ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 3; Issue 2; Page No. 23-26

# Noisy elimination for web mining based on style tree approach

**Deepa.R[1] , Nirmala Devi.R[2]**

[1]Assistant Professor, Department of Computer Science, Nandha Arts and Science College, Erode, Tamil Nadu, India

E-mail: mdsupreet@gmail.com

[2] Assistant Professor, Department of Computer Science, Nandha Arts and Science College, Erode, Tamil Nadu, India

E-mail: nibhunirmala@gmail.com

**ABSTRACT**

The Main Objective is to detect noise data in web documents and eliminate the noise data based on Style Tree approach. The Noise Elimination system is designed to eliminate noise and to identify the informative content section from the Web documents. In a given Web site, noisy blocks usually share some common contents and presentation styles, while the main content blocks of the pages are often diverse in their actual contents and presentation styles. Based on this observation, we propose a tree structure, called Style Tree, to capture the common presentation styles and the actual contents of the pages in a given Web site. By sampling the pages of the site, a Style Tree can be built for the site, which we call the Site Style Tree (SST). We then introduce an information based measure to determine which parts of the SST represent noises and which parts represent the main contents of the site. The SST is employed to detect and eliminate noises in any Web page of the site by mapping this page to the SST. The proposed technique is evaluated with two data mining technique, one is clustering and another one is classification. The K-Means algorithm is used to group the tree elements.

**Keywords:** Noise detection, Noise elimination, Clustering, Classification.

## INTRODUCTION:

The rapid expansion of the Internet has made the WWW a popular place for disseminating and collecting information. Data mining on the Web thus becomes an important task for discovering useful knowledge or information from the Web [6].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 Website owners, they often hamper automated information gathering and Web data mining, e.g., Web page clustering, classification, information retrieval and information extraction. Web noises can be grouped into two categories according to their granularities.

***Global noises*:** These are noises on the Web with large granularity, which are usually no smaller than individual pages. Global noises include mirror sites, legal/illegal duplicated Web pages, old versioned Web pages to be deleted, etc.

***Local* (*intra-page*) *noises*:** These are noisy regions/items within a Web page. Local noises are usually incoherent with the main contents of the Web page. Such noises include banner advertisements, navigational guides, decoration pictures, etc.

In this work, we focus on detecting and eliminating local noises in Web pages to improve the performance of Web mining, e.g., Webpage clustering and classification. This work is motivated by a practical application. A commercial company asked us to build a classifier for a number of products. They want to download product description and review pages from the Web and then use the classifier to classify the pages into different categories.

## II. RELATED WORK

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.

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. The most popular features used for boilerplate detection [3] on two corpora. They show that a combination of just two features - number of words and link density - leads to a simple classification model that achieves competitive accuracy. The features have a strong correspondence to stochastic text models introduced in the field of Quantitative Linguistics. A new content extraction method is thus proposed [5], which can discover web page content according to the number of punctuations and the ratio of non-hyperlink character number to character number that hyperlinks contain. It can eliminate noise and extract main content blocks from web page effectively. In this paper [6] they propose a novel idea for finding near duplicates of an input web page, from a huge repository. This approach explores the semantic structure, content and context, of a web page rather than the content only approach. They present a three-stage algorithm which receives an input record and a threshold value and returns an optimal set of near duplicates. By introducing a new technique known as Minimum Weight Overlapping (MWO) based on the threshold value and finally we get an optimal number of near duplicate records. To extract informative content block from Web documents by removing noise blocks. So, to extract information from these pages, several challenges must be overcome. Another existing informative content extraction system implemented nowadays depends on rule based systems where Web sites with various templates are not applicable. A possible application of Neural Networks [4] is presented for three pattern classification combine with DOM structure to extract content information. The type of Neural Network used to implement our system is feed forward which uses the back propagation learning algorithm. The proposed paper [2] intends to introduce an algorithm which could extract main content that is not necessarily the dominant content and without any learning phase, with one random page and by using visual cues to simulate user page visit and block the page based on it and gains higher precision. Next section demonstrates the proposed algorithm in this paper in two main phases, first block tree construction, and then it finds main block from the block nodes in the computed block tree.

## III. THE PROPOSED TECHNIQUE

The proposed cleaning technique is based on the analysis of both the layouts and the actual contents (i.e., texts, images, etc.) of the Web pages in a given Web site. The Local Noisy blocks can seriously harm Web Data Mining.The noisy blocks usually share some common contents and presentation styles. The Style Tree, to capture the common presentation styles and the actual contents of the Web pages in a Web Site. The Style Tree is proposed for the noise elimination system. The Style Tree is based on DOM (Document Object Model) tree.The DOM tree is commonly used for representing the Single Web Page Structure.

**DOM Tree**

<BODY bgcolor=WHITE>
<TABLE width=800 height=200 >
</TABLE>
<IMGsrc="image.gif" width=800>
<TABLE bgcolor=RED>
</TABLE>
</BODY>

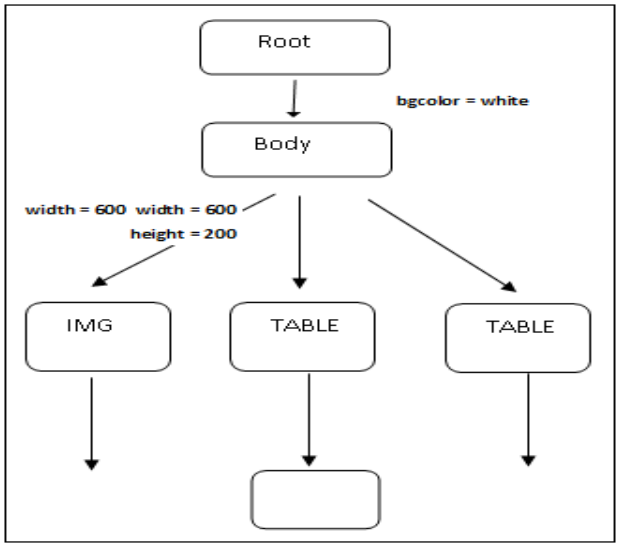


**Figure 1: Example DOM Tree**

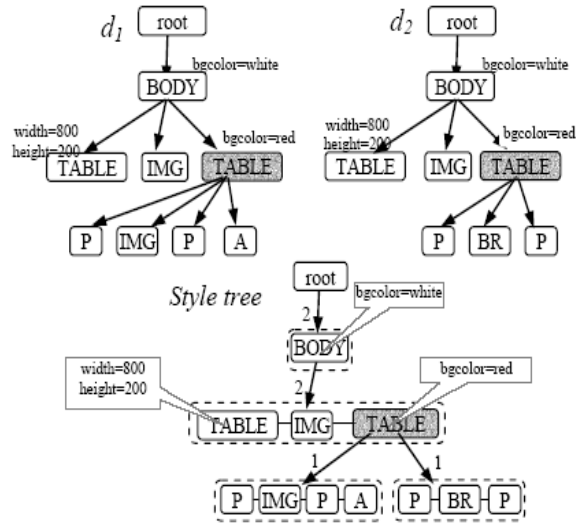


**Figure 2: Example STYLE TREE**

A Style Tree consists of two types of nodes, Style Nodes(S), Element Nodes (E),A Style node (S) represents a layout or presentation Style, which has two components ($E_S$,n) $E_s$ - sequence of element nodes, n - number of pages. An Element node E has three components.

TAG is the tag name, e.g. TABLE and IMG;

Attr is the set of display attributes of TAG, e.g., bgcolor = RED,

$S_s$ is a set of style nodes below E.

An Element node has more presentation style; the element node is called Noise.

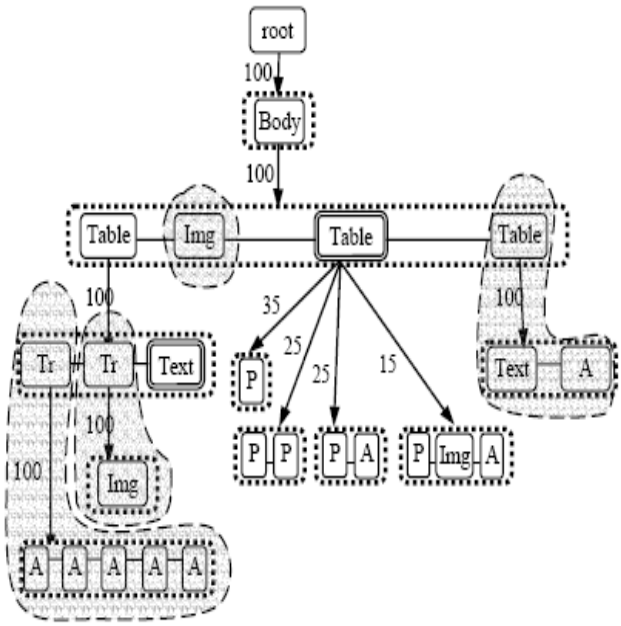


**Figure 3: Example Site Style Tree**

A metric (Node Importance and Composite importance) is used to measure the importance of a presentation style.

$$\text{lNodeImp}(E) = -\sum_{i=1} p_i \log_m p_i \text{ if } m > 1 \quad \text{if } m = 1$$

$p_i$- Probability

m - Number of pages containing E

1 - Number of child style nodes of E (i.e.., l=|E.$S_s$|)

The Node Importance is only considering the element node .It does not consider the importance of its descendents. The Composite Importance is consider the both Element Nodes and its descendents.

$$\text{CompImp}(E) = (1-\gamma^l)\text{NodeImp}(E) + \gamma^l \sum_{i=1} (p_i(\text{CompImp}(s_i))$$

$\gamma^l$ - attenuating factor , whichis set to 0.9 l - number of child style nodes of E (i.e.., l=|E.$S_s$|)

The Composite Importance increases the weight of NodeImp(E) when l is large. It decreases the weight of NodeImp(E) when l is Small.

An element node E in the SST, if all of its descendents and itself have composite importance less than a specified threshold t, element node is called as noisy. The Algorithm Mark Noise(E) to identify noisy element nodes in SST .The Algorithm Mark Noise(E) first checks whether all E's descendents are noisy or not. If any one of them is not noisy, then E is not noisy. If all its descendents are noisy and E's composite importance is also small, then E is noisy.

Input E : root element node of a SST

Return : TRUE if E and all of its descendents are noisy, else FALSE

**Algorithm: MarkNoise(E)**

Step 1: for each *S ϵE.Ss* do
Step 2: for each *e ϵS.Es* do
Step 3: if (MarkNoise(*e*) == *FALSE*) then
Step 4: return *FALSE*
Step 5: end if
Step 6: end for
Step 7: end for
Step 8: if (E.CompImp≤ t) then
Step 9: mark *E* as noisy.
Step 10: return *TRUE*
Step 11: else return*FALSE*
Step 12: end if

The DOM tree is used for the input to the Site Style Tree. The first page DOM tree is assigned as the initial Site Style Tree. The first page DOM tree is called as Page Style Tree. The other page DOM tree values are used for the style tree comparison process. Matched DOM trees are eliminated. Unmatched DOM trees are added to the Site Style Tree.

Input: E: Root element node of the simplified SST

Input: $E_{PST}$: root element node of the page style tree

Return: The main content of the page after cleaning Map SST (E, $E_P$)

Step 1: if E is noisy then
Step 2: delete $E_P$ as noises
Step 3: return NULL
Step 4: end if
Step 5: if E is meaningful then
Step 6: Ep is meaningful
Step 7: return the content under EP
Step 8: else return Content = NULL
Step 9: $S_2$ is the style node in $E_P$.Ss
Step 10: if$S_1$ϵE.Ss ^ S2 matches S1 then
Step 11: $e_{1,i}$is the $i_{th}$element node in sequence
Step 12: $e_{2,i}$is the $i_{th}$ element node in sequence $S_2$.Es;
Step 13: for each pair ($e_{1,i}$, $e_{2,i}$) do
Step 14: return Content += MapSST($e_{1,i}$ , $e_{2,i}$)
Step 15: end for
Step 16: return Content
Step 17: else $E_P$ is possibly meaningful;
Step 18: return the content under $E_P$
Step 19: end if
Step 20: end if

## IV. EXPERIMENATAL 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. We show that how the noise misleads data mining algorithms to produce poor results. 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.

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

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

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

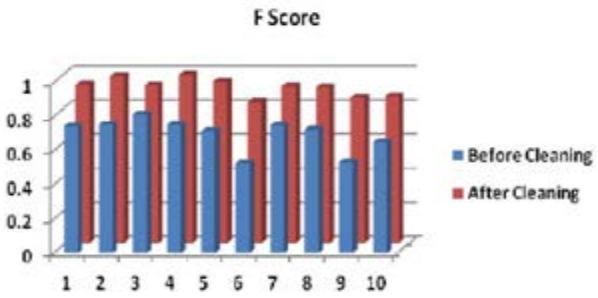


**Figure 4:**

## V. CONCLUSION:

The noisy data elimination system is implemented to perform the noise elimination and relevant content identification tasks .The Document Object Model (DOM) tree and style tree construction tasks are verified with the Hyper Text Markup Language (HTML) code for the Web documents. The noise data elimination and relevant content identification tasks results are verified manually. The K-Means algorithm is used to group up the DOM tree elements. The system produces relevant content without noise data from the Web documents.

## VI. REFERENCES:

1. Lan Yi, Bing Liu and Xiaoli Li, Eliminating Noisy Infromation in Web Pages for Data Mining. ACM- 2003
2. M. Asfia, M. M. Pedram and A. M. Rahmani, Main Content Extraction from Detailed Web Pages. International Journal of Computer Applications- 2010.
3. C. Kohlschutter, P. Fankhauser and W. Nejdl, Boilerplate Detection using Shallow Text Features.WSDM- 2010.
4. T. Htwe, N. S. M. Khan, Extracting Data Region in Web Page by Removing Noise using DOM and Neural Network. ICIFE- 2011.
5. R. Gunaundari and Dr. S. Karthikeyan.A Study of Content Extraction from Web Pages Based on Links.IJDKP-2012.
6. S. N. Das, M. Mathew and P. K. Vijayaraghavan. An Efficient Approach for Finding Near Duplicate Web Pages using Mininum Weight Overlapping Method. IJECE