Eliminating Noisy Information in Web Pages using featured DOM tree

Shine N. Das
Department of Computer Applications
Cochin University of Science & Technology
Cochin, India

Pramod K. Vijayaraghavan
Department of Computer Applications
Cochin University of Science & Technology
Cochin, India.

Midhun Mathew
Department of Computer Science & Engineering
MBITS
Kothamangalam
India

ABSTRACT

The exact information retrieval from the Web is now a great challenge for the researchers to device new methodologies for web mining. Due to the massive information on the Web, the size and number appear to be growing rapidly at an exponential rate which is often accompanied by a large amount of noise such as banner advertisements, navigation bars, copyright notices, etc. Although such information items are functionally useful for human viewers and necessary for the web site owners, they often hamper automated information gathering and web data mining. The efficiency of feature extraction and finally classification accuracy are certainly degraded due to the presence of such noisy information. Thus cleaning the web pages before mining becomes critical for improving the mining results. In our work, we focuses on identifying and removing local noises in web pages to improve the performance of mining. We propose a novel and simple idea for the detection and removal of local noises using a new tree structure called *featured DOM Tree*. A three stage algorithm is proposed in which feature selection is done in the first phase, a featured DOM tree is created in the second phase and noise is marked and pruned in the third phase. The experimental results show that our algorithm outperform in terms of various benchmark measures and an increase in F score and accuracy is obtained as a result of automatic web page classification.

General Terms

Web content mining. Web page classification.

Keywords

Noise elimination, Featured DOM tree, Web page cleaning, Web page Classification, Minimum Weight Overlapping.

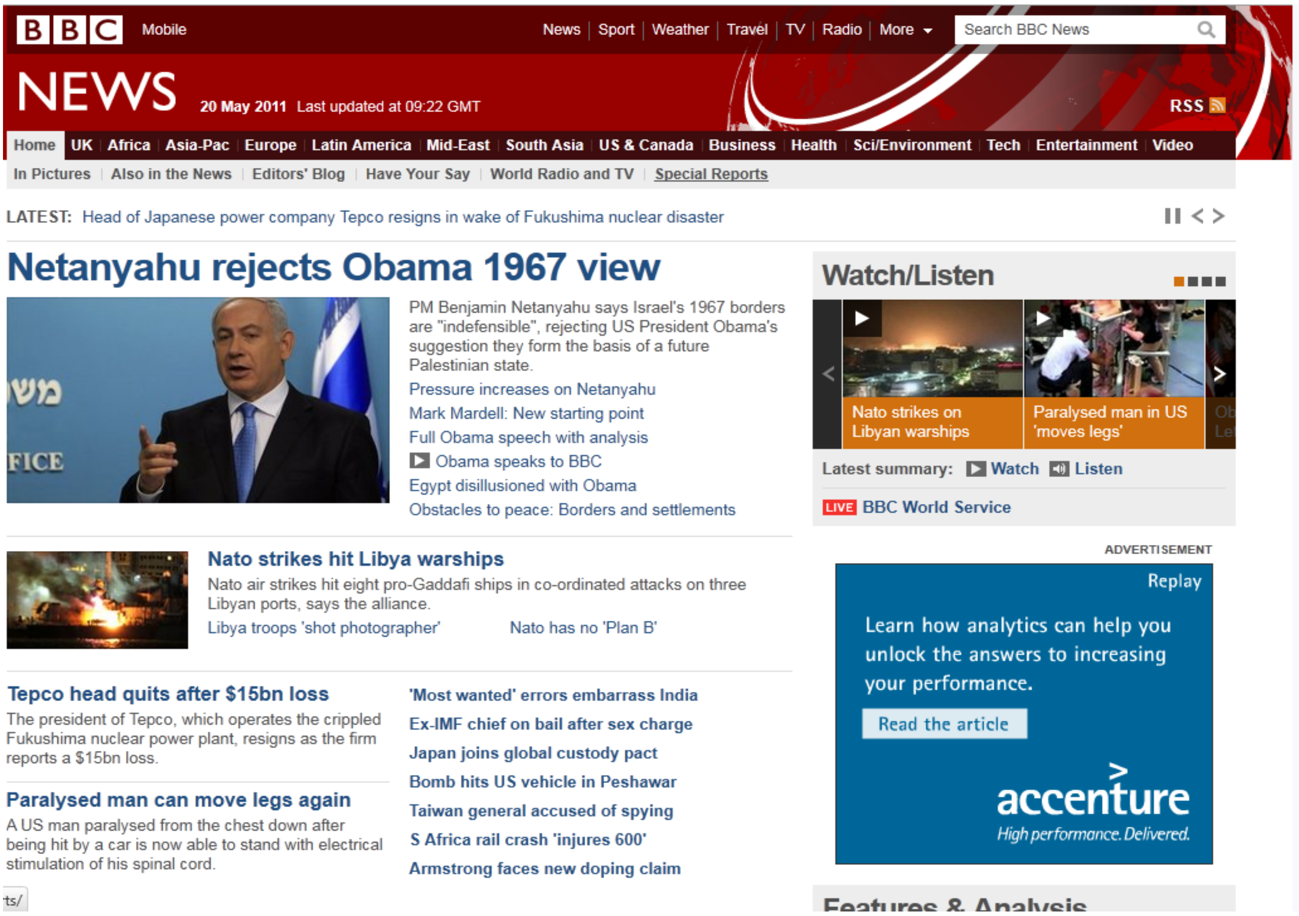
1. INTRODUCTION

The amount of information on the Internet shows tremendous growth, and the size and number appear to be growing rapidly at an exponential rate. Given the enormous volume of web pages in existence, it comes as no surprise that Internet users are increasingly using search engines and search services to find specific information. Searching the Web thus becomes an important task for discovering useful knowledge or information. However, useful information on the Web is often accompanied by a large amount of noise such as banner advertisements, navigation bars, copyright notices, etc. Although such information items are functionally useful for human viewers and necessary for the web site owners, they often hamper automated information gathering and web data mining. Information mixing also increases the difficulty for search engines, crawlers and information agents to extract the relevant information. Since web pages are constructed not only with main contents but also these types of noises, it is also important to distinguish valuable information from noisy data within a single web page.

Web pages are often littered with noisy objects around the article that deflect the user from the actual content. Web noise can be of two types [1]: *Global noise* refers to redundant objects with large graininess, which is no smaller than individual page. Global noise includes mirror sites, duplicated web pages and old versioned web pages to be deleted etc. *Local* (*intra-page*) *noise* refers to irrelevant items within a web page. Such noise includes banner ads, navigational guides, decoration pictures etc. This paper focuses only on local noise elimination methods. Since the main content is surrounded by noises in the retrieved web data, the efficiency of feature extraction and finally classification accuracy are certainly degraded.

According to [2], there are at least four different known categories of noise patterns within web pages of any web sites including banners with links including search panels, advertisements, navigational panel (directory list) and copy right and privacy notice in each web site. Many web pages contain these four noise categories together but most of the noise patterns are structured by using sectioning tags and sectioning separating tags and interactive tags. Moreover, anchor tags are most commonly used to link another web page or another web site. Figure 1 gives a sample web page from BBC News with main contents, advertisements, navigation links etc.

Corresponding Author – Shine N Das, email : shine_das@rediffmail.com, Presently working as an Associate Professor in College of Engineering, Munnar , Kerala India.



BBC Mobile

News | Sport | Weather | Travel | TV | Radio | More

Search BBC News

NEWS 20 May 2011 Last updated at 09:22 GMT

RSS

Home | UK | Africa | Asia-Pac | Europe | Latin America | Mid-East | South Asia | US & Canada | Business | Health | Sci/Environment | Tech | Entertainment | Video

In Pictures | Also in the News | Editors' Blog | Have Your Say | World Radio and TV | Special Reports

LATEST: Head of Japanese power company Tepco resigns in wake of Fukushima nuclear disaster

Netanyahu rejects Obama 1967 view

PM Benjamin Netanyahu says Israel's 1967 borders are "indefensible", rejecting US President Obama's suggestion they form the basis of a future Palestinian state.

Pressure increases on Netanyahu

Mark Mardell: New starting point

Full Obama speech with analysis

Obama speaks to BBC

Egypt disillusioned with Obama

Obstacles to peace: Borders and settlements

Nato strikes hit Libya warships

Nato air strikes hit eight pro-Gaddafi ships in co-ordinated attacks on three Libyan ports, says the alliance.

Libya troops 'shot photographer' Nato has no 'Plan B'

Tepco head quits after $15bn loss

The president of Tepco, which operates the crippled Fukushima nuclear power plant, resigns as the firm reports a $15bn loss.

Paralysed man can move legs again

A US man paralysed from the chest down after being hit by a car is now able to stand with electrical stimulation of his spinal cord.

'Most wanted' errors embarrass India

Ex-IMF chief on bail after sex charge

Japan joins global custody pact

Bomb hits US vehicle in Peshawar

Taiwan general accused of spying

S Africa rail crash 'injures 600'

Armstrong faces new doping claim

Watch/Listen

Nato strikes on Libyan warships

Paralysed man in US 'moves legs'

Latest summary: Watch Listen

LIVE BBC World Service

ADVERTISEMENT

Replay

Learn how analytics can help you unlock the answers to increasing your performance.

Read the article

accenture

High performance. Delivered.

Features & Analysis

Fig 1: A part of a web page with main contents and local noises

Recently, it becomes more difficult to extract the target information from the Web due to the complexity and the diversity of web page representation. This is an expectable phenomenon since the Internet has been so popular and the kinds of contents that are represented on the Web have been so diverse including videos, images, flashes, and so on. In addition to these diverse contents, the HTML structure of a web page is also getting more complicated, making it harder to analyze the page to recognize the target contents [3].

Elimination of noisy and irrelevant contents from web pages has many applications, including web page classification, clustering, web featuring, proper indexing of search engines, efficient focused crawlers, cell phones and PDA browsing, speech rendering for the visually impaired, improving the quality of search results and text summarization. Thus cleaning web pages for web data extraction becomes crucial for improving the performance of information retrieval. We investigate to remove various noise patterns in web pages instead of extracting relevant content blocks from web pages.

In this work, we focus on identifying and removing noises in web pages to improve the performance of web content mining considering the fact that local noises in web pages can seriously harm the accuracy of mining. Here, we propose a novel and simple idea for the detection and removal of local noises from a web page. A three stage algorithm is proposed in which feature selection is done in the first phase, a *featured DOM tree* is created in the second phase and noise is marked and pruned in the third phase. It is done by combining a different term weighting approach for optimal feature subset selection, featured DOM tree modelling of the page showing the layout and finally a new similarity verification technique called Minimum Weight Overlapping *(MWO).* Our experiment results based on web page classification clearly show that our cleaning technique is able to boost up the mining results drastically and the average classification accuracy increases remarkably.

2. RELATD WORKS

The original idea of this work has been evolved while developing an innovative approach for effective optimal feature subset selection for web page categorization. The subsistence of local noise is an issue that accompanies the growing need to extract relevant blocks from a web page. Web content mining face huge problems due to the presence of the local noise. There have been a number of studies that analyze an HTML page visually in order to extract the target information from the pages and most of them have focused on detecting main content blocks in web pages but less work have been evolved on detecting and removing noisy information from a web page.Lan Yi et al. [4] proposed a compressed structure tree (CST) to capture the common structure and comparable blocks in a set of web pages. It then uses information based measure to evaluate the importance of each node in CST. Based on the tree and its node importance values, a weight is assigned to each word feature in its content block. The resulting weights were used in web mining.

T. Sun et al. [5] created a DOM tree on the visual blocks of a web page and for each block, an information block matching ratio is calculated and checked against a threshold to identify the level of relevancy. But the algorithm was mainly based on an assumption that the same site are often made from a different page with an HTML template generation, their structure is very similar to the same or only part of the theme of data with different contents.

Kang et al. [6] built a tree alignment model representing HTML structure and a vector model representing the features of the blocks. They stated that the blocks of a web page might be related to different categories even though they are structurally similar. Since it is difficult to classify the blocks into accurate categories through building one classifier, multiple classifiers are built, one for each training domain, and the block classification proceeded through combining them. Through block classification, relevant and irrelevant blocks are identified.

A new tree structure, called Style Tree, is proposed in [1] to capture the actual contents and the common layouts of the pages in a web site. An information (or entropy) based measure is used to evaluate the importance of each element node in the style tree, which in turn helps to eliminate noises. T. Htwe et al. [7] proposed a DOM tree based approach which is based on the basic idea of Case-Based Reasoning (CBR) to find noise pattern in a page by matching similar noise pattern kept in Case-Based. They applied a back propagation neural network algorithm to classify the stored various noise patterns by matching similar noise data in the page.

Yossef et al. [8] proposed a method to identify frequent templates of web pages and pagelets to perform data cleaning in hypertext corpora. Lin et al. [9] developed a method which partitions a page into several content blocks and according to the entropy value, a method is proposed to dynamically select the entropy threshold that partitions blocks into either informative or redundant. Informative content blocks are distinguished parts of the page, whereas redundant content blocks are common parts.

Unfortunately, segmentations of the web page by HTML layout labels are only used for content display. Thus, the semantic relevance between different parts in the same block is not guaranteed. Two nodes relevant semantically may be segmented into different blocks because they are not in the same layout label. What makes things even worse is that a large amount of web pages fail to use HTML labels canonically. Page segmentation algorithms are based on a common assumption that closer texts in positions are usually more relevant in semantic. But this assumption is not always right [10].

Majority of the techniques were based on the observation that web pages usually share some common layouts and presentation styles, which is not true in all cases especially when loading dynamic web pages. Also many of these methods need a set of web pages even from a single web site, which is an extra burden while dealing with individual pages for web mining. In this paper, we propose a highly effective technique to clean web pages with the purpose of improving the results of web data mining. Here we consider the context of each block rather than its presentation style within a web page for cleaning it.

3. PROPOSED WORK

Here we propose a novel and efficient frame work for the true detection and well suited removal of noisy blocks. Noise elimination can be implemented as a pre-processing step for web content mining and especially for web page classification. Our objective is to find how to identify noisy blocks or irrelevant blocks from an input record, the web page to be processed, with a reduced complexity and increased efficiency. A three stage algorithm is proposed with phases *featuring, modelling* and *pruning*. It is proposed by combining a different term weighting approach (for main content, URL, heading, title, anchor text, information in the meta-tags etc.) for optimal feature subset selection [11], featured DOM tree modelling of the entire web page showing the layout and finally, a new similarity verification technique called Minimum Weight Overlapping (*MWO)* [12].

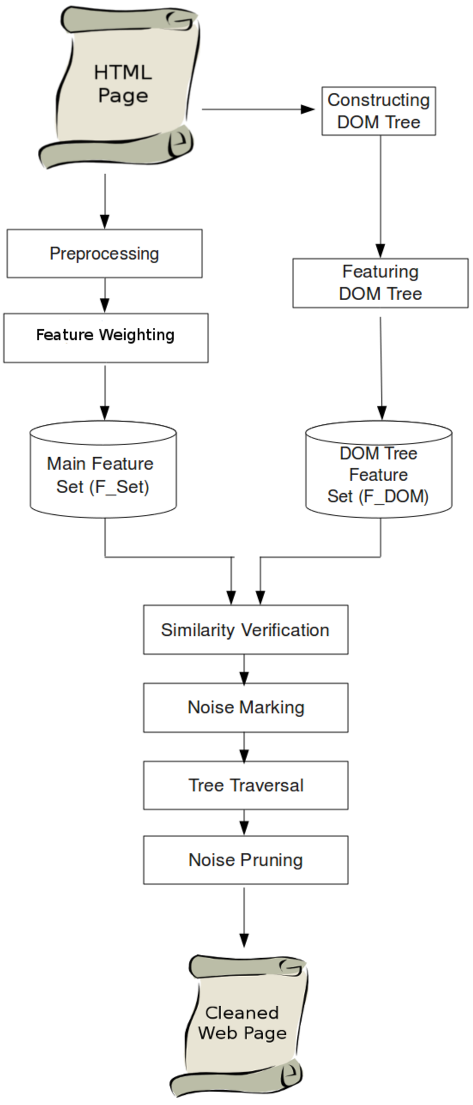
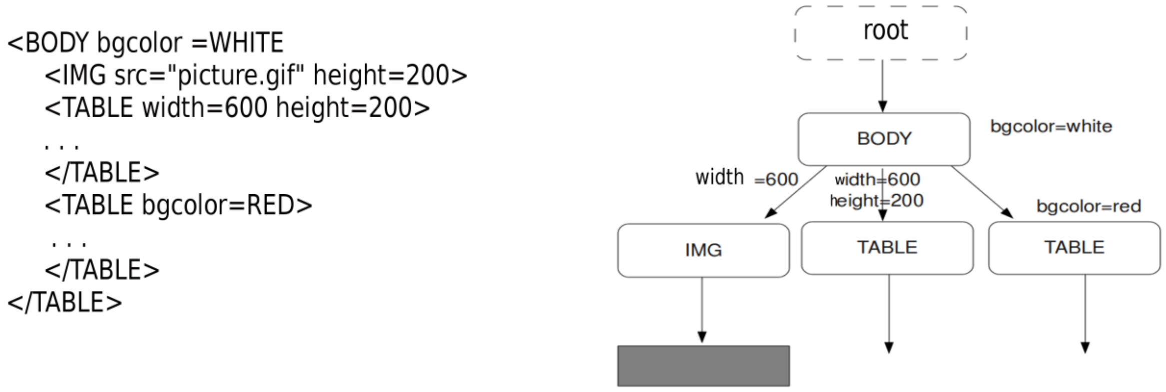


Fig 2: Overall flow diagram

In the first phase, featuring phase, standard web page pre-processing methods like *html* tag removal, tokenization, removal of stop words and stemming are applied on the input record and a feature set F of m tokens $\{x_1,x_2,\ldots,x_m\}$ are retrieved. Then a standard weighting scheme W is proposed, based on the term fields where the term *x* is present in the record. Even though a common term is present in two web pages, it not only depends on the number of occurrences in each page, but also depend on the fields in which it is present, since a web page is entirely different from a normal text file. For example, if a word is present in the title of one web page and in the content block of another web page, they differ by significance. In the first document, it may be a main feature while in the second one it has got less importance. The number of occurrences of each token *x* is multiplied with the weights of respective term fields w_i wherever the term is present and added together for total weight of a term W_x. Further this weight is normalized based on the total weight of the record, W_r using some standard approaches. After applying this weighting scheme proportional to the number of occurrences of a particular feature, we select the features which are having a score above a threshold value. This threshold value is dynamically varying according to the length of the document or maximum weight of the terms [11][12]. Optimal feature set thus obtained, *F_Set*, is used for further similarity verification and for the detection of noisy blocks a.

In the second phase, modelling phase, the HTML document is modelled as a DOM tree (Document Object Model tree). Each HTML page corresponds to a DOM tree where tags are internal nodes and the detailed texts, images or hyperlinks are the leaf nodes. Figure 3 shows a segment of HTML codes and its corresponding DOM tree. In the DOM tree, each solid rectangle is a tag node. The shaded box is the actual content of the node. The study of HTML web pages begins from the BODY tag since all the viewable parts are within the scope of BODY. Each node is also attached with its display properties. For the convenience of analysis, a virtual *root* node is added without any attribute as the parent tag node of BODY in the DOM tree [1].



```
<BODY bgcolor =WHITE
    <IMG src="picture.gif" height=200>
    <TABLE width=600 height=200>
    . . .
    </TABLE>
    <TABLE bgcolor=RED>
     . . .
    </TABLE>
</TABLE>
```

Fig 3: An example DOM tree

Although a DOM tree is sufficient for representing the layout or presentation style of an HTML page, it is hard to study the content or semantics of the page to clean it. Thus, DOM tree is not enough in our cleaning work which considers real content of the web page. We need a more powerful structure for this purpose. This structure is critical because our algorithm needs it to find feature sets from various blocks of the page in order to eliminate local noise. We introduce a new tree structure, called *featured DOM tree*, which is able to represent the presentation style as well as the feature sets of individual blocks of the web page. For creating a featured DOM tree, an optimal feature selection is done for individual leaf nodes of the DOM tree and feature weighting can also be applied here based on the leaf node tag. As a result of this phase, a set of feature sets is obtained, $F\_DOM = \{F\_DOM_1, F\_DOM_2, ...\}$ where $|F\_DOM|$ = Number of leaf nodes in the DOM tree. Figure 4 shows a typical featured DOM tree. Similarity verification is done in the third phase and noisy blocks are marked, propagated and further eliminated.

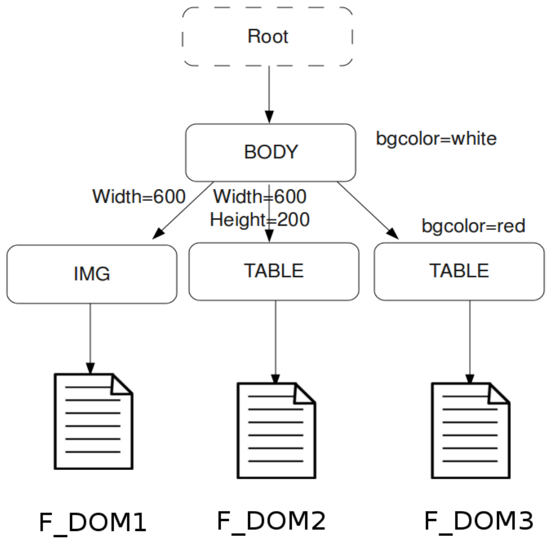


Fig 4: An example Featured DOM tree

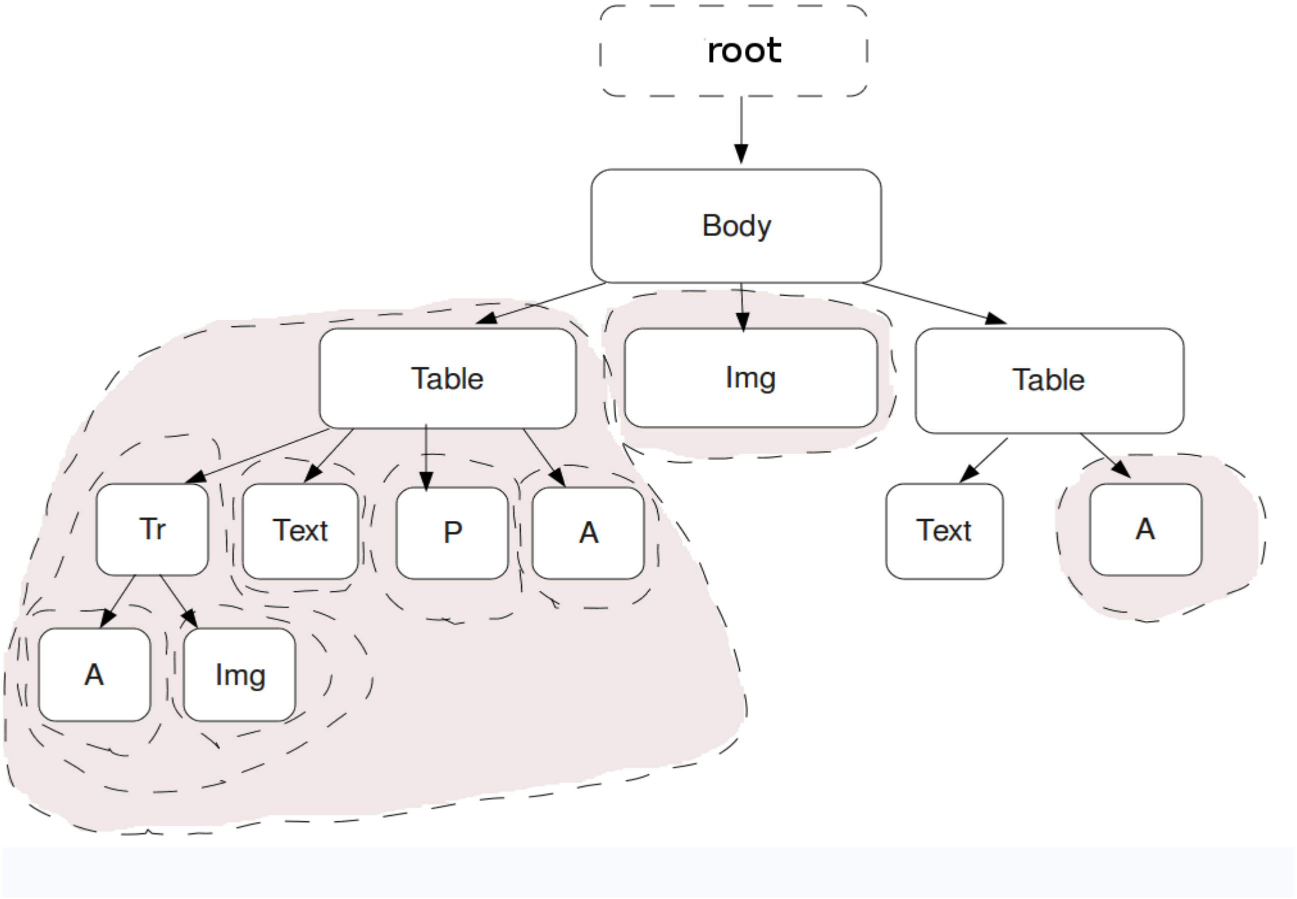


Fig 5: Noise marking and tree traversal in DOM tree

In the third phase, pruning phase, a noise checking is done for each F_DOM_i which is purely based on a feature set similarity measure. The weight percentage of each token in a feature set is calculated with respect to the total weight of the set and a new technique known as Minimum Weight Overlapping (MWO) is applied here for similarity verification. The MWO between *F_Set* and F_DOM_i is calculated as the sum of minimum weights of each token in those two features sets [12]. An example calculation of MWO of two feature sets are given in Table 1 where $F\_Set_1=\{x_1, x_2, x_4\}$, $F\_Set_2=\{x_2, x_3, x_4\}$ and W_{ij} represents the weight percentage of a token x_i in feature set *j*. Feature set representing each leaf node in the featured DOM tree, $F\_DOM_{i,}$ is compared with main feature set *F_Set* to find its MWO value and if it does not overcome a predefined threshold value t, that $Leaf\_Node_i$ in the DOM tree is marked as a noisy node. This procedure is known as *marking* which is tried on all leaf nodes. By changing the value of *t*, we can control the relevancy of noise detection.

Table 1: Example Calculation of MWO

Feature set / **terms**	F_Set_1	F_Set_2	*Min (W)*
x_1	W_{11}	0	0
x_2	W_{21}	W_{22}	min(W_{21},W_{22})
x_3	0	W_{32}	0
x_4	W_{41}	W_{42}	min(W_{41},W_{42})
Total	**100**	**100**	***MWO*** **=** $\sum$**Min (*W*)**

Next step is to remove noisy blocks from DOM tree. For that purpose, a bottom up traversal is done on the tree in such a manner that a parent node is marked as a noisy one if all of its children are noisy (Fig .5) . So this marking can be propagated up the tree. Finally the marked portion of the DOM tree is pruned and remaining tree structured is mapped back into HTML page so that a cleaned web page can be obtained.

Proposed Algorithm

Algorithm: *Noise_Elimination*

Input: A Web page (*Web_Doc*)
Output: Cleaned web page, *Out_Doc*

```
Noise_Elimination (Web_Doc)
    F_Set ← Featuring (Web_Doc);
    F_DOM ← Modelling(Web_Doc);
    Out_Doc ← Pruning(F_Set, F_DOM);
    return Out_Doc;
```

Algorithm: *Featuring*

Input: *Web_Doc*
Output: *F_Set*
Remarks: W_x→ total weight of a term *x*

```
Featuring (Web_Doc)
        In_Doc← Pre_Processing(Web_Doc);
        F←Full_Feature_Set(In_Doc);
        for all $x \in F$
                $W_x$←Weight_Scheme(x);
        $W_r \leftarrow \sum W_x$;
        for all $x \in F$
                $W_x$←Normalize($W_x$, $W_r$);
        T←Dynamic_Threshold($W_r$);
        F_Set ← φ;
        for all $x_i \in F$
                if ($W_x \geq T$)
                        F_Set ←  F_Set ∪ $x_i$;
        return F_Set;
```

Algorithm: *Modelling*

Input: *Web_Doc*
Output: *F_DOM*
Remarks: A virtual *root* node is added to the DOM tree

```
Modelling(Web_Doc)
        DOM ← Build_DOM(Web_Doc);
        root ← Add_Root(DOM);
        F_DOM ← φ;
        for each $Leaf\_Node_i \in DOM$
                Leaf_Doc← Pre_Processing($Leaf\_Node_i$);
                $F\_DOM_i$ ← Full_Feature_Set(Leaf_Doc);
                F_DOM ← F_DOM ∪ $F\_DOM_i$;
        return F_DOM;
```

Algorithm: *Pruning*

Input: *F_Set, F_DOM*
Output: *Out_Doc*
Remarks: *t*, a predefined threshold value
W_{ij} represents the weight of a feature x_i in feature set *j*

```
Pruning (F_Set, F_DOM)
        for each $F\_DOM_i \in F\_DOM$
                $Leaf\_Node_i$.marked ←  false;
                for each feature $x_p \in F\_DOM \cap F\_DOM_i$

                minW ← min($W_{p\ F\_DOM}$, $W_{p\ F\_DOMi}$);
                MWO ← ∑minW;
                if (MWO < t)
                        $Leaf\_Node_i$.marked ← true;
        for each $Leaf\_Node_i$.marked == true
                Propagate ($Leaf\_Node_i$);
        Eliminate_Marked(root);
        Out_Doc←MaptoHTML(root);
        return Out_Doc;

Propagate (Node)
        if (Node == root)
                return;
        Current ← Node.parent;
        Current.marked ← true;
        for each current.child
                if (current.child.marked ==false)
                        current.marked ← false;
        if (current.marked == true)
                Propagate (current);
        return;
```

4. EXPERIMENTAL RESULTS

In this section we evaluate the performance of the proposed noise elimination algorithm. Since the purpose of our noise elimination is to improve web mining, we performed a web mining task, automatic web page classification, to evaluate our system. By comparing the classification results before and after cleaning, we show that the proposed technique is better enough to improve the classification results. The methodology followed here consisted of selecting a random set of web pages from selected categories to form different data sets, determining a set of features to represent each data set, preparing a pair of datasets before cleaning and after cleaning, applying weighting scheme to weight the features, proper indexing of each page, training for automatic web page categorization, and finally the evaluation of the resulting categorization.

4.1 Data Collection

To validate the performance fairly, 10 pairs of different web datasets were prepared in our experiments. The target classes were chosen, arbitrarily, to be Arts, Business, Computers and Health. As we wanted to generate datasets consisting of web pages in these categories, we looked to the Open Directory Project categories available in [13]. The pages from these categories all contain a large amount of noise. Each page collected is pre-processed, featured and weighted according to the weighting scheme and properly indexed to create a Term Document Weight Matrix (TDW matrix) [14] for each data set. This procedure was repeated for 10 pairs of different sets and experiments were conducted on 20 different repositories thus created. Each pair consists of a random set of web pages from those categories before cleaning and after cleaning.

4.2 Experimental Set up

To conduct required experiments, we created an online tool which retrieves the contents of an input web page before cleaning; perform all pre-processing steps and extract weighted feature set. A TDW matrix is formed for each dataset thus processed. The resultant matrix obtained is supplied for building a classifier and classification results are analyzed. The same set of web pages are again fed into another tool which detect and remove the local noises from each web page. A new TDW matrix is created with cleaned documents and again classification is performed. The results are compared in terms of F score and accuracy. Both of the tools were implemented using PHP.

In our experiments, a popular classifier, namely NBC (Naive Bayes Classifier), is chosen to test prediction capability of the selected subset. The reason to choose NBC is because of its relatively high efficiency. The basic idea of NB classifier is to use the joint probabilities of words and classes to estimate the probabilities of classes given a document. NBC utilizes Bayes formula to distinguish which label an instance belongs to. Moreover, the conditional probability distribution of any given class satisfies normal distribution. Many experiments have demonstrated that NB classifier has good performance compared with others on various real datasets [15].

The experimental platform was Weka, which is an excellent tool in data mining and brings together many machine learning algorithms under a common frame work. To achieve impartial results, ten 10-fold cross validations had been adopted for each dataset while verifying classification performance. This process is often used to give statistical validity to situations where the data sets are small. This is to say, for each pair of datasets, before and after cleaning, we run classification algorithm on it 10 times and at each time, a 10-fold cross validation was used, and the final results were their average values.

4.3 Performance Analysis

We show that how the noise misleads data mining algorithms to produce poor results. To ascertain the validity of the proposed measure, we performed the experiments of automatic web page categorization and the obtained results using the proposed measure were analyzed in terms of standard benchmark measures. We use the popular F score measure to evaluate the results before and after cleaning. We also include the accuracy of results for classification. F score measures the performance of a system on a particular class, and it reflects the average effect of both precision and recall during automatic web page classification.

$$precision = \frac{categories\ found\ and\ correct}{total\ categories\ found}$$

$$recall = \frac{categories\ found\ and\ correct}{total\ categories\ correct}$$

$$F\ score = \frac{2 * precision * recall}{precision + recall}$$

Table 2 gives the averaged F scores and accuracies before and after cleaning. In the table, F1 and A1 stand for F score and accuracy before cleaning, while F2 and A2 stand for F score and accuracy after cleaning respectively. From the results, we can see that after cleaning, the classifier performs much better and the average values of both F score and accuracy improves by a large margin. Hence, our technique of detecting and eliminating noise has improved the mining results substantially.

Table 2: Classification results before and after cleaning

Dataset pair No.	F1	F2	A1	A2
1	0.743	0.931	0.773	0.954
2	0.751	0.980	0.818	0.943
3	0.813	0.925	0.749	0.915
4	0.752	0.987	0.798	0.958
5	0.716	0.946	0.840	0.945
6	0.527	0.827	0.791	0.937
7	0.747	0.919	0.767	0.919
8	0.724	0.914	0.732	0.921
9	0.532	0.852	0.812	0.960
10	0.650	0.860	0.835	0.945
Average	**0.696**	**0.914**	**0.792**	**0.940**

Weighting scheme plays an important role in semantic analysis while majority of existing works depend only on simple term frequency based approaches or boolean approaches. In our method, by measuring the MWO values, we could predict the noisy blocks instantly and precisely. We observed that cleaning, in general, improves F score and accuracy in all cases.

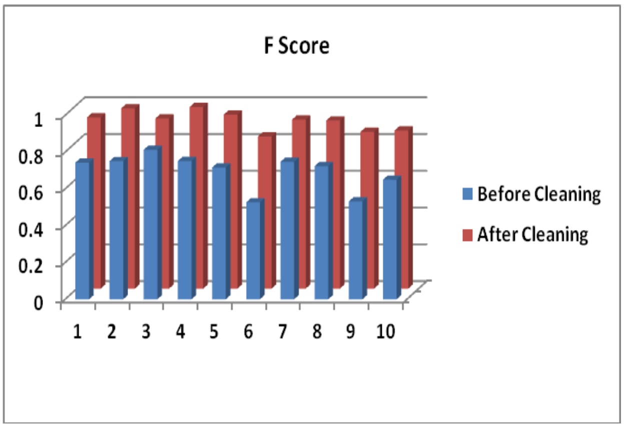


Fig 6: F score before and after cleaning

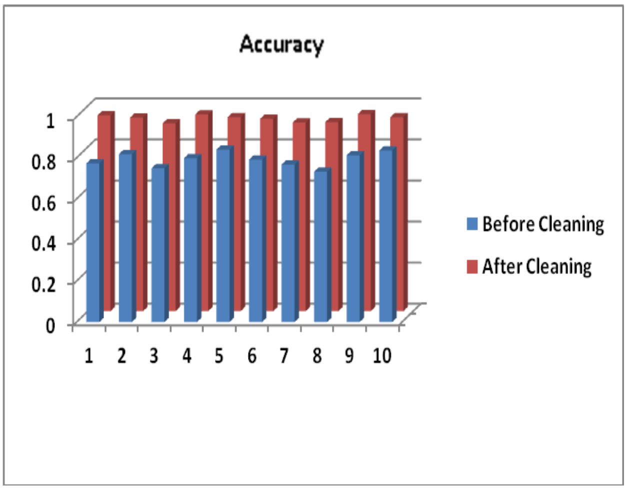


Fig 7: Classification accuracy before and after cleaning

5. CONCLUSION AND FUTURE WORK

This paper proposed a novel task for finding local noise in web pages. The proposed technique aims at helping document classification in web content mining based on a new tree structure, featured DOM tree, and MWO method for similarity verification. Instead of processing a set of web pages as such, we proposed a three stage algorithm which runs on a single web page and increases the mining result remarkably. In this paper, we focus an optimal feature subset selection method along with a similarity verification method for identifying noisy blocks of a page. We could detect and remove local noises with an increased relevancy. We evaluate the performance of our algorithm in terms of F score and accuracy of web page classification and we could achieve an improved result with a large margin than before cleaning.

Further research works can extend this to a more efficient method for directly finding main content blocks rather than identifying and pruning noisy blocks. It can be incorporated with search engines for better indexing and page ranking. Accuracy can be improved further by devising more efficient methods for optimal feature subset selection. Also this method can be easily associated with block classification of web pages directly with the help of featured DOM tree.

6. REFERENCES

[1] Lan Yi, Bing Liu, Xiaoli Li, “Eliminating Noisy Information in Web Pages for Data Mining”, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Washington, pp 296-305, August 2003.

[2] Thanda Htwe, “Cleaning Various Noise Patterns in Web Pages for Web Data Extraction”, International Journal of Network and Mobile Technologies, Vol, 1, Issue 2, pp 74 – 80, November 2010.

[3] Jinbeom Kang, Joongmin Choi, “Detecting Informative Web Page Blocks for Efficient Information Extraction Using Visual Block Segmentation”, International Symposium on Information Technology Convergence, pp 306-310, November 2007.

[4] Yi Lan, Liu Bing. “Web Page Cleaning for Web Mining through Feature Weighting”. Proceeding of Eighteenth International Joint Conference on Artificial Intelligence, Mexico, August 2003.

[5] Tieli Sun, Zhiying Li, Yanji Liu, Zhenghong Liu, “Algorithm Research for the Noise of Information Extraction Based Vision and DOM Tree”, International Symposium on Intelligent Ubiquitous Computing and Education, pp 81-84, May 2009.

[6] Jinbeom Kang, Joongmin Choi, “Block classification of a web page by using a combination of multiple classifiers”, Fourth International Conference on Networked Computing and Advanced Information Management, pp 290 -295, September 2008.

[7] Thanda Htwe, Khin Haymar Saw Hla, “Noise Removing from Web Pages Using Neural Network”, The 2nd International Conference on Computer and Automation Engineering, Singapore, Volume 1, pp. 281 – 285, February 2010.

[8] Ziv Bar-Yossef, Sridhar Rajagopalan, “Template Detection via Data Mining and its Applications”, Proceedings of the 11th international conference on World Wide Web, pp 580-591, 2002.

[9] Shian-Hua Lin, Jan-Ming Ho, “Discovering informative content blocks from Web documents”, Proceedings of ACM SIGKDD'02, July 2002.

[10] Jingqi Wang, Qingcai Chen, Xiaolong Wang, Hongzhi Guo, “Basic Semantic Units Based Web Page Content Extraction”, International Conference on Systems, Man and Cybernetics, pp 1489 – 1494, 2008.

[11] Shine N Das, Midhun Mathew, Pramod K.Vijayaraghavan, An Approach for Optimal Feature Subset Selection using a New Term Weighting Scheme and Mutual Information, Proceeding of the International Conference on Advanced Science, Engineering and Information Technology, Malaysia, pp 273-278, January 2011.

[12] Shine N Das, Midhun Mathew, Pramod K.Vijayaraghavan, “An Efficient Approach for Finding Near Duplicate Web pages using Minimum Weight Overlapping Method”, Proceedings of 20th ACM Conference on Information and Knowledge Management, Glasgow, Scotland, 2011.

[13] http://www.dmoz.org: Open Directory Project - The largest, most comprehensive human-edited directory of the Web.

[14] Midhun Mathew, Shine N Das, T.R Lakshminarayanan, Pramod K.Vijayaraghavan, "A Novel Approach for Near-Duplicate Detection of Web Pages using TDW Matrix", International Journal of Computer Applications, Volume 19, Number 7, April 2011.

[15] Andrew McCallum , Kamal Nigam, “A comparison of event models for naive Bayes text classification”, AAAI-98 Workshop on Learning for Text Categorization, 1998

.