International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 7 Issue 5 (June 2013), PP.87-94

A Generic Approach for Web Page Classification Using URL's Features Along With the Textual Content

D.V.N.Siva Kumar, Sabyasachi Patra,
Dept of Computer Science, I.I.I.T Bhubaneswar, Odisha, India.
Dept of Computer Science, I.I.I.T Bhubaneswar, Odisha, India.

Abstract:- Classification of web pages greatly helps in making the search engines more efficient by providing the relevant results to the user's queries. In most of the prevailing algorithms available in literature, the classification/ categorization solely depends on the features extracted from the text content of the web pages. But as the most of the web pages nowadays are predominately filled with images and contain less text information which may even be false and erroneous, classifying those web pages with the information present alone in those web pages often leads to mis-classification. To solve this problem, in this paper an algorithm has been proposed for automatically categorizing the web pages with the less text content based on the features extracted from both the URLs present in web page along with its own web page text content. Experiments on the bench marking data set "WebKB" using K-NN, SVM and Naive Bayes machine learning algorithms shows the effectiveness of the proposed approach achieving higher accuracy in predicting the category of the testing web pages. Our proposed algorithm achieves higher accuracy 90% when K-NN is employed on the given data set. There is also considerable improvement in accuracy using other two algorithms when employed for classifying the web pages based on our proposed approach.

Keywords: - Vector Space, Textual Content, URL Features, Cosine Similarity, Machine Learning.

I. INTRODUCTION

There is an exponential increase in the amount of data available on the web recently. According to [1], the number of web pages available on the web is around 1 billion with almost another 1.5 million are being added daily. This enormous amount of data in addition to the interactive and content-rich nature of web has made it popular. However, these web pages vary to a great extent in both the information content and quality. Moreover, the organization of these web pages do not allow for easy search. So an efficient and accurate method for classifying the web pages is essential for the search engines in order to provide the relevant results from the already categorized documents to the user's query. Web Page Categorization is an important ingredient as is evident from the popularity of web directories such Yahoo [2], Looksmart [3], Open Directory Project[4]. Classification of web page content is essential to many information retrieval tasks such as constructing and maintaining the web directories, improving the quality of the search results, Building efficient focused crawlers, Helping Question Answering Systems [5]. However, these resources created by large teams of human editors and represent only on kind of classification task that, while widely useful, can never suitable to all applications. Web page classification techniques use concepts from many fields like Information filtering and retrieval, Artificial Intelligence, Machine learning, Text mining and so on. Information filtering and retrieval techniques usually build either a thesauri or indices by analysing a corpus of already classified texts with specific algorithms. Vector representation of the corpus text may be used instead of building thesauri and indices. Natural language techniques also may be used for the classification of the web pages. Many researchers propose the use of text-mining techniques to do web mining/web page classification. As the HTML pages are semi-structured documents containing tags, frames, images and some unwanted information etc, some pre-processing is required to be done in the web pages before applying text mining techniques. However, applying text mining techniques for web page classification has the major drawback as it does not utilize the contextual features like URL's, Structure, Meta, Title Tags, Anchored Text, Tables, Frames, visual layout of HTML pages, which are very much useful for web page classification.

Web page classification needs a lot of preprocessing work because of the presence of hyperlinks and large number of HTML tags. It is estimated that 80% of the preprocessing is needed before the classification of web pages. Feature extraction or selection is one of the most important step in pattern recognition or pattern classification, data mining, machine learning and so on. It is also an effective dimensionality reduction technique and an essential preprocessing method to remove noise features. The basic idea of feature selection algorithms is searching through all possible combinations of features in the data to find which subset of features works best for prediction. The selection is done by reducing the number of features of the feature vectors, keeping the most meaningful discriminating ones, and removing the irrelevant or redundant ones. On one hand, feature increased gives difficulties to calculate, because the more data occupy amount of memory space and computerization time, on the other hand, a lot of features include certainly many correlation factors respectively, which results to information repeat and waste. Therefore, it is necessary to take measures to decrease the feature dimension under not decreasing accuracy; this is called the problems of feature optimum extraction or selection. The characteristics of good features should be simple, moderate, less redundancy and unambiguous.

Search engines needed an automatic categorization of web pages for the following reasons in order to provide relevant results to the user's query. i) Large amount of information available in the internet makes it difficult for the human experts to manually classify them. ii) The amount of expertise needed is high for manual classification. iii) Web pages are dynamic and volatile in nature. iv) More time and effort are required for classification v) Same type of classification scheme may not be applied to all the web pages.

The rest of the paper is organized as follows. Section 2 focuses on the Literature Survey. Section 3 presents our proposed algorithm for classification of any web pages irrespective enough content is present in it or not. Section 4 focuses on Experimental results performed on benchmarking data set WebKb. Section 5 presents the Conclusion and Future Work. Section 6 describes the References.

II. RELATED WORK

In this section, we primarily aim to investigate empirical methods that have been used to extract the relevant features from web pages for classification.

In[6], the authors have used feature selection methods Symmetrical Uncertainty [7] and ReliefF [8] to find the subset of words which help to discriminate between different kinds of web pages. In their work they also used Hidden Naives Bayes, Complement class Naïve Bayes to overcome the problem of high dimensional text vocabulary space. Their results have shown Symmetric Uncertainty is more competitive than RelierF in selecting the relevant words in web pages for classification. In [9], a time efficient and improved accuracy novel approach for classification of web pages is proposed using Mean Filed Independent Component Analysis (MFICA), which is based on Newton's iteration method to improve the FastICA algorithm. In their approach, they first represent the web page as a vector of features with different weights. Then they used Independent Component Analysis [10] and Principal component analysis (PAC) to select the independent features and to reduce the multidimensional data sets to lower dimensional data set respectively. Finally, the reduced independent data set is applied on Naïve Bayes classifier and shown that classifier's accuracy is improved. In [11], a Link Information Categorization algorithm which is the improvement of the K-Nearest Neighbour algorithm, this algorithm determines the category attribute of the current web page through the links which other pages point to the current web page. They have shown that their results have got higher recall and F1 measures when classifying web pages of portal sites. In [12], they have shown that introduction of hyperlink elements of web page can improve the classification accuracy of web pages based on mutual information. In Their work, they used the hyperlink factor which is based on the weight assigned to hyperlink sections based on their contributions to the classification. Thus they improved the accuracy of the classification by including the hyperlink factor in mutual information.

How classification accuracy can be increased by combining linked based and content based methods for classification [13]. They used four different measures of subject similarity derived from link structure, Provided an important insight on which measure derived from links are more appropriate to compare web documents and how these measures can be combined with content based methods to improve the classification accuracy. Several other works in the literature have reported the successful use of links as a means to improve classification performance. Using the taxonomy presented in Sun et al. [14], we can summarize these efforts in three main approaches: hypertext, link analysis, and neighbourhood.

In the hypertext approach, Web pages are represented by context features, such as terms extracted from linked pages, anchor text describing the links, paragraphs surrounding the links, and the headlines that structurally precede them. Furnkranz et al. [15], Glover et al. [16] and Sun et al. [14] achieved good results by using anchor text, and the paragraphs and headlines that surround the links, whereas Yang et al. [17] show that the use of terms from linked documents works better when neighbouring documents are all in the same class.

In the link analysis approach, learning algorithms are applied to handle both the text components of the Web pages and the linkage among them. Slattery and Craven [18], for instance, explore the hyperlink topology using a HITS based algorithm [19] to discover test set regularities. Fisher and Everson [20] shown that link information is useful when the document collection has a sufficiently high link density and the links are of high quality.

Finally, in the neighbourhood approach, the document category is estimated based on category assignments of already classified neighbouring pages. The algorithm proposed by Chakrabarti et al. [21] uses the known classes of training documents to estimate the class of the neighbouring test documents. Their work shows that co-citation based strategies are better than those using immediate neighbours. Oh et al. [22] improved on this work by using a filtering process to further refine the set of linked documents to be used. The authors in [23] have shown that the hyperlink patterns and meta data can be helpful in achieving optimal performance of the classification. They found that using words in web page alone would yield often sub-optimal performance of the classifiers. However, they also observed that if the information such as Meta data, hyperlink patterns from neighbouring pages is not used in careful manner, therefore, it will be more harmful than helpful in classification due to the noise in linked neighbourhood pages. In [24], they improved the classification accuracy by using novel method that eliminates a noisy neighboured links from a web page. They used a feature similarity measure with respect to all links and text of web page that can distinguish related hyperlinks in a web page from the noisy hyperlinks. Finally, they classify the web page by concatenating the text of all non noisy link information with the text present in web page to be classified.

In [25], the authors proposed an Association Rule Classifier (ARC) as a novel framework that captures different hypertext features such as text, anchor text, metadata, which were used as features in classification. They have shown an accuracy improvement over 65% for large vocabulary size data sets. The authors in [26] proposed an approach for an automatic categorization of web pages using the structure information of web document such as the line no, placement of links, text and images present in it for classification. Their results were quite good; however, they suggest that results would have been more encouraging if they combine the traditional text based approach along with their proposed approach.

Our method mixes the web page textual content with the features such as title, metadata, anchored text extracted from the URL's present in a testing web page i.e. target web page, so that there would be enough information available for the classifier to classify any given web page into pre-determined categories. This model is independent of the classifier, thus allowing us to focus on the different ways of extracting the relevant features information for the classification of web page. Our approach also provides a flexible way of incorporating any other link based, web page content based methods while ensuring the accuracy of the classification.

III. PROPOSED APPROACH

Many researchers have focused on either improving a Web page classifier or developing a novel classifier. These classifiers do need certain amount of information or features for classifying any web page. But as the most of the web pages such as commercial web site pages are completely filled with images and links, contain very less information that is not sufficient for the classifiers to classify them. We have proposed an algorithm, which focuses on classifying the web pages irrespective of enough content present in it or not by using either textual information alone present in those web pages or by using both the textual content as well as the features extracted from the URL's present in that given web page. The features we extract from each URL are such as Title, Metadata, Anchored Text and Other Heading tags such H1, H2, H3.etc. Our proposed approach algorithm is as follows:

1. Represent each web page of all categories in a given training data set as a vector.
2. Generate the centroid vector for each category in training data that represents all the web pages corresponding to that category.
3. To predict the category of a given any new testing web page i.e. the target web page, first extract the whole textual information present in that target web page and generate a vector for the whole extracted textual information using term frequency and inverse document frequency (tf-idf) scoring mechanism.
4. Check whether the generated vector value is very less like less than value 0.1 in our approach; If it is more than 0.1 predict its category based on solely the textual content of the target web page. If the vector threshold value is less than 0.1, go to step 5.
5. Retrieve all the URL's present in the given target web page.
6. Get the extra necessary information required for the classifier from each URL by extracting the following features such as Title, Meta data, Anchored text and other Heading tags such as H1, H2 and H3 etc.
7. Concatenate the extracted feature's information from each URL with the actual text information present in that target web page.
8. Generate the vector for the whole concatenated information.
9. Compute the Cosine Similarity between the concatenated text vector and all centroid vectors of each category in training data set.
10. Predict the category of the testing web page i.e. the target web page by checking how close the concatenated vector is close to the centroid vectors using cosine similarity measure, whichever the centroid vector the concatenated text vector is closer to, that category the given target web page belongs to.

In [31] they used a simple centroid based algorithm which is an instance of Rocchio relevance feedback method [32]. In this approach, the documents are represented as vectors using the vector-space model [33]. In this model, each document d is considered to be a vector in the term-space. In its simplest form, each document is represented by the *term frequency* (TF) vector d_{tf}= ($tf_{1,}$ tf_2, $tf_{3,}$ . . . , tf_n) where tf_i is the frequency of the i th term in the document. A widely used refinement to this model is to weight each term based on its *inverse document frequency* (IDF) in the document collection. The motivation behind this weighting is that terms appearing frequently in many documents have limited discrimination power, and for this reason they need to be de-emphasized. This is commonly done [34] by multiplying the frequency of each term t_i by log(N/df_i), where N is the total number of documents in the collection, and df_i is the number of documents that contain the i^{th} term (i.e., document frequency). This leads to the *tf-idf* representation of the document, i.e d_{tfidf}= (tf_1 log (N/df_1), tf_2 log (N/df_2), . . . , tf_n log (N/df_n)). In order to account for documents of different lengths, the length of each document vector is normalized so that it is of unit length, i.e. ,$|d_{tfidf}| = 1$. We will assume that the vector representation of each document d has been weighted using *tf-idf* and it has been normalized so that it is of unit length.

To know how close two documents d_i and d_j in vector space model, we calculate the similarity between two documents d_i and d_j is commonly measured using cosine function [34] only given by

$$\cos(d_i, d_j) = \frac{d_i \, . \, d_j}{|d_i| * |d_j|}$$

Where "." denotes the dot product of the two vectors. Since the document vectors are of unit length, above formula simplifies to$\cos(d_i, d_j) = d_i \, . \, d_j$. Given a set S of documents in each category in training data and for their corresponding category vector representations, we define the centroid vector C for each category of training data is given below

$$C=\frac{1}{|S|}\sum_{d \in S} d$$

Where $|S|$is the no of documents in each category in training data.This is nothing more than vector obtained by averaging the weights of all web page vectors of each category S in training data. Each centroid vector is computed to represent the web pages of each category in training data set, and a new document's class is predicted based on how similar it is to any centroid vectors using cosine similarity function. The similarity between centroid vector C and between a document vector d is computed to predict which category the document vector d belongs to by using the following cosine measure.

$$\cos(d, C) = \frac{d \, . \, C}{|d| * |C|}$$

Note that even though the document vectors are of length one, the centroid vectors need not necessarily be of unit length. The Cosine similarity measure allows the centroid-based scheme to classify a new document based on how closely its behaviour matches the behaviour of the centroid vectors of each category in training corpus.

IV. EXPERIMENTAL RESULTS

We have experimented our proposed approach using the dataset "WebKB" downloaded from UCI repository. It is a benchmark dataset used for machine learning problems. It is the university database containing 8282 web pages, having seven categories such as Course, Project, Student, Faculty, Department, Staff, and Others, with each category containing 930, 182, 1641, 1124,182,137, and 3764 no of web pages respectively. We have applied our proposed approach on the "WebKB" by first applying some pre-processing techniques to delete unnecessary information such as stop words, dead URL's, images and tables etc from web pages to get the quality data for achieving the better classification results.

We have taken 10 sample testing web pages for prediction of their categories using either only the textual content present in each web page or using both the textual content along with features extracted from each URL are Title, Metadata, Anchored text and Heading tags H1,H2, and H3 tags. The comparison of using only the textual content of web page and using both the textual content along with the extracted URL features results are shown in below figures respectively.

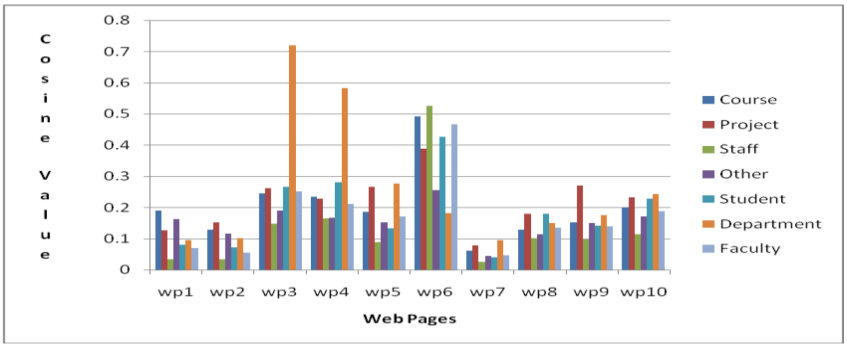

Fig.1: Classification of Web Pages using only Textual Content.

Fig.1 shows how 10 sample testing web pages i.e. target web pages are classified into any one of the predetermined categories such as Course, Project, Staff, Other, Student, Department and Faculty. We first generate the centroid vectors of all those categories in training data set, then generate the vectors using the textual information alone present in those 10 testing web pages, in fig.1 only the textual content of the web pages are used for classification. Prediction of which category those 10 web pages belong to is determined by the cosine similarity value. Cosine similarity value is computed between each testing web page vector and each centroid vector. Whichever the centroid vector the testing web page vector is closer to i.e. maximum similarity, that category the given testing web pages belongs to.

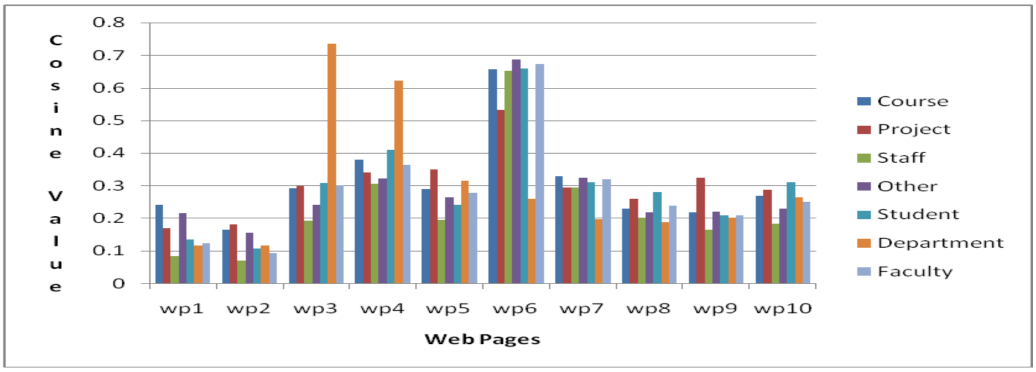

Fig.2: Classification of Web Pages in Vector Space Using both Textual Content and URL Features

Fig.2 shows the classification of web pages by using both the textual content along with the extracted URL features such as Title, Meta data, Anchored text, Heading tags H1, H2, H3 etc. It is shown in Fig.2 that there is an improvement in getting the maximum similarity value compared to Fig.1. Thus our approach shows that classifying web pages using URL's features along with the textual information of web pages helps in achieving improvement in accuracy and allows us to classify any given web page even though it does not contain enough information present in it or not.

We have also experimented our proposed approach on WEKA by simulating our training data set "WebKB" on WEKA. We have tested 100 web pages for classification on WEKA using just textual content first and then by using both actual web page textual content along with the extracted URL features together. The algorithms used for classification that are supported by literature for best text mining algorithms are K-Nearest Neighbour (K-NN), Support Vector Machine (SVM), and NaiveBayes[27] which are available in WEKA giving promising results on the dataset "WebKB" using our proposed approach.

K Nearest Neighbours (K-NN) is a memory based classification algorithm, which is a non-parametric inductive learning algorithm storing the training instances in a memory structure. The key point of KNN algorithm is to predict the class label of a test data point by a majority vote of k nearest neighbour's of these data points, with ties being broken at random.

Support Vector Machine (SVM) is based on well developed statistical learning theory, so it is well founded theoretically. The main idea of SVM is to select a small number of critical boundary samples from each class and build a linear discriminate function that widely separates training instances. The SVM technique will import to automatically map the training samples into a higher-dimensional space, and to learn a separator in the mapped space if instances are not linear separation. A linear SVM is a hyper plane that separates a set of positive examples from a set of negative examples with maximum margin. The margin is the distance from the hyper plane to the nearest of the positive and negative examples. SVM is often time consuming and it is more comfortable to small-size problems since the high complexity.

NaiveBayes algorithm [27] is a simple but effective text classification algorithm for learning from labelled data alone. It provides the basis for probabilistic learning methods that accommodate and require knowledge about the prior possibilities of alternative hypotheses and about the probability of observing various data given the hypothesis.

The results using just only the textual content and also using both the textual content along with features extracted from each URL are Title, Metadata, Anchored text and Heading tags H1, H2, and H3 tags are used for the classification on WEKA by simulating the both training data set “WebKB” and 100 testing web pages that are tested using K-NN, SVM and NaiveBayes classification algorithms. The comparison of using only the textual content of web pages and using both the textual content along with the extracted URL features results are shown in below tables respectively.

Table I: Classification of 100 Testing Web Pages using only the Textual Content

No of Testing Web Pages	No of Training Web Pages(WebKB)	Classifier	Accuracy	Average Precision	Average Recall	F-Measure
100	8282	KNN	89	0.915	0.89	0.895
100	8282	SVM	69%	0.769	0.69	0.691
100	8282	Naive Bayes	60%	0.676	0.6	0.589

Table I shows how 100 testing web pages are classified into any one of the predetermined categories such as Course, Project, Staff, Other, Student, Department and Faculty using WEKA. Here only the textual content of the web pages are used for classification. It shows that KNN outperforms other two algorithms and got highest accuracy of 89%, SVM got 69% and NaiveBayes got 60% by using only the textual content of the web pages.

Table II: Classification of Testing Web Pages using both Textual Context and URL features

No of Testing Web Pages	No of Training Web Pages(WebKB)	Classifier	Accuracy	Average Precision	Average Recall	F-Measure
100	8282	KNN	90	0.927	0.9	0.905
100	8282	SVM	71%	0.75	0.71	0.718
100	8282	Naive Bayes	63	0.694	0.63	0.613

Table II shows that how 100 testing web pages are classified into any one of the predetermined categories using both the textual content of the web pages and URL features.. It is shown that using both textual content and URL's features there is an improvement in accuracy when tested using three algorithms KNN, SVM and Naive Bayes on the same 100 testing web pages. In comparison of Table I and Table II, it shows clearly that K-NN achieves 1% improvement in accuracy from 89% to 90%, SVM achieves 2% improvement in accuracy from 69% to 71% and NaiveBayes achieves accuracy improvement of 3% from 60% to 63% using both the textual content and URL's extracted features.

It also shows that there is considerable improvement in Accuracy, Precision, Recall and F-measure [33] using our proposed approach on “WebKB” data set.

V. CONCLUSION AND FUTURE WORK

We have achieved the improvement in accuracy by using the features extracted from URL's of a given web page along with the textual content of web page. Our proposed approach can also classify any given web page irrespective of enough information present in it or not. We used the best text mining classifiers in our experiment and showed that K-NN performs much better in accuracy than the other two algorithms SVM and NaiveBayes. In future, though our proposed work is time efficient and giving improvement in accuracy, we would like to further reduce the time taken to build classification model by not taking every URL into consideration for extracting additional information through URL features when given web page does not have sufficient information to be classified. We would like to focus in future on designing the approaches which will take only relevant URLs instead of all URL's for classification of web pages along with its actual text content.

REFERENCES

[1]. John.M.Pierre, “Practical Issues for Automatic Categorization of Web sites” , September 2000.

[2]. Yahoo, http://www.yahoo.com/

[3]. Looksmart, http://www.looksmart.com/

[4]. Open Directory Project, http://www.dmoz.org/

[5]. Xiaoguang Qi and Brian D. Davison "Web Page Classification: Features and Algorithms," ACM Computing Surveys, vol.41, no.2, article 12, Feb 2009.

[6]. Sun Bo, Sun Qiurui, Chen Zhong, Fu Zengmei "A study on Automatic Web Pages Categorization" IEEE International Advanced Computing Conference, Patiala, India, 6-7 March 2009.

[7]. L. Yu, H. Liu. "Feature selection for high-dimensional data: a fast correlation-based filter solution". In Proceedings of ICML, 2003.

[8]. M. Robnik-Sikonja and I. Kononenko. "Theoretical and Empirical Analysis of ReliefF and RReliefF". Machine Learning 53(1-2):23.69, 2003.

[9]. Zhongli He, and Zhijing Liu "A Novel Approach to Naive Bayes Web Page Automatic Classification", IEEE Fifth International Conference on Fuzzy Systems and Knowledge Discovery,2008.

[10]. C.H. Chen, "The Use of Independent Component Analysis as A Tool for Data Mining", vol. 2, pages 1032-1034 , *IGARSS '2002*,.

[11]. Zhaohui Xu, Fuliang Yan, Jie Qin, Haifeng Zhu "A Web Page Classification Based on Link Information", IEEE 10th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, 2011.

[12]. Jiao Lijuan, Feng Liping "Improvement of Feature Extraction in Web Page Classification", IEEE 2nd International Conference on E-business and Information System Security, May 2010.

[13]. Pavel Calado Marco Cristo, Edleno Moura "Combining Link Based and Content-Based methods for Web Document Classification", CIKM '03 proceedings of the Twelveth International Conference on information and management, pages 394-401,ACM,New York,USA,2003

[14]. A. Sun, E.-P. Lim, and W.-K. Ng. "Web classification using support vector machine". In Proceedings of the fourth international workshop on "Web information and data management", pages 96-99, ACM Press, 2002.

[15]. J. Furnkranz. "Exploiting structural information for text classification on WWW". ACM proceedings of the Third international symposium on advances in Intelligent Data Analysis,pages-487-498,1999

[16]. E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M.Pennock, and G. W. Flake. "Using Web structure for classifying and describing Web pages". In Proceedings of WWW-02, ACM International Conference on the World Wide Web, 2002.

[17]. Y. Yang, S. Slattery, and R. Ghani. "A study of approaches to hypertext categorization". Journal of Intelligent Information Systems, Kluwer Academic Publishers, volume 18, number 2-3, pages 219-241,2002.

[18]. S. Slattery and M. Craven. "Discovering test set regularities in relational domains". In P. Langley, editor, Proceedings of ICML-00, 17th International Conference on Machine Learning, pages 895-902, Morgan Kaufmann Publishers, San Francisco, Stanford,US,2000

[19]. J. M. Kleinberg. "Authoritative sources in a hyperlinked environment." Journal of the ACM (JACM), 46(5), pages 604-632, 1999.

[20]. M. Fisher and R. Everson. "When are links useful? Experiments in text classification". In F. Sebastianini, editor, Proceedings of the 25th annual European conference on Information Retrieval Research, ECIR 2003, pages 41-56. Springer-Verlag, Berlin, Heidelberg, DE, 2003.

[21]. S.Chakrabarti, B.Dom, and P.Indyk, "Enhanced hypertext categorization using hyperlinks". In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 307-318, Seattle, Washington, June 1998.

[22]. H.-J.-J. Oh, S. H. Myaeng, and M.-H. Lee. "A practical hypertext categorization method using links and incrementally available class information", In Proceedings of the 23rd annual international ACM, SIGIR conference on Research and development in information retrieval, pages 264-271. ACM Press, 2000.

[23]. Rayid Gani, Sean Slattery, Yiming Yang "Hypertext Categorization using Hyperlink Patterns and Meta Data", ACM, ICML Proceedings of the18th International Conference on Machine Learning, pages 175-185,2001.

[24]. Li Xiaoli and Shi Zhongzhi, "Innovating Web Page Classification Through Reducing Noise", Journal of Computer Science and Technology, Volume 17 Issue 1,Pages 9 – 17, January 2002.

[25]. Fathi, Mohamed, N. Adly, and M. Nagi. "Web Documents Classification Using Text, Anchor, Title and Metadata Information." *International conference on Computer Science, Software Engineering, Information Technology, e-Business and Applications*, pages 445-452, Cairo, Egypt, 2004.

[26]. Arul Prakash Ashirvatham, Kranthi Kumar.Ravi "Web Page Classification based on Document Structure", IEEE National Convention,2001.

[27]. Yong Wang, Julia Hodges, Bo Tang, “Classification of Web Documents Using a Naive Bayes Method”, Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’03), 2003

[28]. Susan Dumais, Hao Chen, “Hierarchical Classification of Web Content”, SIGIR , ACM, pp 256 – 263,2000.

[29]. Aixin Sun,Ee-Peng Lim and Wee-Keong Ng, “Web Classification Using Support Vector Machine”, Proceedings of the 4th international workshop on Web Information and Data Management held in conj. With CIKM, USA, Nov 2002.

[30]. JIU-ZHEN LUNG, “SVM MULTI-CLASSIFIER AND WEB DOCUMENT CLASSIFICATION”, Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004, pp 1347 – 1351.

[31]. J.J. Jr. Rocchio, “The SMART retrieval system: Experiments in automatic document processing”.

[32]. In Gerard Salton, editor,” *Relevance feedback in information retrieval”*. Prentice-Hall, Inc., 1971.

[33]. G. Salton. *Automatic Text Processing: “The Transformation, Analysis, and Retrieval of Information by Computer”*. Addison-Wesley, 1989.