ISSN:0975-9646

An Ensemble Approach to Enhance Performance of Webpage Classification

Roshani Choudhary[1], Jagdish Raikwal [2]

[1, 2] *Dept. of Information Technology*
[1, 2] *Institute of Engineering & Technology*
[1, 2] *DAVV Indore, India*

Abstract **- Automatic web page classification is a complex and slow process. Additionally the application of classification need more accurate and efficient methods, due to increasing demand of web page categorization. Therefore, the proposed work is focuses on the web page classification and clustering schemes and obtaining an enhanced classification technique. In order to obtain efficient text mining techniques various machine learning algorithms are studied and two classification techniques namely Bayesian classification and KNN classification techniques are found for efficient and accurate results. Using both the algorithms a new hybrid technique is developed which is able to perform training on the domain specific data and successfully able to classify the web page according to the available domain knowledge.**

Keywords- **web page classification, categorization, content mining, text analysis.**

I. INTRODUCTION

With the advancement of Internet technologies, the number of Internet users is increasing and the amount of information online is also increasing. As browsing information or files on the Internet has become one of important channels for knowledge acquisition, how to effectively manage Internet information/files to assist the users in efficiently absorbing and utilizing required information has become an important issue. That is, if the webpage that contains these Internet information/files can be effectively classified, it would enhance
user convenience and increase webpage browsing rate to derive their required information. Data mining is a technique through which we can get useful information from a large amount of data. When we apply data mining techniques on web it is called web mining.

A. Web Mining

Web mining is the application of data mining techniques to discover patterns from the Web. According to analysis targets, web mining can be divided into three different types[1], which are described below.

1) Web Usage Mining: Web usage mining is the process of extracting useful information from server logs e.g. Web usage mining is the process of finding out what users are looking for on the Internet.

2) Web structure mining: Web structure mining is the process of using graph theory to analyze the node and connection structure of a web site.

3) Web content mining: Web Content Mining is the process of extracting useful information from the contents of Web documents. Content data corresponds to the collection of facts a Web page was designed to convey to the users. It may consist of text, images, audio, video, or structured records such as lists and tables.

Research activities in this field also involve using techniques from other disciplines such as Information Retrieval (IR) and natural language processing (NLP). Webpage classification is an application area of web content mining.

B. Webpage Classification

Web page classification, also known as web page categorization, is the process of assigning a web page to one or more predefined category labels.

According to [2], Web page classification can be used in multiple types of classification problems such as

1) Subject classification- Subject classification is concerned about the subject or topic of a web page. For example, judging whether a page is about "arts", "business" or "sports" is an instance of subject classification.

2) Functional classification- Functional classification cares about the role that the web page plays. For example, deciding a page to be a "personal homepage", "course page" or "admission page" is an instance of functional classification.

3) Sentiment classification- Sentiment classification focuses on the opinion that is presented in a web page, i.e., the author's attitude about some particular topic.

And many other type of classification such as genre classification, search engine spam classification and so on. This work focuses on subject classification.

II. BACKGROUND

This section briefly reviews early work on classification of web pages and related topics. In the early days, classification was done manually by domain experts. But when the growth of internet increased, classification was also carried out in semiautomatic or automatic manner. Many approaches for text classification and web page classification are proposed. Some of these approaches uses statistical and machine leaning techniques [1] like k-

Nearest Neighbor approach [11], Bayesian probabilistic models [7] [9], Fuzzy Similarity Based Models [10] etc.

Some other papers used specialized methods, *[3]* proposes several Web-page summarization algorithms for extracting the most relevant features from Web pages for improving the accuracy of Web page classification. In [12] a method is proposed to improve the Web-page classification performance by removing the noise through summarization techniques. In *[4]* webpage design characteristics for webpage classification are explored. That is, concerning complexity of webpage structure, this paper analyzes the webpage design characteristics including tag attributes and tag-region layout to develop an algorithm for webpage classification.

In [6], the concepts and techniques of data mining, a promising and flourishing frontier in database systems and new database applications is explored. In [9] the focuses is on the Web Page Classification based on combination of both the content and structure of the web page. A survey in *[2]* review the web-specific features and algorithms that have been explored and found to be useful for web page classification. The contributions of this survey are:

- A detailed review of useful web-specific features for classification.
- An enumeration of the major applications for web classification.
- A discussion of future research directions.

In *[5]* study of automatic classification of Web documents into pre-specified categories, with the objective of increasing the precision of Web search is provided. They propose an Automatic Classifier, with a set of categories and a pre-classified training set of pages. According to [11], Classification is a model finding process that is used for portioning the data into different classes according to some constrains. In other words we can say that classification is process of generalizing the data according to different instances. Several major kinds of classification algorithms including C4.5, k-nearest neighbor classifier, Naive Bayes, SVM, Apriori, and AdaBoost. This paper provide a inclusive survey of different classification algorithms.

A. Algorithm Study

In this section the algorithms that are used for the web page classification is provided.

1) K-nearest-neighbor (KNN) algorithm: The K-nearest-neighbor (KNN) algorithm measures the distance between a query scenario and a set of scenarios in the data set. We can compute the distance between two scenarios using some distance function d(x, y), where x, y are scenarios composed of features, such that [9]

X={x1, x2, x3…..}

Y= {y1, y2, y3…..}

Two distance functions are discussed here:

Absolute distance measuring:

$$\mathbf{d_A(x,y)} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{|x_i - y_i|}$$

Euclidean distance measuring:

$$\mathbf{d_A(x,y)} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \sqrt{\mathbf{x_i^2 - y_i^2}}$$

KNN can be run in these steps:

1. Store the output values of the M nearest neighbors to query scenario Q in vector

r = {r_1…...r_m} by repeating the following loop M times:

a. Go to the next scenario S_i in the data set, where I is the current iteration within the domain {1……P}

b. If Q is not set or q < d (q, S_i): q ←d (q, S_i), t ←O_i

c. Loop until we reach the end of the data set.

d. Store q into vector c and t into vector r.

2. Calculate the arithmetic mean output across r as follows:

$$\bar{r} = \frac{1}{M}\sum_{i=i}^{M} r_i$$

3. Return r as the output value for the query scenario q

2) Bayesian Classifier

The Naive Bayes classification algorithmic rule is a probabilistic classifier. It is based on probability models that incorporate robust independence assumptions. The independence assumptions usually don't have an effect on reality. So they're thought of as naive. You can derive probability models by using Bayes theorem (proposed by Thomas Bayes). Based on the nature of the probability model, you'll train the Naive Bayes algorithm program in a very supervised learning setting. In straightforward terms, a naive Bayes classifier assumes that the value of a specific feature is unrelated to the presence or absence of the other feature, given the category variable. There are two types of probability as follows:

- Posterior Probability [P (H/X)]
- Prior Probability [P (H)]

Where, X is data tuple and H is some hypothesis. According to Baye's Theorem

$$P\left(\frac{H}{X}\right) = \frac{P\left(\frac{X}{H}\right)P(H)}{P(X)}$$

The below given pseudo code helps to understand the bay's algorithm

```
MyData
{       String DOMAIN;
```

```
    HashSet<Word,Frequency>  Words;
}

 Function Predict_NaivB (String  txt_Test)
1) HashSet  <MyData>  WordsWeight  =
   Load_TrainData();

2) TOKENS = Extract_Tokens_From ( txt_Test );

3) HashSet <Domain, Probability> Prob_TestData  =
     getProbabilty ( TOKENS, WordsWeight);
4) HashSet <Domain, Probability> PriorProb_Domain  =
     getPriorProbabilityOfDomain(WordsWeight);

5) HashSet<Domain, Probability> PriorProb_TestData  =
   getPriorProbabilityOfTestData(Prob_TestData,
   PriorProbDomain);

6) THREAH=getTreashHoldAVG(PriorProb_TestData);

7) MAX_PROB=GetMax(PriorProb_TestData);

8) IF  (PriorProbTestData[MAX_PROB]  .value  >
   THRESH)
      {
      RETURN
   PriorProbTestData[MAX_PROB].DOMAIN;
       }

9) ELSE
       {
     RETURN     GetKNNPrediction(     WordsWeight,
  TOKENS)
       }
```

III. PROPOSED WORK

A. Problem Domain

Web page classification is a complex domain of data classification due their uneven format and page structure. The following issues are targeted to resolve in this study.

1) Automatic classification of web pages may not much accurate. Additionally, efficiency is also an issue for automatic classification.

2) Pre-processing of web page is another critical issue to
extract the useful content of a Web page.

3) HTML, UML or webpage tags may be used for webpage classification. So using these tags according to the requirement is another issue.

4) In web page classification, one page can belong to multiple categories or domains. So another issue is to classify the web page in appropriate category.

This section describes the issues and challenges that intended to study in addition of that the appropriate solution is also listed in the next section.

B. Proposed Solution

The proposed solution is aim to resolve the issues listed in above section, therefore the following solution is suggested to implement for the accurate and efficient classification scheme.

*1) Implement a dynamic pre-processing system***:** In this phase a dynamic pre-processing scheme is implemented to remove tags, similes, special characters and other words those are not appropriate for domain knowledge development.

*2) Implement Bayesian classification scheme***:** Bayesian classifier estimates the word frequency for estimating the probability distribution for a specified domain and according to the probability distribution the class labels of the document is predicted.

*3) Improve the performance of classifier using ensemble learning techniques***: S**ometimes classifiers are not performed well because of weak learning. Therefore for improving the classification a strong learner is employed with the weak learner to enhance the performance of classifier.

*4) Performance analysis of proposed technique***:** For measuring the effectiveness of the proposed data model the performance of the system is performed using different experimentation.

This section describes the general steps of the solution development and the next step demonstrates the proposed system architecture and their components.

IV. PROPOSED ARCHITECTURE

The proposed system architecture is given in figure 1, this system contain different sub-components. The given system is divided into two major parts first training and seconds testing of the system. In the training module we train the classifier using the domain keywords. The second module is the test module, in this module we find the domain of a test web page. To find the domain of a unknown web page we applied two algorithm in our proposed system, naïve Bayesian algorithm and KNN algorithm.

Step by step description of the architecture given below

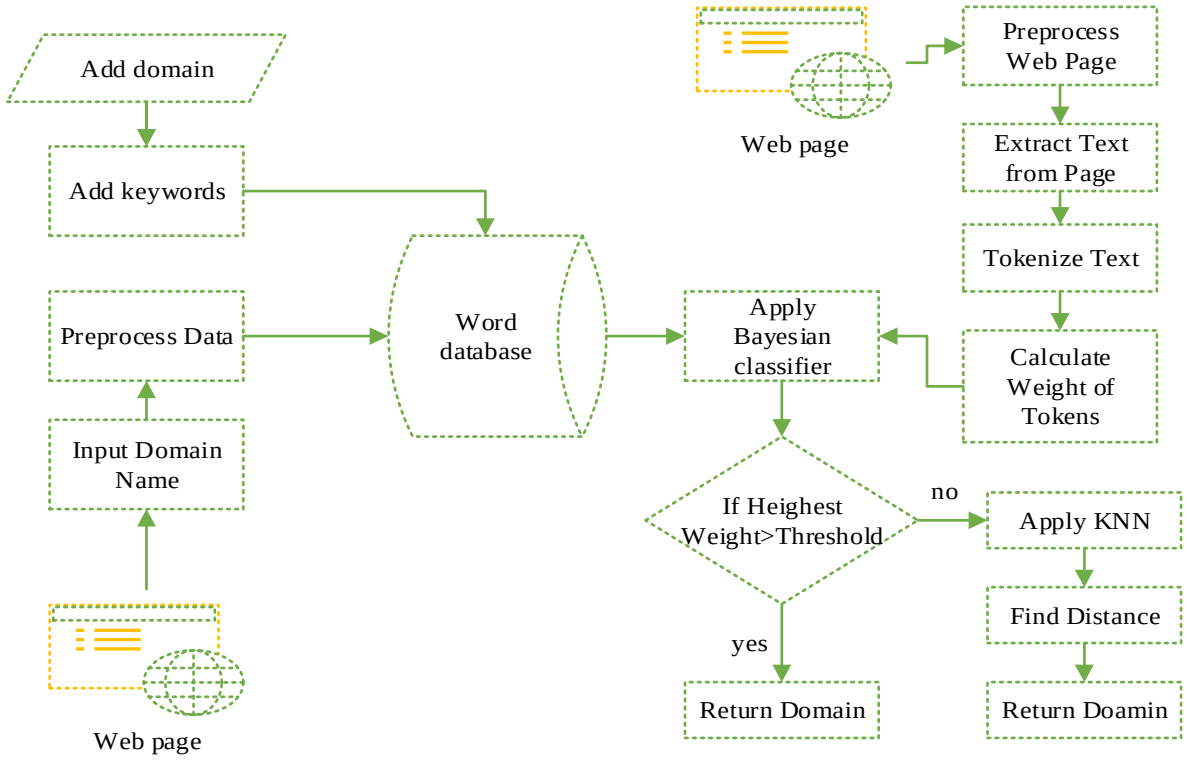

Figure. 1 Architectural model

First we train our classifier, there are two ways for this.

a. Manual training
   - In manual training we manually add the domain keywords. For this we first select the domain name and then add keyword and their weight to that domain.
   - These keywords with their weights are added in the database.

b. Automatic training
   - In automatic training first we chose the domain name.
   - Then we upload some file for that domain.
   - Then the pre-processing system pre-process that file, and extract the domain keyword and their frequency from the file.
   - The extracted keywords with their frequency are added in the database.

c. Now after training we test the system, in testing to extract the keywords from a webpage we follow the steps described below:
   - Upload the web page using upload function.
   - Then the web page is pre-processed to extract the text content. In this step the HTML tags, scripts, styles are removed from the web page and it is saved as a text document.
   - Then the text is tokenized .
   - After tokenizing the text, the stop words are removed.
   - After removing the stop word, the word weight is fined using the word frequency of the test page, and the word weight of the training domain keywords.

d. Now using the keywords the domain of the test web page is find, for this first we apply naive Bayesian classification algorithm to find the probability of each domain.

e. If the highest probability is greater than the threshold value. Then the domain with highest probability is predicted as the domain of test page.

f. If the highest probability is less than the threshold, then the KNN is used to find the distance between the test page and each domain.

g. The domain with smallest distance is predicted as the domain of the test page.

In previous section we discussed the different steps in the classification system. Now we briefly discuss the algorithms used in our system for classification.

V. RESULTS ANALYSIS

This section provides the results and performance analysis of the proposed web page classification system. That includes the description of parameters and obtained results during various experimentations. The given results in this section are compared with the result of the previous system. In the previous system[7] naïve Bayesian algorithm is used for classification purpose, and in the proposed system we also used Naïve Bayesian algorithm. In addition we also used K-nearest neighbor algorithm to enhance the accuracy of the classifier. So we can say that is this section we compared the results of naïve Bayesian algorithm with, naïve Bayesian algorithm plus k-nearest neighbor algorithm used together.

A. *Accuracy*

The accuracy is a measurement how accurately the classifier classifies the given samples after performing the training. We use recall, precision and F-measure to verify the accuracy of our classification approach [7]. F-measure is the harmonic mean of recall and precision. Recall, Precision and F-Measure are calculated as follows:

- Recall = $\frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$
- Precision = $\frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}$
- F-measure = $\frac{(2 \times \text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})}$

The classifier was subjected to training and testing in 9 steps each time increasing the input by 50 documents. In table 1 the comparison of the old system[7] and the proposed system is given in terms of f- measure. First we take 50 documents for training the classifier and after training check the accuracy of classifier in terms of f-measure. Than we increase the training documents to 100 documents and again calculate the f- measure. We repeat this process 9 times, until the number of training documents reaches to 450. In the previously proposed system in [7], the results with the same approach are given. So we compared the results of both systems which is given in Table 1.

Table 1 – Comparison of f-measure for old system and the proposed system

Number of training Web Pages	Old f-measure in % (Using naïve Bayesian algo)	New f-measure in % (Using proposed system)
50	48	71
100	58	74
150	65	79
200	70	81
250	74	83
300	79	88
350	82	90
400	87	91
450	88	92

The obtained results are given in the figure 2. In this diagram the X axis presents the number of documents for training and Y axis provides the accuracy in terms of percentage f-measure. According to the given results the performance of classifier improved as the size of data for training increases. In addition of that the given graph provides the comparative results between old system[7] and proposed system. The obtained results demonstrate the proposed system provides much accurate results as compared to old system. It also demonstrate that the proposed system gives much better results when trained with small number of training documents.

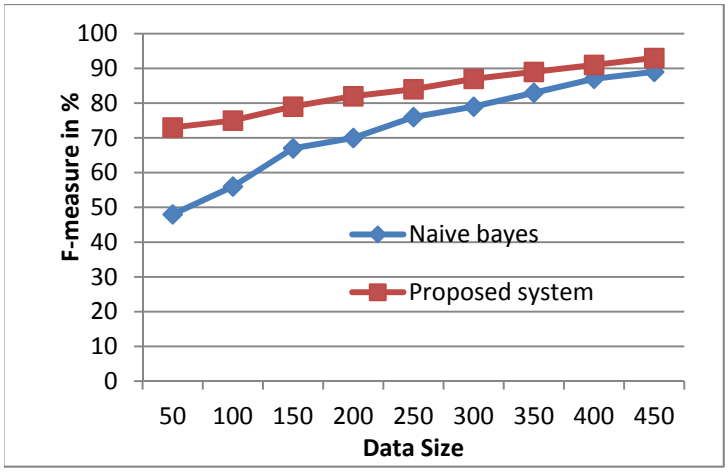

Figure 2. Comparison of accuracy for old system and proposed system

VI. CONCLUTION AND FUTURE WORK

A. *Conclusion*

Various tools and techniques are recently developed for web page classification. But most of them are not much accurate for classifying web pages. Therefore a new concept for web page classification is proposed in the presented work.

The proposed model includes the Bayesian classification scheme and k nearest neighbor classification scheme. In order to classify the web pages first web page data is extracted. Then pre-processing is used for refining the data extracted from web pages. The refined contents are used for creating content features. This feature vector of web page is used for classification using the Bayesian classifier. Additionally for improving the classification accuracy k nearest neighbor algorithm is also applied for similarity measurement between training documents and test document.

The presented work is implemented using visual studio dot net environment, and ASP dot net is used for web application development. After implementation of the desired system the performance of the system is adoptable and providing efficient results.

The proposed data model for classification of web pages is adoptable and efficient due to their obtained performances. The next section provides the future extension on the implemented data model.

B. *Future Work*

Web page classification is an essential contribution of web mining. Various information and essential contents are available in these data formats. Therefore the proposed system provides the efficient results for mining the web pages. In near future the presented work is improved for their efficiency in terms of accuracy of classification and automated web page classification. In addition of that the presented work is limited for the HTML 4.0 that can be extended to HTML 5 version.

In the proposed system only the text content is used as features, in future we can also use other data from web pages such as images, hyperlinks, audio, video, HTML tags etc as features. In the presented work we did not used the dependencies between keyword. We considered all keywords as independent form each other. In future we can also add dependencies between keywords which can increase the efficiency of the system.

REFERENCES

[1] S. Gowri Shanthi, Dr. Antony Selvadoss Thanamani, *Enhanced Approach on Web Page Classification Using Machine Learning Technique*, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 7, September 2012

[2] Xiaoguang Qi and Brian D. Davison, *Web Page Classification: Features and Algorithms*, Department of Computer Science & Engineering, Lehigh University, June 2007

[3] Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zeng, Benyu Zhang, Yuchang Lu, Wei-Ying Ma, *Web-Page Classification Through Summarization*, Sheffield, South Yorkshire, UK, SIGIR'04, July 25–29, 2004

[4] Shih-Ting Yang, *A Webpage Classification Algorithm Concerning Webpage Design Characteristics*, International Journal of Electronic Business Management, Vol. 10, No. 1, pp. 73-83 (2012)

[5] Chandra Chekuri Michael H. Goldwasser, *Web Search Using Automatic Classification*, Computer Science Department, Stanford University

[6] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques, Second Edition*, University of lllinois at urbana-champaing 2006

[7] Ajay S. Patil, B.V. Pawar, *Automated Classification of Web Sites using Naive Bayesian Algorithm*, IMECS vol-1, 2012

[8] Ms. Darshna Navadiay, Mr. Mehul Parikh, Ms. Roshni Patel, *Constructure Based Web Page Classification*, International Journal of Computer Science and Management Research, Vol 2 Issue 6 June 2013 ISSN 2278-733X

[9] Victor Fresno, Raquel Martinez, Soto Montalvo, Arantza Casillas, *Naive Bayes Web Page Classification with HTML Mark-Up Enrichment*

[10] Shalini Puri and Sona Kaushik, *A Technical Study And Analysis On Fuzzy Similarity Based Models For Text Classification,*, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.2, No.2, March 2012

[11] Raj Kumar, Dr. Rajesh Verma, *Classification Algorithms for Data Mining: A Survey*, International Journal of Innovations in Engineering and Technology (IJIET), Vol. 1 Issue 2 August 2012

[12] Dou Shena, Qiang Yang a, Zheng Chen, *Noise reduction through summarization for Web-page classification*, Information Processing and management 43 (2007) 1735–1747