Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 22, Sayı 2, 583-594, 2018

Süleyman Demirel University
Journal of Natural and Applied Sciences
Volume 22, Issue 2, 583-594, 2018

DOI: 10.19113/sdufbed.40209

Performance of Using Tag-based Feature Sets in Web Page Classification

Havva Esin ÜNAL[1], Selma Ayşe ÖZEL*[2], İlker ÜNAL[3]

[1]Çukurova University, Department of Informatics, 01330, Adana
[2]Çukurova University, Faculty of Engineering, Department of Computer Engineering, 01330, Adana
[3]Çukurova University, Faculty of Medicine, Department of Biostatistics, 01330, Adana

(Alınış / Received: 06.11.2017, Kabul / Accepted: 08.06.2018, Online Yayınlanma / Published Online: 05.07.2018)

Keywords
Web mining,
Classification,
HTML tags,
Feature extraction

Abstract: As the Web is a large collection of data growing daily, an automatic Web page classification mechanism is needed to effectively reach to useful information. Majority of the Web pages are in the form of HTML documents, therefore the aim of this study is to explore the effect of HTML tags on classification process, and try to determine the most valuable HTML tags for feature extraction of the classification task. To achieve this goal, we employ 13 different datasets, and use 5 popular classifiers that are SVM, naïve bayes (NB), kNN, C4.5, and OneR. The statistical analysis shows that, the features extracted by using solely the anchor, <p> or <title> tags can be used as an alternative to the features extracted from the whole Web page. SVM is the best among the classifiers used in this study. Using the HTML tags for feature extraction improves classification accuracy.

Web Sayfası Sınıflamada Etiket-tabanlı Nitelik Kümesi Kullanımının Performansı

Anahtar Kelimeler
Web madenciliği,
Sınıflama,
HTML etiketleri,
Nitelik çıkarımı

Özet: Web sürekli büyüyen geniş bir veri kümesidir. Buna bağlı olarak yararlı bilgilere etkili bir şekilde erişmek için otomatik bir Web sayfası sınıflandırma mekanizmasına ihtiyaç duyulmaktadır. Web sayfalarının çoğunluğu HTML dokümanları biçimindedir. Bu nedenle bu çalışmanın amacı, HTML etiketlerinin sınıflandırma işlemi üzerindeki etkisini araştırmak ve sınıflandırmanın nitelik çıkarımı aşamasında kullanılabilecek en etkili HTML etiketlerini belirlemektir. Bu amaca ulaşmak için, 13 farklı veri seti ve 5 popüler sınıflayıcı (SVM, Naive Bayes, kNN, C4.5 ve OneR) kullanılmıştır. İstatistiksel analiz sonuçları, "anchor","<p>" ve"<title>" etiketlerini kullanarak çıkarılan niteliklerin, tüm Web sayfası kullanılarak çıkarılan niteliklere alternatif olarak kullanılabileceğini göstermektedir. SVM, bu çalışmada kullanılan sınıflandırıcılar arasında en başarılısıdır. Nitelik çıkarımı için HTML etiketlerini kullanmak sınıflandırma doğruluğunu arttırmıştır.

1. Introduction

The Web is a large collection of documents of various kinds. Many people use the Internet to find and gather information on certain topics. However, it is not easy to reach to a desired information by using the standard search engines. Possible reasons for this problem are [1];

1. The Web pages are increasing exponentially, hence, it is difficult to keep the index of search engines up-to-date.
2. When a user seeks information on a search engine, too many irrelevant pages containing search terms are presented.

In order to overcome these search problems, accurate classifiers which can assign correct class labels to Web pages are needed [2].

Nowadays most of the Web pages are written in HTML (Hyper Text Markup Language) which consists of tags indicating the structure of texts. Those pages not only include plain texts but also hyperlinks and multimedia information (i.e., images, animations, sounds). Because of this complex structure, the Web page classification confronts more difficulties and more challenges than the text classification [3]. In this study our aim is to investigate the effects of using HTML tags on classification performance of Web pages. Majority of the previous studies that have been done for the Web page classification have ignored

* Corresponding author: saozel@cu.edu.tr

HTML tags and tried to solve this problem as a plain text classification problem. However, only some of the studies [1, 2, 4 – 16, 24 - 29] have used feature extraction methods which involve HTML tags. Although HTML tags are considered during feature extraction, none of the previous studies have made an extensive analysis on the effect of each HTML tag separately.

There are in principle three kinds of HTML tags: logical, physical, and meta-tags [4]. The physical tags are related to the formatting of the text, such as, bold or italic; the logical tags have richer semantic imports like, headlines or anchors; and the meta-tags give information about a document [4, 5]. Thus, as a whole, these tags provide information about the content of a document. Unfortunately, the HTML tags are usually omitted in many researches [17, 18, 19]. Those studies count only the frequencies of terms in Web pages, without making any distinction with respect to the HTML tags and this feature extraction approach is called as "bag-of-words" [6].

In this work, we use only logical and physical tags and omit meta-tags, because in majority of Web pages, meta-tags often include terms that are not related to the content of the Web page to increase ranking score assigned by the search engines or they are left empty [20, 21]. We focus on the text content of Web pages, and do not consider hyperlink structure and multimedia information. We extract features only from a set of logical and physical HTML tags by using each stemmed term with its associated HTML tag as a feature, therefore identical terms in different tags are deemed as different terms. This kind of feature extraction algorithm is named as "tagged-terms" method in [6]. In this study, by using the tagged-terms method, we investigate the effect of each tag on the Web page classification accuracy. We compare the performance of the Naïve Bayes (NB), decision tree (C4.5), k-nearest neighbour (kNN), rule based (OneR), and support vector machine (SVM) classifiers on different feature sets that are extracted by using different HTML tags; and repeat experiments on different datasets to find out the effects of different HTML tags on classification accuracies.

Web page classification/categorization is the process of assigning a class label to the Web pages from a set of predefined categories [22]. Web page classification is a kind of text classification task however; it has been demonstrated in many studies that, using the information derived from HTML tags can increase the classifier's accuracy. In an early study, Golub and Ardo [7] determined the significance of different parts of a Web page for automated classification. They used four elements of a Web page: title, headings, metadata, and main text. The experimental analysis showed that using all of these elements is necessary for automated Web page classification since, only some of these elements occur at the same time on Web pages.

Later, Ru and Horowitz [23] presented a method for automated classification of HTML forms. Algorithms have been developed for automatic feature generation from HTML forms and a neural network has been applied for classification. For the feature extraction <form> tag is used and high classification accuracy is observed.

Another study that involves HTML tags for Web page classification belongs to Yang, Slattery, and Ghani [8] who have concluded that the HTML tags in hypertext pages improve classification performance when considered jointly with the text contained in the Web pages. In [24], it is demonstrated that, SVM classifier using the text on the target page, page title, and anchor text from parent pages can improve classification compared with a pure text classifier.

Fresno, Martinez, Montalvo and Casillas [9] have proposed a NB based Web page classification system which uses HTML mark-up information to find the term relevance in a Web page. The experiments showed that, gaussian models give better accuracy than event models when enriched representations are considered.

According to [2], a new feature set, which is the hierarchical structure of headings appearing in the Web page, enhances the classification performance. The weights for the words appearing in the heading tags are assigned related to their hierarchy. As a result, it has been found that the hierarchical structure of headings has a high impact and could improve the classification performance.

Kim and Zhang [5] proposed a method to learn the internal structure of HTML documents by using genetic algorithms. The proposed algorithm learns the important factors of the HTML tags which are then used to re-rank the documents retrieved by standard weighting schemes. The results indicate that the proposed approach significantly improves the performance of retrieval accuracy.

Xue, Bao, Huang and Lu [3] studied several key aspects of the SVM for Web page classification. For feature extraction, a set of commonly used features of Web pages, such as body, title, headings, and meta-tags are used. They have concluded that composite of plain text and HTML structure gives better classification performance.

Werner, Böttcher and Beckmann [4] presented an approach which uses the HTML tags to improve the quality of the classification. The developed classification system uses changes in the typographical style of an HTML document. Therefore, one can detect the parts of the document that is emphasized by the HTML page developer. These

emphasized parts are weighted stronger, which leads to significant improvement on the classification of documents.

In another study [6], a genetic algorithm (GA) based Web page classification system has been developed which uses both the HTML tags and stemmed terms belong to each tag as features for classification. The proposed system learns the best weights for each feature by the GA, and the experimental evaluation showed that, using the HTML tagged-terms as features increases the classification accuracy with respect to using terms alone.

Belmouhcine et. al [10] proposed an approach which classifies Web pages by using plain text and text between the HTML tags. In the first step of the method an SVM implementation is used to generate a reduced vector representation based on plain text and text from the HTML tags. Then in the second step, the NB algorithm is used to determine the class of the Web page. The experiments showed that, using the combination of HTML tags with plain text increases the performance of NB classifier.

Saraç and Özel [11] used firefly algorithm in order to find the best features for Web page classification. The features are extracted from URL and <title> tag, and the Web pages are classified without loss of accuracy. In another study of Saraç and Özel [12], ant colony optimization algorithm has been applied to reduce the number of features used for Web page classification. After the experimental evaluations it is concluded that, using the URL and <title> tags for feature extraction gives a good classification performance with respect to that of using the bag-of-words method.

In [13], Meshkizadeh and Rahmani illustrated that using the HTML tags and URL features of a Web page along with features of sibling pages, and NB as a classifier, could increase the classification accuracy. Jeong et al. [14] developed a method for extracting the title of a Web page by using anchor tags. They verified that by using anchor tag information, the accuracy of the classifier increases.

Bhalla and Kumar [24] employed HTML tags to extract features from Web pages and applied SVM for classification. Experimental evaluation showed that the tag based feature extraction gives satisfactory performance.

Navadiay, Parikh and Patel [25] focused on the Web page classification based on a combination of the content and the structure of a Web page. They used the same feature extraction algorithm as that used by Özel [6]. The results indicated that the NB is good for Web page classification when combination of HTML tag and term is used as features.

In [26], Sarhan, Hamissa and Elbehiry proposed 2 algorithms which they called “Important HTML tags only algorithm” and “Weighted Important HTML tags only algorithm”. They compared these algorithms with the traditional feature selection algorithm (i.e. using bag-of-words). They used two famous classifiers SVM and NB to classify the Web pages by using the features selected by employing these algorithms. As a result, they showed that using the proposed algorithms improves the accuracy of the classifiers.

In our recent study [15] we used 6 HTML tag sets in the tagged-terms feature extraction method and performed experiments on 9 different datasets using 4 classifiers. We concluded that C4.5 and kNN classifiers perform better when tags are used for feature extraction. However in this study we employ 13 datasets with 4-folds cross validation, 5 classifiers are applied, 8 different feature extraction methods are compared, and results are analyzed statistically in more detail. As we perform more experiments with respect to the previous study, results obtained in this study is more general and reliable.

In a more recent study [27], Thanasopon et. al focused on text mining and they aimed to detect the most popular online trends. While extracting the topics, they used TF-IDF and HTML score. They assumed that words in certain tags are more related to the main concept than the others. For this purpose, weight of words in these tags such as <h1> and <b> are increased. By using this term extraction method, they conducted experiments on a popular discussion forum and concluded that SVM classifier outperformed other classifiers.

As summarized above we have evaluated most of the previous studies related to Web page classification that involve HTML tags. Although physical HTML tags are generally used to form the appearance of text on a browser, they provide important clues about the topic, theme, and genre of the Web page as shown in the previous studies. Therefore, utilizing HTML tags for classification of Web pages improves accuracy of the classifiers as proved in the previous studies [4, 6, 13, 26]. However, except our recent study, none of the previous studies have made a comparison among HTML tags to use in feature extraction. In this study, our aim is to make an extensive experimental evaluation on the effects of each HTML tag over Web page classification and try to determine which HTML tag(s) should be considered for feature extraction. To reach our goals we use 13 distinct datasets, whereas the other studies have used only a few datasets. We investigate the effects of each <title>, <h1>, <h2>, <h3>, <a href=…>, , , <b>, <i>, <p>, and <li> tags and compare them with the traditional bag-of-words and tagged-terms methods. We repeat our experiments with five classifiers that are SVM, NB,

C4.5, kNN, and OneR to also show the combined effects of classifiers and HTML tags, however previous studies have used only a few classifiers. We perform statistical analysis to determine best methods. To our knowledge, no one has applied such statistical methods in their studies. Therefore our study will be helpful to researchers and practitioners who work in the area of Web page classification and information extraction from Web pages by indicating which HTML tags can give more valuable features for classification, which classifier performs better, and the interactions between feature extraction methods and classifiers.

The rest of the paper is organized as follows: In the following section, we describe our feature extraction method, the datasets used in the experiments, and the evaluation metrics. Section 3 presents the experimental results and discussions on them. Finally, Section 4 gives conclusions and some future works that we plan to perform.

2. Material and Method

The block diagram of the applied methods in this study is presented in Figure 1. As shown in the figure, each dataset used in this study is first partitioned as train and test sets. We use training dataset to extract features and learn a classification model. The extracted features are then used to compute document vectors for each Web page in the training and the test sets. After these steps, the documents in the test set are assigned class labels by using the learned classifier. Finally, accuracy of the classification task is computed. These steps are repeated for each feature extraction method, classifier, and dataset four times as we apply 4-fold cross validation. After that we apply statistical analysis to show the effects of using HTML tags in classification. The details of each method and datasets used in this study are explained in detail in the following subsections.

2.1. Datasets

In this study we make binary classification as it is used by many focused crawlers to improve search performance of search engines. Binary classification tries to determine whether a Web page is in the class of interest or not. Therefore we prepare 13 binary classification datasets from the publicly available WebKB, Benchmark, Syskill Webert datasets, as well as manually collected Conference dataset. We apply 4 fold cross validation. The number of instances for the first fold for each dataset is listed in Table 1. These numbers are very similar for the other folds and to save space they are not listed in this paper. The details of each dataset are given in the below subsections.

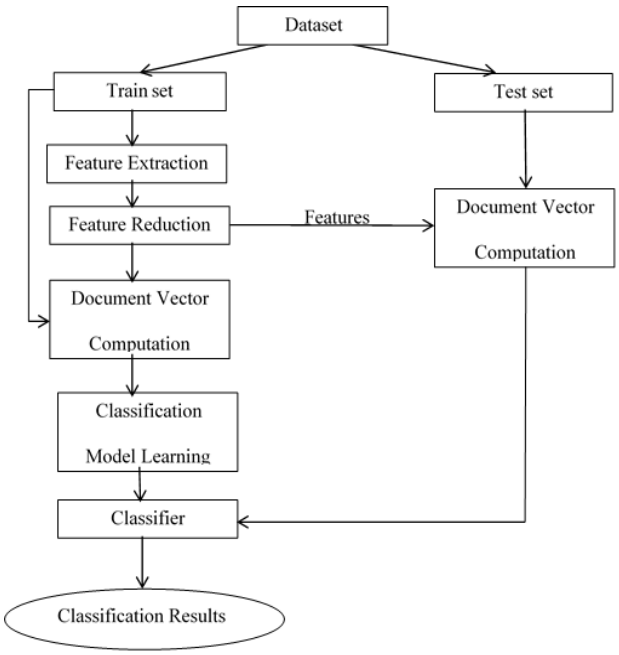


Figure 1. Block diagram of the applied methods

Table 1. Number of documents in the datasets.

Dataset	Class	Train	Test	Total
Conference	Conference	618	206	2369
	Not Conference	1159	386	
Course	Course	869	61	4694
	Not Course	2822	942	
Faculty	Faculty	1087	38	4889
	Not Faculty	2822	942	
Project	Project	482	22	4268
	Not Project	2822	942	
Student	Student	1500	140	5404
	Not Student	2822	942	
Biology	Biology	750	250	4500
	Not Biology	2625	875	
Commercial Banks	CBanks	750	250	4500
	Not CBanks	2625	875	
Programs	C/C++	750	250	4500
	Not C/C++	2625	875	
Motor Sport	MSport	750	250	4500
	Not MSport	2625	875	
Bands	Bands	46	15	327
	Not Bands	199	67	
Biomedical	BioMed	98	33	327
	Not BioMed	147	49	
Goats	Goats	53	18	328
	Not Goats	192	64	
Sheep	Sheep	49	16	327
	Not Sheep	197	65	

2.1.1. Conference dataset

The Conference dataset consists of the Computer Science related conference homepages. This dataset is manually collected and used in [6, 28]. The names of the conferences in the dataset are obtained from the DBLP Computer Science Bibliography (http://www.informatik.uni-trier.de/~ley/db/) and then these names are queried by using the Google search engine (http://www.google.com). The

conference homepages in the query results are labelled as positive documents; and the pages that include similar information with conference homepages but are irrelevant as negative documents. Then, all the positive and negative documents are randomly distributed among training and test sets. The Conference dataset contains 2369 Web pages in total (824 positive, 1545 negative documents).

2.1.2. WebKB dataset

WebKB dataset was prepared by the WebKB project at CMU [29]. The dataset consists of Web pages collected from Cornell, Texas, Washington, and Wisconsin Universities; and the pages are classified into seven categories. We use a subset of the WebKB dataset (i.e., only the student, faculty, course, and project category pages) because these categories have more instances than the remaining. For each category, we generate a binary classification dataset, therefore we obtained Course, Student, Faculty, and Project datasets. For each dataset, we use "others" category of WebKB dataset as negative class instances. As an example the Course dataset contains Computer Science related course homepages and some irrelevant Web pages from the "others" category of WebKB and has 4694 Web documents in total. 4-fold cross validation is applied as described in the WebKB project Web site [30].

2.1.3. Benchmark dataset

The Benchmark [31] is a dataset of 11,000 Web documents pre-classified into 11 equally-sized categories, each containing 1,000 Web documents. It was generated by Sinka and Corne, with the main aim of proposing a general dataset for Web document clustering and similar experiments. The Benchmark dataset consists of four main themes namely "Banking & Finance", "Programming Languages", "Science", and "Sport". From each theme, we chose one class. These are "Commercial Banks", "C/C++", "Biology", and "Motor Sport". Negative pages are selected randomly from the rest of the seven classes. Therefore we obtain "Commercial Banks", "Programs", "Biology", and "Motor Sport" datasets, each containing 4500 documents in total. Then, we apply 4-fold cross validation.

2.1.4. SyskillWebert dataset

SyskillWebert dataset [32] has a similar structure with WebKB dataset. It contains HTML source of Web pages. The Web pages are on four separate subjects that are Bands (recording artists), Goats, Sheep, and Biomedical. All of the four subjects are involved in our study and 4-fold cross validation is applied.

2. 2. Proposed feature extraction methods

As our aim is to evaluate the effect of each HTML tag on the performance of Web page classification and to determine which HTML tag covers valuable features, we use the terms that are surrounded by HTML tags as features, and propose 8 feature extraction methods.

We use <title>, <h1>, <h2>, <h3>, <a href=…>, , , <b>, <i>, <p>, and <li> HTML tags, as well as the text content to extract features. We choose these tags because in [1, 2, 4 – 10, 13, 14] it is observed that these tags include the most useful information. We group some of the related tags given above in order to reduce the feature space. The tags <h1>, <h2>, <h3> are grouped together as "header"; <b>, , <i>, are grouped as "bold"; <p> and text content are grouped as "text" features. We take <a href= >, <li>, and <title> tags separately and call them as "anchor", "list", and "title" features, respectively. Therefore we have 6 HTML tags (or tag groups) that are used for feature extraction. For each of these HTML tags or tag groups, all the terms that belong to each tag or tag group are taken; the stopwords are removed from the extracted terms; Porter's stemming algorithm [33] is applied; and each stemmed term for each tag or tag group forms a feature. Therefore, we collect *anchor, bold, header, title, list,* and *text* feature sets for each dataset; and use each feature sets separately.

In the seventh feature extraction method, we also use these term-tag pairs to form another feature set named as tagged-terms. Apart from using each tag group alone, we use all the terms from all the tag groups such that a term can be in the feature list several times because every term is used with its corresponding tag (i.e., the word "course" in the <title>, <li>, and <b> tags are considered as different features).

Finally, we use the bag-of-words method to form a different set of features. In the bag-of-words method all the HTML tags are removed and the remaining pure text is used. In this method there is no distinction between the words with respect to HTML tags. As in the tagged-terms feature extraction method the stopwords are removed, and the remaining terms are stemmed according to Porter's stemming algorithm [33].

All of the above mentioned feature extraction methods are applied to the positive instances in the training part of each dataset. As most of the datasets used are not balanced, and we have higher number of negative instances, extracting features from positive instances give higher accuracy and produces lower number of features as we observed in our previous study [6]. After extracting features as described above, document vectors for the training and the test sets are created by using the term frequencies in the associated Web page. Then these document vectors are normalized according to the document lengths.

Table 2. Number of features after reduction for each dataset

Dataset	Anchor	Bold	Header	List	Title	Text	Tagged-terms	Bag-of-words
Conference	330	184	66	404	20	1120	2123	1261
Course	252	97	111	389	30	906	1785	922
Faculty	212	89	62	287	16	889	1554	905
Project	272	84	72	261	25	1057	1771	1069
Student	216	52	48	194	5	682	1197	698
Biology	807	603	47	847	37	4914	7256	5178
CommercialBanks	289	155	14	304	51	989	1801	1167
MotorSport	616	386	37	44	65	1523	2670	1845
Programs	676	346	63	452	47	1957	3541	2181
Bands	95	852	151	296	82	4409	5885	4422
Biomedical	789	327	170	707	56	1531	3580	1546
Goats	465	387	225	703	115	3927	5821	3976
Sheep	372	422	178	723	116	2377	4189	2420

2.3 Feature reduction

The number of features obtained by using the proposed methods is very large for some datasets. As an example we extract approximately 50000 features for the Student dataset when tagged terms feature extraction is applied. Therefore we apply document frequency filtering to reduce the feature space. According to Salton [34], the most useful terms are the ones having document frequencies between 1% and 10% due to the fact that low document frequency terms are generally misspelled ones, and high document frequency terms are often stopwords. For this reason we removed features having document frequency less than 2% for each dataset to eliminate misspelled terms. We determined this threshold experimentally.

The numbers of features obtained by each feature extraction method after the document frequency filtering is applied are presented in Table 2. As an example, the "Title" column in Table 2 gives the number of features extracted only from <title> tags of the Web documents in the training set of each dataset having document frequency greater than 2%. The values given in the table belong to the first fold of each datasets. The numbers of features obtained for the other folds are similar, and to save space, they are not included in this paper.

As our aim is to measure the effect of each tag separately, we classify each dataset by using the features extracted from the above mentioned 8 feature extraction methods, and compare the results.

2.4 Classifiers

In our study, five different classifiers, namely Naïve Bayes, decision tree, k-nearest neighbor (kNN), rule based, and support vector machine (SVM) are used to show the effect of the HTML tags. For implementation, we use the WEKA-package [35]. As Naïve Bayes classifier we employ Naïve Bayes Multinomial (NBM) since it performs better than ordinary Naïve Bayes model for document classification [36]. We use LibSVM package for SVM classifier, choose linear kernel as we have high dimensional feature space [37], and used the default parameters for each dataset.

For decision tree classifier, we apply J48 which is an implementation of C4.5 algorithm. For kNN, we employ IBk with k=1 for each dataset; and finally for rule based classifier we use OneR from the WEKA package. Among the used classifiers NB and SVM have been used in the majority of the Web page classification studies such as [3, 15, 19]. In [6, 11, 12, 28] it has been showed that kNN, decision tree, and rule based classifiers also have high accuracy for Web page classification. Therefore we include all these five classifiers in our study.

2.5. Evaluation metric

In our experiments the F-measure, which is commonly used metric [1, 2, 6, 38, 39], is employed for performance evaluation. The F-measure is a harmonic mean of the precision and the recall of the test and it is defined as:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1)$$

where, precision and recall are computed as;

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (2)$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (3)$$

Given two classes, positive documents are the documents of the main class of interest (e.g., class C1), and negative documents are the documents that do not belong to the main class of interest (e.g., Not C1). According to these definitions "TruePositives" means the positive documents that are correctly labelled by the classifier, and "FalsePositives" ("FalseNegatives") are the negative (positive) documents that are incorrectly labelled [39].

Table 3. Average running time for Conference dataset.

Feature Extraction Method	# of Features	IBk	J48	NBM	OneR	LibSVM
Anchor (<a href=…>)	330	2sec.	<1sec.	<1sec.	<1sec.	1sec.
Bold (,< b>, <i>,)	184	1sec.	<1sec.	<1sec.	<1sec.	1sec.
Header (<h1>, <h2>, <h3>)	66	<1sec.	<1sec.	<1sec.	<1sec.	<1sec.
Title (<title>)	20	<1sec.	<1sec.	<1sec.	<1sec.	<1sec.
List item ( )	404	3sec.	1sec.	<1sec.	<1sec.	1sec.
Text (<p>, text content)	1120	8sec.	2sec.	1sec.	<1sec.	2sec.
Bag-of-words (BW)	1261	8sec.	2sec.	1sec.	<1sec.	2sec.
Tagged-terms (TT)	2123	15sec.	4sec.	1sec.	1sec.	3sec.

2.6. Statistical analysis

The statistical analyses are performed by using SPSS. The F-measure values for the methods are summarized as mean and standard deviation. Repeated Measurement analysis is used for comparing the F-measure values of the methods. To assess the effect of using the HTML tags on classification accuracy, analysis of variance (ANOVA) is used. The well-known Bonferroni test is applied for pairwise comparisons. $p<0.05$ is accepted as statistically significant.

3. Results

In this section, experimental results that include the effects of feature extraction methods and classifiers are presented. From the statistical analyses applied, we try to conclude which classifier and feature extraction method have the best performance for each dataset in specific and for all datasets in general.

3.1. Time to build and test the classification models

We measure the total time required to train and test the classification models for each dataset. Table 3 gives the average running time of 4-folds for the Conference dataset. Experiments were done on a hardware which has 8 GB of RAM and Intel® Core™ i7-2600 3.80GHz processor. To save space, average running times for only one dataset are presented in this subsection. Similar trends were observed for the remaining datasets.

As seen in Table 3 the running times change depending on the number of features and classifiers used. As it is expected, when one employs a feature extraction that yields small number of features, running time decreases sharply. We should also point out that among the classifiers we have tested, IBk is the slowest classifier since it is a lazy method.

3.2. F-Measure values of classifiers for each dataset

For each feature extraction and classification methods, average F-measure values for 4-fold cross validation on each dataset are given in Figure 2, where the x-axis shows the feature extraction methods, and y-axis gives the F-measure values.

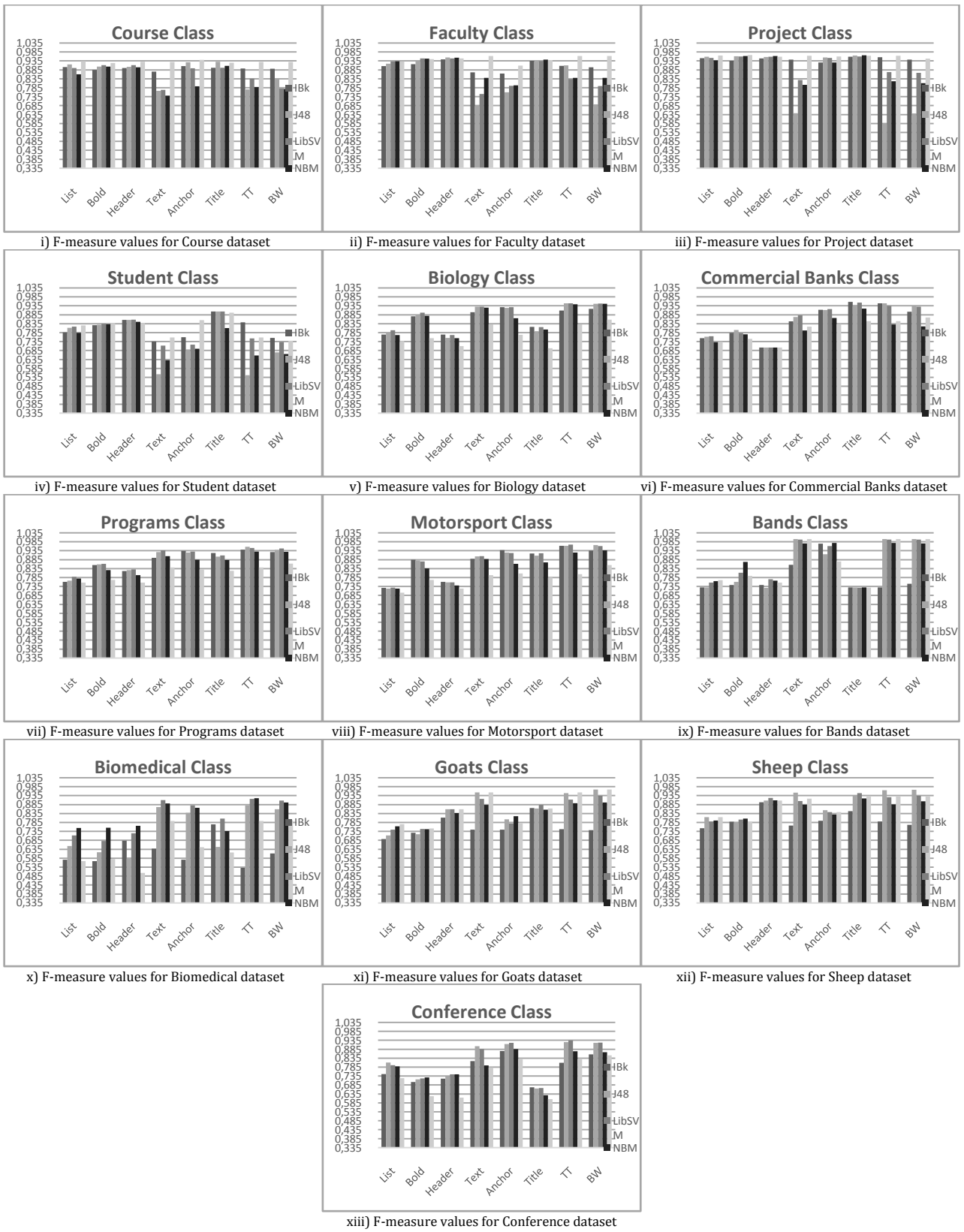
As shown in Figure 2 (“i” through “xiii”), the best feature extraction method and the best classifier can change for different datasets, however using anchor, title, text, bag-of-words and tagged-terms feature extraction methods produce the best classification performance as shown by the statistical analyses given in the consequent subsections.

3.3. Comparison of classifiers

When the mean of all the F-measure values obtained from different feature sets are taken into consideration, the average F-measure values for the classifiers can be calculated. The results are given in Table 4. Based on the p value given in Table 4, there is a significant difference among the classifiers ($p<0.001$). As a result of pair-wise comparisons between the classifiers; the LibSVM classifier performs better than the IBk, OneR, and NBM classifiers.

We used the NBM classifier since it performs better than NB implementation in WEKA for document classification [36]. The results of our experiments, applying both NB and NBM classifiers of WEKA to all the datasets, have supported that conclusion. We have got an overall of 0.755(±0.127) classification accuracy for the NB classifier, on the other hand as seen in Table 4 it is 0.841(±0.085) for the NBM classifier. This result is also compatible with that of [40] which compares the NB with the SVM for text classification and applies some corrections to improve the performance of the NB classifier. Although corrections applied to the NB classifier had improved its text classification accuracy, the corrected version also had worse classification performance than the SVM [40] as we observe in this experiment.

When we examine the classification accuracy of the rule based classifier (OneR), we observe that it has the best performance with the data obtained from the WebKB dataset (see Table 5), this result occurred due to the fact that in the WebKB dataset, class specific terms like “course”, “student”, “faculty”, “project” occur with the HTML tags as well as in the text, so OneR can easily find these terms and generate rules which involve class specific terms to classify the Web pages. However, as we repeat the experiments for the other datasets, the overall performance of the OneR classifier reduces, and becomes worse than the LibSVM, J48, and NBM.



i) F-measure values for Course dataset
ii) F-measure values for Faculty dataset
iii) F-measure values for Project dataset
iv) F-measure values for Student dataset
v) F-measure values for Biology dataset
vi) F-measure values for Commercial Banks dataset
vii) F-measure values for Programs dataset
viii) F-measure values for Motorsport dataset
ix) F-measure values for Bands dataset
x) F-measure values for Biomedical dataset
xi) F-measure values for Goats dataset
xii) F-measure values for Sheep dataset
xiii) F-measure values for Conference dataset

Figure 2. F-measure values for each feature extraction method and classifier for all datasets.

The J48 has been found to be the second best performer classifier, and this conclusion is compatible with our previous experiments [11, 12, 15, 28], where we had found that the J48 performs better than the NB and IBk.

The IBk has good performance in approximately 50% of the datasets, however, as it is a lazy approach it has high testing time as shown in Table 3, and its overall classification accuracy is not as good as the LibSVM, J48, OneR and NBM.

According to Table 4, one can conclude that the LibSVM with linear kernel and default parameter settings is the best among the classifiers used in the experiments. Moreover, if you apply optimal

parameter settings you may get better results with LibSVM. Alternatively, the J48 can be used instead of the LibSVM as can be seen from Table 4.

On the other hand, when one compares the running times of the classifiers it is seen that the LibSVM has similar running times to that of the J48, and both are extremely faster than the IBk (Table 3).

Table 4. F-measure values of classifiers

Classifier	F-Measure	P
IBk	0.828±0.103	
J48	0.843±0.118	
NBM	0.841±0.085	<0.001
OneR	0.833±0.108	
LibSVM	0.862±0.088	

3.4. Effect of classifiers and feature extraction methods on each dataset

Table 5 summarizes the datasets, and the corresponding classifiers, and the feature extraction methods that give the tabulated best F-Measure values. According to the analysis presented in Table 5, for most of the datasets, the bag-of-words or the tagged-terms methods give the highest F-measure values when used with the LibSVM or the J48 classifiers.

For Course, Project, and Student datasets, using features extracted from the anchor, bold and title tags give the highest F-measure values when used with OneR and IBk classifiers as these tags include class specific terms for these datasets.

Table 5. Datasets, the corresponding classifiers, and feature extraction methods giving the best F-measure values

Dataset	Classifier	Feature Extraction Method*	F-Measure
Course	OneR	Anchor	0.941±0.019
Faculty	OneR	Text, BW, TT	0.963±0.005
Project	OneR	Bold	0.967±0.004
Student	IBk	Title	0.904±0.020
Biology	J48	TT	0.949±0.010
Commercial Banks	IBk	Title	0.957±0.016
Programs	J48	TT	0.957±0.007
Motor Sport	LibSVM	TT	0.970±0.009
Bands	J48,OneR	Text, BW, TT	1.000±0.000
Biomedical	NBM	TT	0.921±0.028
Goats	J48, OneR	BW	0.969±0.024
Sheep	J48	BW	0.968±0.008
Conference	LibSVM	TT	0.935±0.010

TT= Tagged-terms, BW= Bag-of-words

3.5. Effect of feature extraction methods on classification accuracy

The mean of all the F-measure values obtained by using different classifiers when the corresponding feature extraction method is taken into consideration is given in Table 6. The p value given in the table indicates that there are significant differences among the feature extraction methods ($p<0.001$). As a result of pair-wise comparisons between the methods one can conclude that the title, anchor, text, bag-of-words and tagged-terms feature extraction methods perform better than the bold, header, and list feature extraction methods (see Table 6).

Table 6. F-measure values according to feature extraction methods

Feature Extraction Method	F-Measure	P
Anchor (<a href=…>)	0.860±0.087	
Bold (,< b>,<i>,)	0.817±0.096	
Header (<h1>,<h2>,<h3>)	0.811±0.103	
Title (<title>)	0.850±0.103	<0.001
List item ( )	0.793±0.088	
Text (<p>, text content)	0.852±0.100	
Bag-of-words (BW)	0.873±0.096	
Tagged-terms (TT)	0.875±0.108	

Also when numerically reviewed, tagged terms and bag-of-words feature extraction methods are the first and second best performer methods, respectively, as we have observed in our previous studies [6, 11, 12, 28]. Surprisingly, features extracted from anchor tags have better classification accuracy from features extracted from text content, and anchor feature extraction method is the third best performer in terms of classification accuracy.

The feature sets formed by the bag-of-words or the tagged-terms methods have large number of features (see Table 2). Thus, using these features decrease runtime performance of the classifiers (see Table 3). Therefore, to choose tag based feature sets can be more appropriate for large datasets. According to the pairwise comparisons done between the feature extraction methods; the feature sets formed by using only the anchor tag, text tag, or title tag can be an alternative to the feature sets of the bag-of-words, and tagged-terms methods for large datasets.

3.6. Effect of feature extraction methods on classifiers

For the tagged-terms and the bag-of-words methods, the differences between the F-measure values of the IBk and the LibSVM classifiers are statistically significant ($p=0.029$ and $p=0.019$ respectively). If these methods are used in classification, then one will get better classification accuracy from the LibSVM classifier than the IBk classifier (see Figure 3).

For the remaining feature extraction methods, the differences between the F-measure values of the classifiers are not statistically significant. When these feature extraction methods are used, there is no difference in using the IBk, J48, NBM, OneR, or LibSVM from statistical point of view.

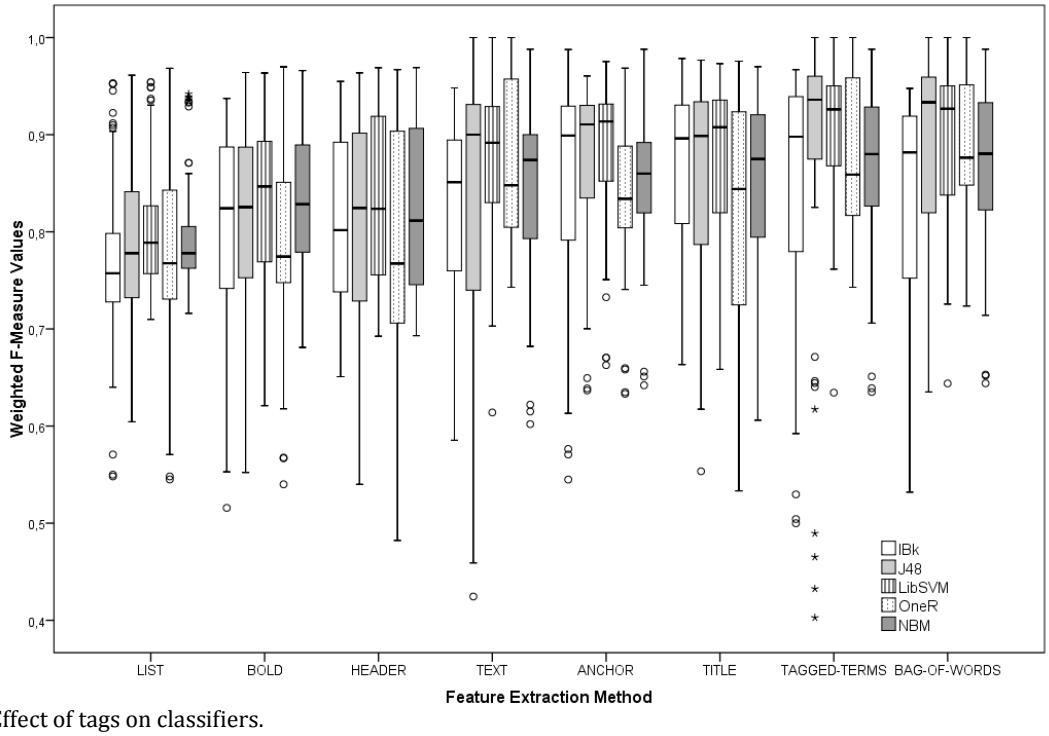


Figure 3. Effect of tags on classifiers.

However, for each feature extraction method one or two classifiers can be chosen numerically according to their F-measure values (see Figure 3). The feature extraction method and the corresponding classifier that best suits the method are given in Table 7.

Table 7. Feature extraction methods and the best corresponding classifiers

Feature Extraction Method	Classifier
Anchor (<a href=…>)	LibSVM
Bold (,< b>, <i>,)	LibSVM
Header (<h1>, <h2>, <h3>)	J48 and LibSVM
Title (<title>)	LibSVM
List item ( )	LibSVM
Text (<p>, text content)	J48
Bag-of-words (BW)	J48
Tagged-terms (TT)	J48

4. Discussion and Conclusion

In this study we used both the HTML tags and the stemmed terms that belong to each tag, and also all the terms from the Web pages as classification features. We performed our experiments on 13 datasets with 8 feature extraction methods and repeated the experiments with 5 different types of classifiers using 4-fold cross validation to explore the effects of using HTML tag based features on classification accuracy. First of all, we compared classification performances of classifiers. When all the F-measure values are taken into consideration, the SVM classifier seems to be the best choice in terms of classification accuracy and time.

The results of the statistical analysis show us that different feature set-classifier couples give higher classification accuracy for different datasets. But, we have also observed that in most of the datasets the bag-of-words or the tagged-terms methods give the highest classification accuracy when used with the SVM or the decision tree classifiers.

According to pair-wise comparisons of the feature sets; the anchor, tagged-terms, title, text, and bag-of-words feature sets perform better than the feature sets formed by using the bold, header, and list tags. The tag-based feature sets (anchor, bold, header, list, title, and text feature sets) have smaller number of features than the tagged-terms and bag-of words feature sets, and thus using these sets improves the runtime performance of the classifiers. According to pair-wise comparisons between the feature sets, using only the anchor tag, text tag or title tag can be an alternative to the bag-of-words and tagged-terms methods.

When the effect of tags on the classification accuracy is examined, it is seen that features extracted by the bag-of-words and the tagged-terms methods give better results mostly with using the SVM classifier than the kNN classifier. On the other hand, there is no difference in using the kNN, decision tree, NB, rule based, or SVM classifiers when the other feature sets are used for feature extraction.

By this study, apart from the works done on web page classification [1, 3, 9, 17, 18, 19, 23], we have the chance to emphasize which tag gives better performance when used in a feature extraction method. Our results are compatible with the studies in which the HTML tags have been used and tried to show the impact of them [2, 4 – 8, 10 – 15, 24, 25, 28, 29]. Our study has also proven the positive impact of using the HTML tags on classification accuracy. The title, anchor, text, and tagged-terms feature sets give

better performance in many cases than the bag-of-words feature set.

As a future work, we plan to examine the combined effects of the HTML tag sets as a comparison to the results of this study. Furthermore, the experiments may be repeated for multi-class classification, and some other classifiers like Random Forests, and Maximum Entropy may be applied.

References

[1] Shaker, M., Ibrahim, H., Mustapha, A. and Abdullah, L. N. 2009. Information Extraction From Hypertext Mark-up Language Web Pages. Journal of Computer Science, 5(8), 596-607.

[2] Soonthomphisaj, N., Chartbanchachai, P., Pratheeptham, T. and Kijsirikul, B. 2002. Web Page Categorization Using Hierarchical Headings Structure. Proceedings of the 24th International Conference on Information Technology Interfaces in Cavtat, Croatia, IEEE, 37-42.

[3] Xue, W., Bao, H., Huang, W. and Lu, Y. 2006. Web Page Classification Based on SVM. Proceedings of the 6th World Congress on Intelligent Control and Automation in Dalian, China, IEEE, 6111-6114.

[4] Werner, L., Böttcher, S. and Beckmann, R. 2005. Enhanced Information Retrieval by Using HTML Tags. Proceedings of the 2005 International Conference on Data Mining in Las Vegas, Nevada, USA, CSREA Press, 24-29.

[5] Kim, S. and Zhang, B.-T. 2003. Genetic Mining of HTML Structures for Effective Web-document Retrieval. Applied Intelligence, 18(3), 243–256.

[6] Özel, S. A. 2011. A Web Page Classification System Based on a Genetic Algorithm Using Tagged-terms as Features. Expert Systems with Applications, 38(4), 3407-3415.

[7] Golub, K. and Ardo, A. 2005. Importance of HTML structural elements and metadata in automated subject classification. Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries in Vienna, Austria, Springer-Verlag, 368–378.

[8] Yang, Y., Slattery, S. and Ghani, R. 2002. A Study of Approaches to Hypertext Categorization. Journal of Intelligent Information Systems, 18(2-3), 219–241.

[9] Fresno, V., Martinez, R., Montalvo, S. and Casillas, A. 2006. Naive Bayes Web Page Classication with HTML Mark-up Enrichment. Proceedings of the International Multi-Conference on Computing in the Global Information Technology in Bucharest, Romania, IEEE, 48-53.

[10] Belmouhcine, A., Idrissi, A. and Benkhalifa, M. 2013. Web Classification Approach Using Reduced Vector Representation Model Based on HTML Tags. Journal of Theoratical and Applied Information Technology, Vol.55 No.1, 137-148.

[11] Saraç, E. and Özel, S. A. 2013. Web Page Classification Using Firefly Optimization. 2013 IEEE International Symposium on INnovations in Intelligent SysTems and Applications in Albena, Bulgaria, IEEE, 1-5.

[12] Saraç, E. and Özel, S. A. 2014. An Ant Colony Optimization Based Feature Selection for Web Page Classification. The Scientific World Journal, Vol. 2014, Article ID 649260 (2014), 16 pages.

[13] Meshkizadeh, S. and Rahmani, A. M. 2010. Webpage Classification Based on Compound of Using HTML Features & URL Features and Features of Sibling Pages. International Journal of Advencements in Computing Technology, 2(4), 36-46.

[14] Jeong, O., Oh, J., Kim, D., Lyu, H. and Kim, W. 2014. Determining the Titles of Web Pages Using Anchor Text and Link Analysis. Expert Systems with Applications, Vol. 41 No. 9 (2014), 4322-4329.

[15] Ünal, H. E., Özel, S. A. and Ünal, İ. 2013. Effect of Tagged-Terms on Web Page Classification Accuracy. Global Journal on Technology, Vol. 3 (2013), 244-250.

[16] Bhalla, V.K. and Kumar, N. 2016. An Efficient Scheme for Automatic Web Pages Categorizaiton Using the Support Vector Machine. New Review of Hypermedia and Multimedia, Vol. 22 No:3 (2016), 223-242.

[17] Ester, M., Kriegel, H.-P. and Schubert, M. 2002. Web Site Mining: A New Way to Spot Competitors, Customers and Suppliers in the World Wide Web. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining in Edmonton, CA, USA, ACM Press, 249-258.

[18] Qi, D. and Sun, B. 2004. A Genetic k-means Approaches for Automated Web Page Classification. Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration in Las Vegas, Nevada, USA, IEEE, 241–246.

[19] Bie, R., Fu, Z., Sun, Q. and Chen, C. 2010. A Comparison Study of Bayesian Classifiers on Web pages classification. New Generation Computing, 28(2), 161-168.

[20] Davison, B. D. 2000. Topical Locality in the Web, Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval in Athens, Greece, ACM Press, 272-279.

[21] Pierre, J. M. 2001. On the Automated Classification of Web Sites. Linköping Electronic Articles in Computer and Information Science, Vol. 6 (2001), arXiv preprint cs/0102002.

[22] Qi, X. and Davison, B. D. 2009. Web Page Classification: Features and Algorithms. ACM Computing Surveys, 41(2), Article 12.

[23] Ru, Y. and Horowitz, E. 2007. Automated Classification of HTML Forms on E-commerce Web Sites. Online Information Review, Vol. 31 No. 4 (2007), 451 - 466.

[24] Sun, A., Lim, E.-P. and Ng, W.-K. 2002. Web Classification Using Support Vector Machine. Proceedings of the 4th International Workshop on Web Information and Data Management in New York, USA, ACM Press, 96–99.

[25] Navadiay, D., Parikh, M. and Patel, R. 2013. Constructure Based Web Page Classification. International Journal of Computer Science and Management Research, 2(6), 2742-2746.

[26] A. M. Sarhan, G. M. Hamissa and H. E. Elbehiry, 2015. Feature Selection Algorithms Based on HTML Tags Importance. 2015 Tenth International Conference on Computer Engineering & Systems (ICCES), Cairo, pp. 185-190.

[27] B. Thanasopon, N. Sumret, J. Buranapanitkij and P. Netisopakul. 2017. Extraction and evaluation of popular online trends: A case of Pantip.com. 9th International Conference on Information Technology and Electrical Engineering (ICITEE), Phuket, pp. 1-5.

[28] Özel, S. A. 2011. A Genetic Algorithm Based Optimal Feature Selection for Web Page Classification. Proceedings of the 2011 International Symposium on Innovations in Intelligent Systems and Applications in Istanbul, Turkey, IEEE, 282-286.

[29] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K. and Slattery, S. 1998. Learning to Extract Symbolic knowledge From the World Wide Web. Proceedings of the 15th National Conference on Artificial Intelligence in Madison, Wisconsin, USA, American Association for Artificial Intelligence, 509–516.

[30] Ghani, R. 2001. CMU World Wide Knowledge Base (Web->KB) Project. http://www.cs.cmu.edu/~webkb/ (Access Date: 12 February 2016).

[31] Sinka, M. and Corne, D. (2002), "A large benchmark dataset for Web document clustering", Soft Computing Systems: Design, Management and Applications, Vol. 87, 881-890.

[32] Pazzani, M. 1998. Syskill and Webert Web Page Ratings. http://kdd.ics.uci.edu/databases/SyskillWebert/SyskillWebert.data.html (Access Date: 12 February 2016).

[33] Porter, M. F. 1980. An Algorithm for Suffix Stripping. Program, 14(3), 130–137.

[34] Salton, G., Wong, A. and Yang, C. S. 1975. A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11), 613-620.

[35] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H. 2009. The WEKA Data Mining Software: An Update. ACM Special Interest Group on Knowledge Discovery in Data Explorations Newsletter, 11(1),10-18.

[36] Witten, I. H., Frank, E. and Hall, M. A. 2011. Data mining: practical machine learning tools and techniques with Java implementations, Morgan Kaufmann Publishers, San Francisco, CA.

[37] Joachims, T. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of the 10th European Conference on Machine Learning in Chemnitz, Germany, Springer-Verlag, 137-142.

[38] Baykan, E., Henzinger, M., Marian, L. and Weber, I. 2011. A Comprehensive Study of Features and Algorithms for URL-based Topic Classification. ACM Transactions on the Web, 5(3), Article 15.

[39] Han, J., Kamber, M. and Pei, J. 2011. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, CA.

[40] Rennie, J.D.M., Shih, L., Teevan, J. and Karger, D.R. 2003. Tackling the Poor Assumptions of Naive Bayes Text Classiers. Proceedings of the Twentieth International Conference on Machine Learning in Washington DC, USA, AAAI Press, 616-623.