN), Naive Bayes (NB) and Decision Tree (DT). Each web page is represented by the three feature extraction methods. The principal component analysis (PCA) is used to select the most relevant features for the classification as the number of unique words in the collection set is big. Then the final output of the PCA is sent to the three different classification methods to find the best method for web page classification.

The remainder of the paper is organized as follows: In section II feature extraction methods used are illustrated. In section III the feature selection method is described.

classification methods are given in Section IV. Experimental results and dataset are given in Section V. and conclusions are given in Section VI.

II. Features extraction

In this paper three diffrent types of features extraction methods are used in study which are: Term Occurrence (TO), Term Frequency (TF) and Term Frequency Inverse Document Frequency (TF-IDF). Following gives a description for each type.

A. term occurrence

In document, term occurrence is number of times term k occurred in document j, it can be represented by n(term k, doc j).

Example 1:

Consider the document given by "I Love Egypt, Egypt is my home" then n(Egypt,doc 1)=2.

B. term frequency

Term frequency measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (the total number of terms in the document) as a way of normalization. n(term k,doc j)/n(term all/doc j) where term all means all terms.

Example 2:

Consider the document given by "I Love Egypt, Egypt is my home" then TF=2/7.

C. Term frequency - Inverse Document frequency

Inverse document frequency measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:
IDF(term) = log (Total number of documents/Number of documents with term k in it).

Example 3:

Consider we have 1000 documents and the word Egypt appears in three hundred of these . Then, the inverse document frequency (i.e., IDF) is calculated as log(1000/300) = 0.5228 .
Term frequency Inverse Document frequency is combination between TF and IDF which calculated by the equation: TF-IDF = TF(term k, Doc j) * IDF(term k, doc j).

Example 4:

Consider the same documents given in example 3 and let one document of them containing 100 words and the word Egypt in it appears 3 times. The term frequency (TF) for Egypt is given by (3/100) = 0.03, then TF-IDF = 0.03 * 0.5228.

III. Feature Selection

Feature selection methods are used to reduce the extracted data being processed to save time, computations and Noise removal during classification process. In this paper, principle component analysis (PCA) method is used for feature selection.

Principal component analysis (PCA)

The advantages of using PCA method are reduce the dimensionality of a data set by finding a new set of variables smaller than the original set of variables, retains most of the sample's information and help in classification of data. Following give details about PCA method.

Feature reduction using the PCA

Suppose we have matrix A which contains the term weights obtained by feature extraction techniques:

$$A = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1k} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2k} & \cdots & X_{2m} \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ X_{n1} & X_{11} & \cdots & X_{nk} & \cdots & X_{nm} \end{pmatrix}$$

Where x_{jk} (j=1,2,....,n; k=1,2,....,m) is the terms weight that exists in the collection of documents. The definitions of j=1, . . . , n. and k=1,. . . . . , m. n is the number of documents to be classified and m is the number of term weights obtained from feature extraction. Because number of terms obtained from each document are diffrent, the missed row terms is completed from the end by null values.
The steps used by PCA in order to reduce the dimensionality of matrix A are as follows:

1) Calculate the mean of m variables in matrix A:

$$\overline{X_k} = \frac{1}{n} \sum_{j=1}^{n} x_{jk}$$

2) Calculate the variance S_{kk}^2 of m variables in matrix A:

$$S_{kk}^2 = \frac{1}{n} \sum_{j=1}^{n} (x_{jk} - \overline{X_k})^2$$

3) Calculate the covariance S_{ik} of m variables in matrix A:

$$S_{ik} = \frac{1}{n}\sum_{j=1}^{n}(x_{ji} - \overline{X_i})(x_{jk} - \overline{X_k})$$

where i = 1, . . . ,m. Then we determine the eigenvalues and eigenvectors of the covariance matrix S which is a real symmetric positive matrix. An eigenvalue λ and the eigenvector e can be found such that, Se =λe.

In order to find the eigenvector e the characteristic equation |S $-\lambda$I|= 0 must be solved. If S is an m × m matrix of full rank, m eigenvalues (λ_1, λ_2, ..., λ_m) can be found. By using (S $-\lambda_i$I)e_i= 0, all corresponding eigenvectors can be found. The eigenvalues and corresponding eigenvectors will be sorted so that λ_1>= λ_2>= λ_3>=....>= λ_m.

Let a square matrix E be constructed from the eigenvector columns where E=[$e_1 e_2 e_3 ... e_m$].

Also let us denote Λ as

$$\Lambda = \begin{pmatrix} \lambda 1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda 2 & 0 & \dots & 0 \\ 0 & 0 & \lambda 3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda \mathrm{m} \end{pmatrix}$$

In order to get the principal components of matrix S, we will perform eigenvalue decomposition which is given by E^TSE=Λ

Then we select the first d≤m Eigen vectors where d is the desired value ,e.g., 100, 200, 400, etc. The set of principal components is represented as Y_1=$e_1^T s$, $Y_2 = e_1^T s$,. . . , $Y_d = e_1^T s$.

An n×d matrix M is represented as

$$\mathbf{M} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & \cdots & f_{1d} \\ f_{21} & f_{22} & f_{23} & \cdots & f_{2d} \\ f_{31} & f_{32} & f_{33} & \cdots & f_{3d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & f_{n3} & \cdots & f_{nd} \end{pmatrix}$$

where f_{ij} is a reduced feature vectors from the n×m original data size to n×d Size.

Example 5:

Suppose we have the following data set:

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 5 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 7 \end{pmatrix}, \begin{pmatrix} 7 \\ 8 \end{pmatrix}$$

The mean is given by:

$$\overline{X_k} = \begin{pmatrix} 4.5 \\ 5 \end{pmatrix}$$

Then we calculate (x_{jk} - $\overline{X_k}$)

$$\begin{pmatrix} -2.5 \\ -4 \end{pmatrix}, \begin{pmatrix} -1.5 \\ 0 \end{pmatrix}, \begin{pmatrix} -0.5 \\ -2 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}, \begin{pmatrix} 1.5 \\ 2 \end{pmatrix}, \begin{pmatrix} 2.5 \\ 3 \end{pmatrix}$$

To get the covariance we need to calculate $(x_{ji} - \overline{X_i})(x_{jk} - \overline{X_k})$ as followos:

$$\begin{pmatrix} -2.5 \\ -4 \end{pmatrix} \begin{pmatrix} -2.5 & -4 \end{pmatrix} = \begin{pmatrix} 6.25 & 10 \\ 10 & 16 \end{pmatrix},$$

$$\begin{pmatrix} -1.5 \\ 0 \end{pmatrix} \begin{pmatrix} -1.5 & 0 \end{pmatrix} = \begin{pmatrix} 2.25 & 0 \\ 0 & 0 \end{pmatrix}, \ldots$$

by the end we calculate the covariance matrix by the equation

$$S_{ik} = \frac{1}{n}\sum_{j=1}^{n}(x_{ji} - \overline{X_i})(x_{jk} - \overline{X_k})$$

$$S_{ik} = \frac{1}{6}\begin{pmatrix} 6.25 & 10 \\ 10 & 16 \end{pmatrix} + \begin{pmatrix} 2.25 & 0 \\ 0 & 0 \end{pmatrix} + \ldots\ldots$$

$$\text{Covariance matrix} = \frac{1}{6}\begin{pmatrix} 17.5 & 22 \\ 22 & 34 \end{pmatrix} = \begin{pmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{pmatrix}$$

By solving the equation |S $-\lambda$I|= 0 we can find the eigen values.

$$\begin{pmatrix} 2.92 - \lambda & 3.67 \\ 3.67 & 5.67 - \lambda \end{pmatrix} = 0$$

λ_1 = 8.21

λ_2 = -8.21

Considering d=1, $\lambda_1 > \lambda_2$, we find its own vector:

λ_1 =8.21

$$\text{A-}\lambda_1\text{*E} = \begin{pmatrix} 2.92 - 8.21 & 3.67 \\ 3.67 & 5.67 - 8.21 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0$$

By solving this equation we get:

$$x_1 = 1.44 * x_2$$

We have reduced features to only 1×1 instead of 1×2.

IV. Classification methods

In this paper we are using three classification methods which are Naive Bayes, Dicision Tree and K-nearest neighbor.

A. Naive Bayes(NB)

NB model are popular in machine learning applications, due to their simplicity in allowing each attribute to contribute towards the final decision equally and independently from the other attributes [2]. This simplicity equates to computational efficiency, which makes NB techniques attractive and suitable for many domains. NB is a straightforward and frequently used method for supervised learning. Performs surprisingly well in a very wide variety of problems inspite of the simplistic nature of the model.

Naive Bayes mechanism:

Consider D=$\{d_1, d_2, d_3, \ldots d_n\}$ to be a set of documents and C=$\{c_1, c_2, c_3, \ldots . c_q\}$ be set of classes. Each of the n number of documents in D are classified into one of the q number classes from set C. We classify D as the class which has the highest posterior probability P(C|D). The probability of a document d being in class c using Bayes theorem is given by:

$$P(C|D) = \frac{P(D|C)P(C)}{P(D)} \tag{1}$$

To calculate $P(D|C)$ we calculate $P(w_k|C)$ for each word,

$$P(w_k|C) = \frac{n_k + 1}{n + |vocabulary|} \tag{2}$$

where n is the number of words in class and n_k= number of times word k occurs in class and $|vocabulary|$ number of unique words for all documents.

Example 6:

Consider we have 5 documents shown in table I.

TABLE I

DOC	TEXT	CLASS
1	I Loved the movie	+
2	I hated the movie	-
3	A great movie. Good movie	+
4	Poor acting	-
5	Great acting.a good movie	+

The unique words in the five documents are: I, Loved, the, movie, hated, a, great, poor, acting, good.

The term occurrence for each word in each document is computed in table II:

TABLE II

DOC	I	Loved	The	Movie	Hated	A	Great	Poor	Acting	good	Class
1	1	1	1	1							+
2	1		1	1	1						-
3				2		1	1			1	+
4								1	1		-
5				1		1	1			1	+

To classify new document d_{test} = "I hated the poor acting", assume that the document is belong to class (+) then P(+)=$\frac{3}{5}$, $P(I|+) = \frac{1+1}{14+10}$, $P(hated|+) = \frac{0+1}{14+10}$,

$P(the|+) = \frac{1+1}{14+10}$, $P(poor|+) = \frac{0+1}{14+10}$, $P(acting|+) = \frac{1+1}{14+10}$

$$P(+|d_{test}) = P(+)P(I|+)P(hated|+)P(the|+)$$
$$P(poor|+)P(acting|+) = 6.03 * 10^{-7} \tag{3}$$

Next we assume that the document is belong to class (-) then

$$P(-|d_{test}) = P(-)P(I|-)P(hated|-)P(the|-)$$
$$P(poor|-)P(acting|-) = 1.22 * 10^{-5} \tag{4}$$

By comparing the two results, we classify d_{test} as the class which has the highest posterior probability P(C|D) which is class (-).

B. Decision Tree (DT)

Unlike NB classifier, DT classifier can cope with combinations of terms and can produce impressive results for some domains. However, training a DT classifier is quite complex and they can get out of hand with the number of nodes created in some cases. Decision trees may be computationally expensive for certain domains, however, they make up for it by offering a genuine simplicity of interpreting models, and helping to consider the most important factors in a dataset first by placing them at the top of the tree.

Decision Tree mechanism works in two steps:

1) Tree construction:
   a) At start, all the training examples are at the root.
   b) Partition examples recursively based on selected attributes.
2) Tree pruning
   a) Test the attribute values of the sample against the decision tree.

Decision Tree Algorithm (greedy algorithm):

1) Tree is constructed in a top-down recursive divide-and-conquer manner
2) At start, all the training examples are at the root
3) Attributes are categorical (if continuous-valued, they are discretized in advance)
4) Samples are partitioned recursively based on selected attributes

Example 7:

Consider the documents given in example 6. We construct our tree shown in Fig. 1

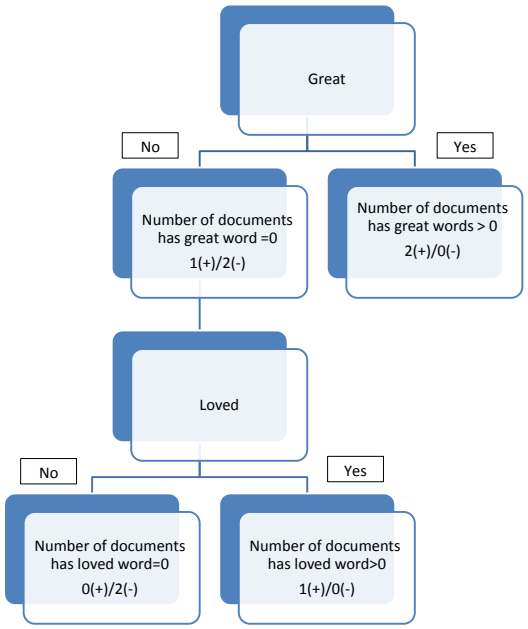

Fig. 1: Decision Tree

Consider the test example is given by: "I hated the poor acting". Because number of great words =0 then go left, number of loved words=0 then go left. Then the class of the example is negative.

C. K-nearest neighbor (K-NN)

In K-NN classifier, 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.

Algorithm:

1) Compute Distance D(x, x_j) to every training example x_j.
2) Select k closest instances $x_{i1}, x_{i2}, \ldots\ldots, x_{ik}$ and their labels $y_{i1}, y_{i2}, \ldots., y_{ik}$.
3) Output the class y* which is most frequent in $y_{i1}, y_{i2}, \ldots., y_{ik}$.

Where distance is calculated using Euclidian distance equation:

$$D(x, x_j) = \sqrt{\sum_d (x - x_j)^2} \tag{5}$$

Example 8:

Consider the documents given in example 6 which can be represented as given by table III:

TABLE III

DOC	I	Loved	The	Movie	Hated	A	Great	Poor	Acting	good	Class
1	1	1	1	1							+
2	1		1	1	1						-
3				2		1	1			1	+
4								1	1		-
5				1		1	1			1	+
6	1		1		1			1	1		

Consider the test document is given by: "I hated the poor acting". The calculations used by K-NN are as follows:

$$D(<1,1,1,1,0,0,0,0,0,0>,<1,0,1,0,1,0,0,1,1,0>) = \sqrt{|1-1|^2+|1-0|^2+|1-1|^2+|1-0|^2+|0-1|^2+|0-0|^2+|0-0|^2+|0-1|^2+|0-1|^2+|0-0|^2} \tag{6}$$

$$= \sqrt{5}$$

$$D(<1,0,1,1,1,0,0,0,0,0>,<1,0,1,0,1,0,0,1,1,0>) = \sqrt{|1-1|^2+|0-0|^2+|1-1|^2+|1-0|^2+|1-1|^2+|0-0|^2+|0-0|^2+|0-1|^2+|0-1|^2+|0-0|^2} \tag{7}$$

$$= \sqrt{3}$$

$$D(<0,0,0,2,0,1,1,0,0,1>,<1,0,1,0,1,0,0,1,1,0>) = \sqrt{|0-1|^2+|0-0|^2+|0-1|^2+|2-0|^2+|0-1|^2+|1-0|^2+|1-0|^2+|0-1|^2+|0-1|^2+|1-0|^2} \tag{8}$$

$$= \sqrt{12}$$

$$D(<0,0,0,0,0,0,0,1,1,0>,<1,0,1,0,1,0,0,1,1,0>) = \sqrt{|0-1|^2+|0-0|^2+|0-1|^2+|0-0|^2+|0-1|^2+|0-0|^2+|0-0|^2+|1-1|^2+|1-1|^2+|0-0|^2} \tag{9}$$

$$= \sqrt{3}$$

$$D(0,0,0,1,0,1,1,0,0,1>,<1,0,1,0,1,0,0,1,1,0>) = \sqrt{|0-1|^2+|0-0|^2+|0-1|^2+|1-0|^2+|0-1|^2+|1-0|^2+|1-0|^2+|0-1|^2+|0-1|^2+|1-0|^2} \tag{10}$$

$$= \sqrt{9} = 3$$

If we take k=2, Then the closest two documents are, document 2 which class is (-) and document 4 which class is (-). So the majority is (-) so we classify the test document as (-)

V. Experiment

In this section, experiments to test and evaluate the best method to classify web page using text are given. The experements are done using the three feature extraction methods Term Occurrence, Term Frequency and Term Frequency-Inverse Document Frequency for each one of the three classifiers used which are K-nearest neighbor (KNN), Naive Bayes (NB) and Decision Tree (DT).

A. Data set

The WebKB[8] collection is used for our classification. Stopped words are removed from each web page then stemmer algorithm is applied to reduce each word to it's word stem root form. The dataset contains the web pages related to each of the four categories used for classification. As the categories are student home page, faculty, course, and project. Number of web pages in each category, 1096 in student category, 336 in project category, 746 in faculty category and 619 in course category. The total web pages used for classification is 2798.

B. EXPERIMENTAL RESULTS

Data mining tool WEKA[9] is used to perform the classification. The classification is done with 2798 examples. The classifiers performances has been analyzed and compared by the measures Precision, Recall and F-measure, which are obtained from the confusion matrix shown in table IV as follows:

TABLE IV

	Category 1	Category 2
Classified as 1	True Positive	False Positive
Classified as 2	False Negative	True Negative

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (11)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (12)$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (13)$$

Precision measure the relevant documents found against all documents found i.e. the percentage of selected documents that are correct. Recall measure the relevant documents found against all relevant documents i.e. the percentage of correct documents that are selected. F-measure is weighted harmonic mean between precision and recall. Experimental result are showen in table V, figure 2, 3 and 4.

TABLE V: Accurecy table

	Precision	Recall	F-Measure
Bayes(TO)	0.817	0.817	0.815
Bayes(TF)	0.817	0.817	0.815
Bayes(IDF-TF)	0.83	0.83	0.829
KNN(TO)	0.754	0.735	0.716
KNN(TF)	0.754	0.735	0.716
KNN(IDF-TF)	0.784	0.75	0.733
DT(TO)	0.648	0.647	0.634
DT(TF)	0.648	0.647	0.634
DT(IDF-TF)	0.67	0.671	0.657

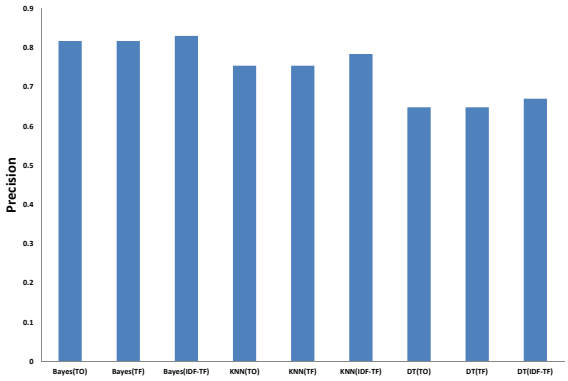

Fig. 2: Precision of the classifiers

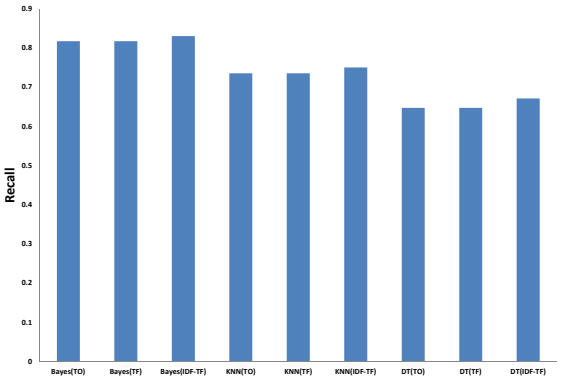

Fig. 3: Recall of the classifiers

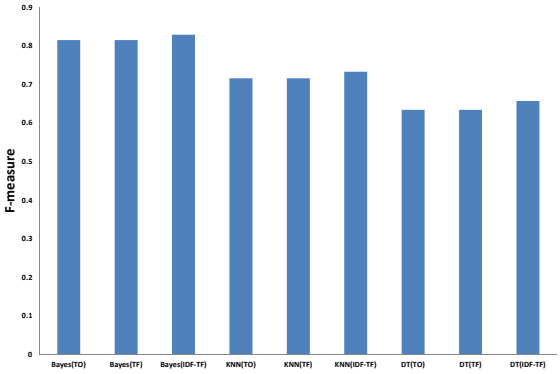

Fig. 4: F-measure of the classifiers

From the three classifiers used, Naive Bayes showed the best performance in precision, recall, f-measure. Feature extraction method has also impacted the performance of the classifiers, Term frequency and term occurrence had the same results with different classifiers but term frequency-Inverse document frequency made an improvement to all classifiers used.

VI. Conclusion

In this paper comparative study has been done to show the best method to deal with web page classification problem. The web page classification is based on web page text to determine the class of each web page. Text is feeded into feature extraction method to generate features, then PCA algorithm has been used as feature selection to reduce the dimensionality of the features, save time, reduce computations and Noise removal. A comparison study among nine methods have been done based on three feature extraction methods Term Occurrence, Term Frequency and Term Frequency-Inverse Document Frequency for each one of the three classifiers used which are K-nearest neighbor (KNN), Naive Bayes (NB) and Decision Tree (DT). Results show that Naive Bayes is the best classification method compared to other classification methods. Term Frequency-Inverse Document Frequency made an improvement to all classifiers used. Naive Bayes classification method with Term Frequency-Inverse Document Frequency is proposed to handle this domain of problems. In future research we will Improve the accuracy of the proposed classifier. Studying web page classification problem with different features like photos, links, etc. Applying the proposed methods in other applications and domains.

References

[1] QI, Xiaoguang; DAVISON, Brian D. Web page classification: Features and algorithms. ACM computing surveys (CSUR), 2009, 41.2: 12.

[2] PATIL, Ajay S.; PAWAR, B. V. Automated classification of web sites using Naive Bayesian algorithm. In: Proceedings of the international multiconference of engineers and computer scientists. 2012. p. 14-16.

[3] APTE, Chid, et al. Text mining with decision rules and decision trees. IBM Thomas J. Watson Research Division, 1998.

[4] MCCALLUM, Andrew, et al. A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. 1998. p. 41-48.

[5] TAN, Songbo. Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Systems with Applications, 2005, 28.4: 667-671.

[6] KWON, Oh-Woog; LEE, Jong-Hyeok. Text categorization based on k-nearest neighbor approach for web site classification. Information Processing & Management, 2003, 39.1: 25-44.

[7] SELAMAT, Ali; OMATU, Sigeru. Web page feature selection and classification using neural networks. Information Sciences, 2004, 158: 69-88.

[8] The 4 Universities data set [Online]. Available: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/, Accessed June 2012

[9] FRANK, Eibe, et al. Weka. Data Mining and Knowledge Discovery Handbook, 2005, 1305-1314.

[10] http://www.worldwidewebsize.com

[11] https://www.google.com

[12] https://www.yahoo.com

[13] https://www.bing.com