Journal of Computer Science 10 (10): 1969-1976, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.1969.1976 Published Online 10 (10) 2014 (http://www.thescipub.com/jcs.toc)

SUPERVISED TERM WEIGHTING METHODS FOR URL CLASSIFICATION

R. Rajalakshmi

Department of Computer Science and Engineering, SSN College of Engineering, Chennai, India

Received 2014-04-03; Revised 2014-04-10; Accepted 2014-05-13

ABSTRACT

Many term weighting methods are suggested in the literature for Information Retrieval and Text Categorization. Term weighting method, a part of feature selection process is not yet explored for URL classification problem. We classify a web page using its URL alone without fetching its content and hence URL based classification is faster than other methods. In this study, we investigate the use of term weighting methods for selecting relevant URL features and their impact on the performance of URL classification. We propose a New Relevance Factor (NRF) for the supervised term weighting method to compute the URL weights and perform multiclass classification of URLs using Naive Bayes Classifier. To evaluate the proposed method, we have conducted various experiments on ODP dataset and our experimental results show that the proposed supervised term weighting method based on NRF is suitable for URL classification. We have achieved 11% improvement in terms of Precision over the existing binary classifier methods and 22% improvement in terms of F1 when compared with existing multiclass classifiers.

Keywords: Web Page Classification, URL Features, Term Weighting Method, ODP

1. INTRODUCTION

Web page classification is the task of assigning one of the predefined category labels to the web page being considered based on its contents and topic it talks about. It resembles text categorization, but with more challenges due to the presence of hyperlinks, images and multimedia content. So automated web page classification systems make use of structure, title of the web page and sibling pages in addition to the full text of the page. Exponential increase in number of web sites in World Wide Web makes it difficult for classification systems to handle large number of web pages with high dimensional feature space. Some of the issues in the content based classification systems are the following: (i) Contents are needed for extracting features forcing us to download the page for classification purpose (ii) wastes bandwidth in unnecessary downloads (iii) slows down the classification process as excessive features are to be extracted. Other than this additional burden, content based classification systems are not sufficient to address the following challenges when (i) web page contains only images (ii) the content is hidden behind the images (iii) contains dynamic content.

URL based classification systems are developed to overcome all these difficulties. As every web page is associated with an unique URL, the information present in the URL can be exploited for classifying the web pages. URL based web page classification is a very challenging task since URL is a small fraction of the web page. URL may contain compound words (http://www.moneycontrol.com/), abbreviations (http://www.isbm.edu.in), non-meaningful and part-of-a-word (http://www.infosys.com). The advantages of URL classification includes the following: (i) Features are extracted from URLs alone thereby avoiding the need for unnecessary downloads that waste bandwidth (ii) increases the speed of classification (iii) helpful for information filtering task in blocking some websites before accessing.

URL classification problem is studied by many researchers (Kan, 2004; Kan and Thi, 2005; Baykan *et al*., 2011; Rajalakshmi and Aravindan, 2011; Singh *et al*., 2012) and various URL features are suggested in the

literature. Kan and Thi (2005) suggested segmentation techniques for extracting features from URLs. Token based features are suggested in Baykan *et al.* (2011); Rajalakshmi and Aravindan (2011). The n-gram based approach for URL classification is discussed in (Jianping *et al*., 2006; Baykan *et al*., 2011; Rajalakshmi and Aravindan, 2013). In the method suggested by Jianping *et al.* (2006), sequences of n-grams are derived from URLs and feature selection methods are applied to reduce the dimension of feature space. They have proposed a feature selection measure known as R-measure to filter the relevant features and used those URL features to classify the objectionable web pages. 3-grams alone are extracted from URLs and used as features in a fixed feature dimensional space of 26^3 in the approach suggested by Rajalakshmi and Aravindan (2013). Baykan *et al.* (2011) derived all-gram (n-grams with n = 4 to 8) features from URLs and they used traditional *tf * idf* scoring method as the feature weighting method.

The popular term weighting method *tf * idf* and its variants are widely used in Information Retrieval (IR) and for Supervised Learning tasks such as Text Categorization (TC). Baykan *et al*. (2011) followed this weighting method for URL classification also. But for categorization tasks, training data is available with class labels and this rich source of information can be utilized in weighting the features. For text categorization, different supervised term weighting methods are suggested in the literature (Debole and Sebastani, 2003; Lan *et al*., 2009). By assigning higher weights for relevant terms, the performance of classification can be improved (Lan *et al*., 2009).

In this study, we propose a New Relevance Factor (NRF), a variant of supervised term weighting method for URL classification. Using this measure, the relevance of each term with respect to category can be accurately measured. As the URLs are of short length, excessive feature removal will affect the classification performance and hence we apply continuous feature selection method in which all the features are taken into account along with their feature weights. To the best of our knowledge, this is the first work in URL classification that applies term weighting methods to select the relevant URL features. To study the significance of term weighting methods, we evaluated our proposed method on ODP dataset using Naïve Bayes as the classifier. Experimental results show that feature weighting method based on the proposed new relevant factor NRF, improves the multiclass performance of URL classification. When compared with existing methods, there is an improvement of 11% in precision with existing binary classifiers and 22% improvement in F1 for multiclass classifiers.

This study is organized as follows: Related works are discussed in section 2. The proposed method is described in section 3. Experiments and results are detailed in section 4 followed by conclusion in section 5.

2. RELATED WORKS

2.1. Web Page Classification

Content based web page classification is slow as the features are extracted from the pages and also requires downloading the contents. To improve the classification speed, Kan and Thi (2005) suggested URL features for web page classification. In this approach, they segment the URL into meaningful chunks and features are derived from URLs. They reported an F1 measure of 0.525, by applying Maximum Entropy as their classifier on WebKB dataset. For classifying URLs, an n-gram based approach is followed by Rajalakshmi and Aravindan (2011) in which only 3-grams derived from URLs are used as the features. In this approach, the dimensionality of feature vector is restricted to a maximum of 26^3 features. This method was evaluated on two datasets ODP and WebKB method with two machine learning algorithms SVM and maximum entropy classifier. Another URL based approach is suggested by Rajalakshmi and Aravindan (2013), in which tokens of URLs are used as features. They classified the web pages based on the token match strategy by using 9 categories of ODP dataset with naïve bayes classifier. Using URL features, a statistical based approach is followed by Hernandez *et al.* (2012) to cluster the web pages. They used tokens of URL and the sequence information to cluster the pages. Rose and Chandran (2012) proposed a similarity measure based method for web query classification. By using normalized web distance based method along with NLP based technique, they classified the queries.

In the proposed approach, we do not use title, anchor text or contents of the web page for classification purpose and use only URL features. To select the relevant URL features, we apply term weighting methods and use naïve bayes classifier to perform multiclass classification. In the next section, we briefly describe the existing term weighting which are widely used in text categorization.

2.2. Term Weighting Methods

Term Weighting is a method of assigning appropriate weights to each term in the feature space to improve the performance of the classifier and also

useful in other data mining tasks. The representation of a document is important for any learning algorithm in text categorization and information retrieval problems. Vector Space Model (VSM) is the commonly adopted method, in which each document is represented by a vector of its terms in the term (feature) space. The creation of document representation, also known as document indexing involves two phases: Term selection and term weighting (Debole and Sebastani, 2003). In the term selection phase, a Subset of terms (S) is selected from the set of all Terms (T) in training documents and Term weighting is the second phase in which, for every document d_j, for every term t_i selected in first phase, a weight $0 < w_{ij} < 1$ is computed. This weight represents, how much that term t_i contributes to discriminate the semantics of document. In general, the classifier learning does not affect the term weighting phase. Debole and Sebastani (2003) proposed the idea of supervised term weighting method, in which the term weight w_{ij} reflect the importance of a term t_i to decide the membership of the document d_j to the categories. We briefly describe the traditional term weighting method and the supervised term weighting methods in this section.

The Inverse Document Frequency (IDF) is widely used in Information Retrieval tasks. It is based on the ratio of total number of documents in the collection (N) and the number of documents in the collection (n_i) which contain (or are indexed by) the term in question. This measure is used as the weight to be applied to term t_i by combining it with term frequency. The term frequency *tf* is used to denote the number of occurrences of the term t_i in the document d_j. To eliminate the length effect, the term weight is restricted to lie in the range between 0 and 1, so it is normalized using cosine normalization. The *idf* is defined as follows Equation 1:

$$idf = log\ (N / n_i) \quad (1)$$

The assumption of Inverse Document Frequency (IDF) is that, if a term t_i in query q, appears in too many documents, then it is not helpful to discriminate the documents relevant to q from the irrelevant and it should be given less weight than one which occurs in few documents. This IDF assumption is reasonable for IR tasks as the training data is not available with class information. But for categorization tasks, we have training data with class labels. This class information can be used in term weighting methods so that relevant terms will get higher weights and hence improve the classification performance. For URL classification, we have categorized / labelled URLs as training data, so we prefer to use term weighting methods by taking class labels into consideration.

For Text Categorization (TC) problems, the feature selection metrics such as Chi-Square test (CHI), Information Gain (IG) and Mutual Information (MI) are combined with tf and used as term weighting methods. The methods tf * CHI, tf * IG and tf * MI suggested in the literature make use of class information while computing weights for each term and are known as Supervised term weighting methods. These methods assign weights to the terms based on the distribution among both positive and negative categories. But in multiclass classification, one category is considered positive and all the remaining categories are considered as negative. The above supervised methods take the distribution of term t_i in the positive and negative categories same as that of negative and positive categories respectively.

As the URL classification is a multiclass problem, we want to weight the terms in positive category more than the negative categories. Lan *et al*. (2009) suggested a measure named Relevance Frequency (RF) and proposed a supervised term weighting method *tf * rf* by considering the distribution of relevant documents in the collection. The basic idea of *tf * rf* is that, if a high frequency term is more concentrated in the positive category than in the negative category, then it makes more contributions in selecting the positive samples than the negative samples. The notations *a* and *c* are used to denote the number of documents in the corpus in positive/negative category that contain the term. The number of documents that do not contain the term in the positive/negative category is denoted by *b* and *d*. In *tf * rf* method, a term's discriminating power is determined by the number of relevant documents that contain this term only, i.e., *a* and *c*. They defined the ratio of a to c as the Relevance Frequency (RF) and replaced the *idf* measure while combining it with tf to weight a term. This method of term weighing is defined as follows Equation 2:

$$tf * rf = tf * log\ (2 + (a / max(1, c)) \quad (2)$$

We use a variant of *rf* to determine the relevancy of URL terms accurately and follow our own method to weight each URL term while using Naïve Bayes classifier.

3. PROPOSED TERM WEIGHTING METHOD FOR URL CLASSIFICATON

In this section, we describe URL feature extraction process and the proposed methodology for selection of relevant features by applying term weight method.

3.1. URL Features

For any classification problem, we need to preprocess the input data, extract features and then apply feature selection method to obtain suitable features. For classifying URLs we extract tokens and n-grams from URLs that are referred as URL features. Consider the following example:

- URL: http://www.isbm.edu.in
- Tokens: Isbm, edu, in

Tokens of an URL are obtained by preprocessing the URL in which the words 'http', 'www' and all non-alphabet characters such as hyphen, underscore, space, ':', '/', '.' are removed. As discussed in section 1, tokens contain concatenated words, abbreviations and non-meaningful words, so we derive n-grams from URLs instead of applying complex segmentation techniques. n-grams are the sequence of characters of length n. By concatenating the URL text, sequence information can also be captured. So we concatenate the URL after the preprocessing step and then derive n-grams. We extract 3-grams, 4-grams and 5-grams from the concatenated URL text. The n-gram (n = 3 to 5) features considered in the above example are shown below:

Concatenated URL text: Isbmeduin
3-grams: isb, sbm, bme, med, edu, dui, uin
4-grams: isbm, sbme, bmed, edui,duin
5-grams: isbme, sbmed, bmedu, medui, eduin

As it is evident from the above example, we have many noisy and redundant URL features in the feature space that may affect the classification performance and also increase the feature dimensionality. So we apply a feature selection method to obtain the relevant features by eliminating the unnecessary ones from the feature space. We propose two methods for selecting relevant URL features.

3.2. Feature Selection with Binary Relevance Based Supervised Term Weighting Method

We have a large scale data containing millions of URLs that result in a very high dimensional feature space with many thousands of features. To filter the unnecessary features, we apply a feature selection method that involves simple term weighting approach. The traditional term weighting method based on tf * idf is not utilizing the class labels in weighting the features. URL classification is a multiclass problem and we consider URL as a short document with tokens and n-grams as its terms (features). Assume that we have K distinct terms t_i in the corpus. We define our own term weighting method for each of these terms $t_i \in K$ according to its relevance with respect to category. Our idea is based on the assumption that, a term t_i is relevant and important for a category C_m, if the corresponding training URLs contain that term. We assign a relevant factor of 1 for all those terms that appear at least once in C_m category training URLs so that its feature space contains only those relevant terms of C_m. We define the binary relevance factor BRc_m, of a term t_i for category C_m as follows:

$$BRc_m = 1 \; if \; t_i \in T_m \tag{3}$$

Here the T_m denotes the set of all terms present in category C_m URLs. The relevancy is either 1 or 0 based on the term's presence/absence in the corresponding training URLs of C_m.

We refer this term weighting method as Binary Relevance based term weighting method (BR). Unlike IDF method, the proposed BR method is a supervised term weighting method as it takes class information into account. Taking the category information for weighting the terms is helpful for multiclass URL classification.

3.3. Feature Selection with New Relevance Factor Based Supervised Term Weighting Method

In the Binary Relevance based term weighting (BR) method, without considering the presence of that term in other categories, we assign a relevance factor of 1 to all the terms in that category. This may increase the confusion among classes, when a term is present in more than one category. We want to weight the relevant terms for a particular category C_m to be high even if it appears only once in that category but absent in all the other categories. Similarly we have to assign a larger weight to those terms that appear many times in one particular category than the remaining categories in which it may be less frequent. So we propose another relevance factor that considers the ratio of term's presence in one category over the other categories in the corpus.

By applying relevance measure suggested in the *tf * rf* method, we weight the relevant terms highly than the irrelevant terms. In this method, we use all the training URLs in the corpus and compute the

Relevance Frequency (RF) of each term with respect to each category. The rf method assigns a constant value of log 2 as the relevance frequency even if a term is not relevant for category C_m and does not appear at least once in the training URLs of C_m. In URL classification, each term present in URL contributes its weight for classification decision. So we avoid this constant value and assign a Relevancy Factor (RF) of 0 for those terms if it is not present in category C_m. So we propose a New Relevance Factor (NRF) that accurately measures the relevancy of a term for a category C_m and define it as given below Equation 4:

$$NRFC_m = \log (2 + (a / \max(1,c)) - \log 2 \quad (4)$$

Here 'a' denotes the number of URLs that contain term t_i and belong to category C_m and 'c' denotes the number of URLs that contain term but belong to categories other than C_m. In this way, we calculate category-wise relevance factor for all the terms in the training corpus so that it is having high discriminating power while classifying the URLs. By using the proposed new relevance factor, we can filter the irrelevant terms for a category and classification performance can be improved.

3.4. Naïve Bayes Approach for URL Classification

We use Naive Bayes (NB) classifier to classify the URLs. It is suitable for URL classification as it can handle large scale data with many thousands of features. For text categorization tasks, NB classifier is applied by considering words as features and likelihood probability of a document is estimated from its individual word probabilities. For URL classification, we use the tokens and n-grams derived from the URLs as the features. The contribution of these features in classification decision of the given URL depends on individual weights of each term. We use *tf * nrf* for weighting each term in the test URL. The final weight of the URL is obtained by adding all its term weights, from which the likelihood probability $P(URL|C_m)$ is computed. Instead of using traditional cosine normalization, we apply our own normalization method while computing the likelihood probability. We add nrf value of all the relevant terms in a particular category C_m and refer it as $NRFsumC_m$. We use $NRFsumC_m$ to normalize the weight obtained for an URL. We estimate the prior probabilities $P(C_m)$ for every category C_m by dividing the number of URLs in each category by the total number of URLs in the corpus.

Using Naïve Bayes assumption, the posterior probability $P(C_m|URL)$ of an URL to belong to category C_m is computed by multiplying the likelihood probability with the corresponding prior probability. The procedure applied for classifying URLs using Naive Bayes Classifier is summarized in the following steps:

1. Preprocess the URL U and split into terms t_i and then count their respective frequency tf_{iU} in U.
2. For every category C_m repeat the following steps:
   a. For every term t_i in the given URL, for this category C_m, obtain the new relevance frequency $NRFC_m(t_i)$ from training phase if it is present
   b. Calculate weight of term t_i in C_m using:

$$w_{iU} = tf_{iU} * NRFC_m(t_i)$$

   c. Calculate the weight of URL U in C_m using:

$$wgt(U,C_m) = \sum_{i=1}^{n} w_{iU}$$

   d. Calculate the likelihood probability using:

$$P(URL|C_m) = wgt(U,C_m) / NRFsumC_m$$

   e. Calculate the posterior probability:

$$P(C_m|URL) = P(URL|C_m) * P(C_m)$$

3. Find the maximum value and assign that category as the predicted category to the URL U:

$$C = argmax_m P(C_m|URL)$$

4. RESULTS

To evaluate our proposed method we conducted experiments on the ODP dataset. We used Java for implementing all algorithms including Naive Bayes Classifier.

4.1. Benchmark Dataset

The Open Directory Project (ODP) DMOZ dataset is the most widely used benchmark dataset for we page classification. We considered 8 categories in the ODP dataset viz., Computers (C1), Games (C2), Health (C3), Home (C4), News (C5), Recreation (C6), Reference (C7) and Shopping (C8). We have taken a total of 4,32,500

URLs from the above 8 categories in which the number of URLs is not uniformly distributed. The number of URLs in each category is as follows: 96900, 42900, 51300, 22300, 7500, 84500, 49000, 78100. We have randomly selected 80% of URLs for training and kept the remaining 20% URLs for testing. We performed multiclass classification using this test set comprising a total of 86,500 URLs, combined from 8 different categories. We used Accuracy, Precision and F1 as the evaluation measures.

4.2. Experiments

To analyze the significance of term weighting methods for URL classification and to study the performance our proposed method, we performed experiments using both existing term weighting methods (idf and rf) and the proposed relevance factor based methods (BR and NRF). In the first experiment, we used IDF with NB algorithm in which, category information is not used to compute the weight of terms in training URLs. To classify a test URL, we extracted tokens and n-grams (n = 3 to 5) from it and followed the procedure detailed in Algorithm 1, by using IDF. As IDF is not considering relevant terms, multiclass accuracy of 0.22 was achieved in this method. For experiment 2, we followed the BR method discussed in section 3.2 and assigned relevance factor of 1 to all the relevant terms present in the training URLs of C_m. We used those relevant terms of every category C_m, as features and applied Naive Bayes algorithm. This improved the accuracy of multiclass classifier to 0.41. In Experiment 3, we used tf*rf method, in which more relevant features were selected. We achieved multiclass accuracy of 0.52 for this tf*rf method. To accurately measure the relevancy of each term, we applied New Relevance Factor (NRF) based term weighting method as discussed in Section 3.3 and performed Experiment 4. We computed the weight of the test URL based on the new relevance factor $NRFC_m$ already computed for each category for each term in the training corpus. We classified the URL using Naive Bayes Classifier. The accuracy is improved further and with our proposed method, we achieved 0.54 for this multiclass URL classification. By this method, we are able to achieve higher precision for all the categories, especially the categories Games, Health, Home and News have a precision of above 0.90.

5. DISCUSSION

We compared the multi-class accuracy for existing term weighting methods (IDF and RF) with our proposed two new methods, Binary Relevance (BR) and New Relevance Factor (NRF) based term weighting methods. As discussed in the section 4.2, we achieved multi-class accuracy of 0.22, 0.41, 0.52 and 0.54 for IDF, RF, BR and NRF as shown in **Fig. 1**. The BR method performs better than IDF as we considered class information in weighting the terms, but compared to existing RF method, the performance of BR is low. The proposed NRF based term weighting method is better than the RF method and the accuracy is improved by 3% for this method.

We compared the precision of individual classifiers of Naïve Bayes multiclass classifier with four term weighting schemes discussed in this study. The precision of individual classifiers are as follows: 0.55, 0.97, 0.92, 0.99, 0.91, 0.75, 0.85 and 0.34. It is illustrated in **Fig. 2**. We achieved an average precision of 0.08, 0.62 and 0.64 for IDF, Binary Relevance and RF methods respectively for individual classifiers of multiclass classifier. With the proposed NRF method, we are able to achieve an average precision of 0.79 which is higher than all the other three term weighting methods.

For this multiclass problem of URL classification, many existing methods use binary classifiers rather than direct multiclass classifier. The confusion of categorizing an URL among two classes is less compared to confusion among m different categories. Even if we compare our multiclass individual classifier's precision with the existing individual binary classifiers, our method outperforms by achieving higher precision. By designing binary classifiers using SVM, Reinforcement Learning and Online Incremental Learning with same 8 categories of ODP dataset, Singh *et al*. (2012) classified URLs. We compared the results of our NRF method with those three methods of Singh *et al*. (2012) and it is illustrated in **Fig. 3**.

With our proposed method, we are able to reduce the false positive rate and achieve high precision as we eliminate the irrelevant terms and take only the highly relevant terms for a category, so that their weights make positive contribution in classification.

We compared the multiclass performance of our proposed method with the existing works in the literature for multiclass classification and tabulated the results in **Table 1**.

No feature selection method was applied in the approaches suggested in (Kan and Thi, 2005; Baykan *et al*., 2011) and they utilized all the URL features without discarding the irrelevant ones and used SVM as their classifier. As we take only the relevant features, we are able to achieve 0.54 as the F1 value.

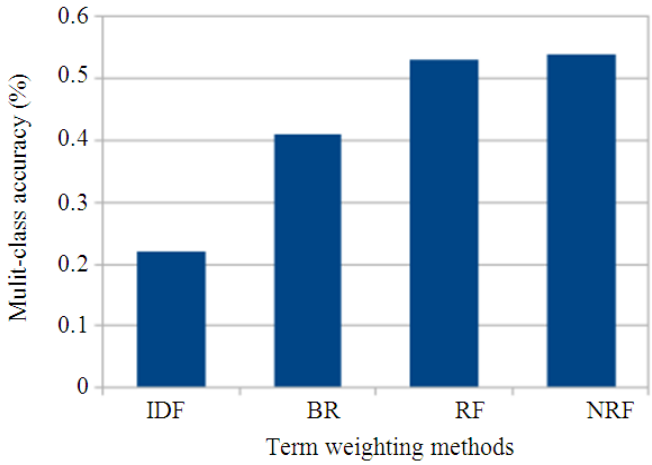

Fig. 1. Comparison of multiclass accuracy for various term weighting methods

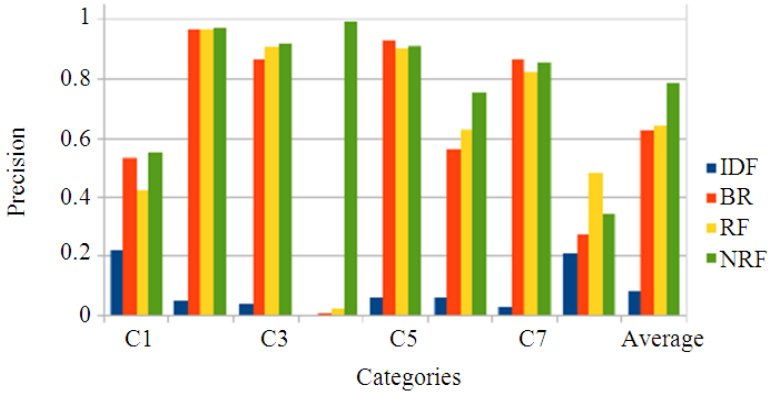

Fig. 2. Comparison of multiclass precision for various term weighting methods

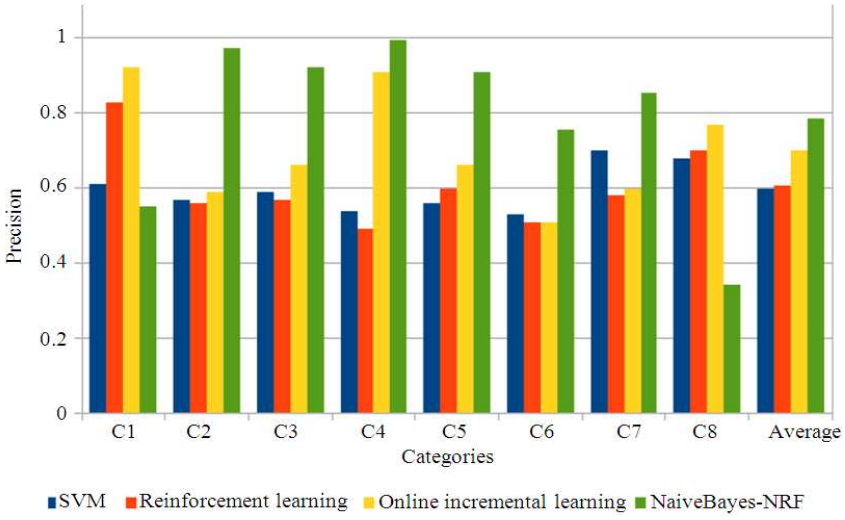

Fig. 3. Performance of multiclass individual classifiers Vs existing binary classifiers

Table 1. Comparison of proposed multiclass classification performance with the existing methods

	Kan and Thi (2005)	Baykan *et al*. (2011)	Proposed
F1	0.368	0.414	0.54

With the proposed new relevance factor based term weighting method, we are able to achieve 22% improvement in terms of F1, which is significant improvement over the existing methods. Also we achieve this result with the simple Naive Bayes Classifier that does not require much training time as other classifiers like SVM.

6. CONCLUSION

We have proposed a web page classification system based on URLs alone. Using this approach, web pages can be classified without downloading the contents of the page, thereby increasing the speed of classification and also avoiding the unnecessary wastage of bandwidth. Even though URLs have very less information, we exploited it fully and classified them by extracting features only from URLs. We have proposed a new relevance factor and have suggested a supervised term weighting method for accurately selecting the relevant features for URL classification with Naive Bayes Classifier. With our proposed method, we have achieved an improvement of 11% in precision over the existing method by reducing false positives greatly. Also, we were able to achieve 22% improvement in terms of F1 for the multiclass classification. This method is helpful for information filtering and focused crawling where high precision is required. By combining other techniques of feature selection, the proposed system's performance can further be improved.

7. ACKNOWLEDGEMENT

The researchers would like to thank the management of SSN College of Engineering for funding the High Performance Computing Lab (HPC Lab) where this research was carried out. The author expresses sincere thanks to Dr. Chandrabose Aravindan, Professor, Dept. of CSE, SSN College of Engineering for his valuable guidance and motivation.

8. REFERENCES

Baykan, E., M. Henzinger, L. Marian and I. Weber, 2011. A comprehensive study of features and algorithms for URL-based topic classification. ACM Trans. Web. DOI: 10.1145/1993053.1993057

Debole, F. and F. Sebastiani, 2003. Supervised term weighting for automated text categorization. Proceedings of the ACM Symposium on Applied Computing, (SAC '03), ACM, New York, USA., pp: 784-788. DOI: 10.1145/952532.952688

Hernandez, I., C.R. Rivero, D. Ruiz and J.L. Arjona, 2012. An experiment to test URL features for web page classification. Proceedings of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems, (ICP '12), pp: 109-116. DOI: 10.1007/978-3-642-28795-4_13

Jianping, Z., J. Qin and Q. Yan, 2006. The role of URLs in objectionable web content categorization. Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Dec. 18-22, IEEE Xplore Press, Hong Kong, pp: 277-283. DOI: 10.1109/WI.2006.170

Kan, M.Y. and H.O.N. Thi, 2005. Fast webpage classification using URL features. Proceedings of the 14th ACM International Conference on Information and Knowledge Management, (CIK '05), ACM, pp: 325-326. DOI: 10.1145/1099554.1099649

Kan, M.Y., 2004. Web page classification without the web page. Proceedings of the 13th International World Wide Web Conference on Alternate track papers and posters, (WWW '04), ACM, New York, USA, pp: 262-263. DOI: 10.1145/1013367.1013426

Lan, M., C.L. Tan, J. Su and Y. Lu, 2009. Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans. Patt. Anal. Machine Intell., 31: 721-735. DOI: 10.1109/TPAMI.2008.110

Rajalakshmi, R. and C. Aravindan, 2011. Naive bayes approach for website classification. Commun. Comput. Inform. Sci., 147: 323-326. DOI: 10.1007/978-3-642-20573-6_55

Rajalakshmi, R. and C. Aravindan, 2013. Web page classification using n-gram based URL features. Proceedings of the 15th International Conference on Advanced Computing, Dec. 18-20, IEEE Xplore Press, Chennai, India.

Rose, S.L. and K.R. Chandran, 2012. Normalized web distance based web query classification. J. Comput. Sci., 8: 804-808.

Singh, N., H. Sandhawalia, N. Monet, H. Poirier and J.M. Coursimault, 2012. Large scale URL-based classification using online incremental learning. Proceedings of the 11th International Conference on Machine Learning and Applications, Dec. 12-15, IEEE Xplore Press, Boca Raton, FL., pp: 402-409. DOI: 10.1109/ICMLA.2012.199