
International Journal of Computational and Electronics Aspects in Engineering
Vol. 1, Issue. 2, March 2015, Revised in June 2020, pp 81-84
https://doi.org/10.26706/ijceae.1.2.20150108

# Discovering Informative Blocks from Web Pages for Efficient Information Extraction using DOM tree

**Rakesh M. Kohale**[1]
rakeshkohale77@gmail.com

**Shreyash G. Balbudhe**[2]
shreyash_balbudhe@rediffmail.com

Datta Meghe Institute of Engineering, Technology and Research, Sawangi, Wardha, India

***Abstract*— A webpage generally contains data along with navigation panels, advertisements, copyright and privacy notices. Except data these other things do not contain any important information. These blocks can be called as non-informative blocks. As these blocks are non-informative, they can affect the result of web data mining. To avoid this, it is important to separate the main data i.e. informative blocks and non-informative blocks from the web page. In a website these non-informative blocks are generally present in different web pages and have same format. Also, the data contained in these blocks is also same. In case of informative blocks, data contained by the block and their format are different. We need a structure at site level to capture the same format of the blocks and the data present in the blocks. DOM Tree structure is available at page level. Many tools are available to construct a DOM Tree of a webpage. But DOM Tree structure is not useful at site level. So, we need to construct a Site Style Tree (SST) for a website. After analyzing this SST, we can identify which part of SST is informative and which is non-informative. There is no tool available to construct a style tree for a given website. This work aims at constructing a style tree for given website and separating informative and non-informative blocks from the website.**

***Index Terms*—** DOM tree, Site Style Tree, Tokens, Parsing, Informative blocks, Non-informative blocks

## I. INTRODUCTION

The system aims to develop an automated tool to remove the non-informative blocks from the web pages. A webpage generally contains data along with navigation panels, advertisements, copyright and privacy notices. Except data these other things do not contain any important information. These blocks can be called as non-informative blocks. As these blocks are non-informative, they can affect the result of web data mining. To avoid this, it is important to separate the main data i.e. Informative blocks and non-informative blocks from the web pages. In a website these non-informative blocks are generally present in different web pages and have same format. Also, the data contained in these blocks is also same. In case of informative blocks, data contained by the block and their format are different. We need a structure at site level to capture the same format of the blocks and the data present in the blocks. DOM Tree structure is available at page level. Many tools are available to construct a DOM Tree of a webpage. But DOM Tree structure is not useful at site level. So, we need to construct a Site Style Tree (SST) for a website. After analyzing this SST, we can identify which part of SST is informative and which is non-informative.

Technical Article
First Online on – 30 March 2015, Revised on – 30 June 2020



Cite this article – Rakesh M. Kohale and Shreyash G. Balbudhe, "Discovering Informative Blocks from Web Pages for Efficient Information Extraction using DOM tree", *International Journal of Computational and Electronics Aspects in Engineering*, RAME Publishers, vol. 1, issue 2, pp. 81-84, 2015, Revised in 2020.
https://doi.org/10.26706/ijceae.1.2.20150108

There is no tool available to construct a style tree for a given website. This work aims at constructing a style tree for given website and separating informative and non-informative blocks from the website. DOM trees can be combined to generate a Style tree.

These all items are useful for viewers and necessary for website. Due to these items, retrieving required information from the web becomes very difficult. To improve the process of web data mining it is necessary to remove non-informative blocks from the web pages. For this we first need to identify such blocks from the website. This work aims at identifying such non-informative blocks from the website. These non- informative blocks can be removed to make the data mining more efficient. Non-informative blocks generally share same contents and presentation style in multiple web pages. So, capture this at site level a structure called Site Style Tree (SST) is needed. Once the SST for a website is built, informative and non-informative parts can be easily identified.

The example SST formation from [1] is shown below.

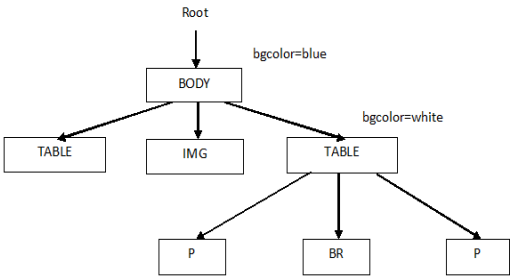


Fig. 1. DOM Tree D1

In Fig. 1. an example DOM tree D1 of a web page is shown. Intermediate nodes in the DOM Tree represents different HTML tags from corresponding web page and leaf node contains the actual content from the web page.

Fig. 2. Shows another DOM tree D2. All tags in D1 has its Corresponding tags in D2 except the bottom level tags. So, these two DOM trees can be combined to generate a Style tree. The resultant Style tree is shown on Fig. 3. This style tree contains all the nodes from $D_1$ and $D_2$. There are two types of nodes in style tree, element nodes and style nodes.

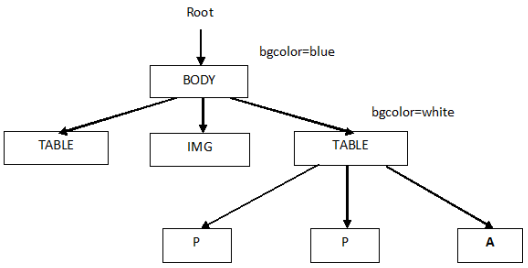


Fig. 2. DOM Tree D2

In Fig. 3. P-BR-P and P-P-A are two style nodes. Tag nodes in the style node are called element nodes. A count is maintained which indicates how many pages have same presentation style at that level. From this style tree we can observe that two presentation styles are present under rightmost table tag. Thus, by applying informative measure we can identify non-informative blocks. To clean the website, we can remove these non-informative parts. Also, when the new pages are added in the website, that page can be mapped on the SST of that site.

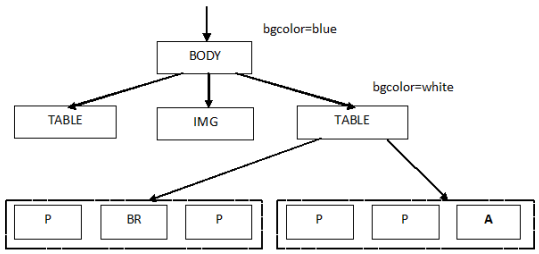


Fig. 3. Style Tree

## II. Literature Survey

A method is proposed in [4] to detect informative blocks from the news website. This work assumes that system already knows how webpage is partitioned and blocks containing similar information from different pages. But partitioning webpage and identifying corresponding blocks in different web pages are big issues. Also, in [4] web page is considered as collection of blocks and each block as collection of words. This is true in case of news website. Generally, these assumptions are very strong. Web page cleaning is considered as frequent template detection problem in [3]. In [3] webpage partitioning depends on the

number of hyperlinks of an HTML element. This partitioning method is not useful in case of web pages from the same website. In [8] some learning mechanisms are proposed. These helps to identify banner advertisements, redundant links of web pages. But these techniques require large training data set. These also require domain knowledge to generate classification rules. Some work includes duplicate records detection and cleaning of data for data mining. In [6] one approach called block analyzer is mentioned. Using this approach blocks are preclustered. An entropy-based value is assigned to each cluster. Using this value, the blocks in the cluster can be classified as informative or non-informative blocks. But it may cluster some informative blocks into a noisy cluster.

## III. Implementation Details

### *A. Cleaning HTML Pages*

Step 1: Cleaning the HTML page:

a. Symbols, "<" and ">", should only contain html tags. When used in another place, they should be replaced by "&lt" and "&gt" respectively.

b. All tags must be matched, i.e. every starting tag has a corresponding ending tag.

c. Attributes of all tags must be encircled by quotation marks.

d. All tags must be nested correctly. For example, <a>. . .. . .. . .</a> is a correct nest, while <a>. ... . .</a>. . . is incorrect.

Step 2: Preprocessing the web page tags.

All tags on the page form a tree structure. Those nodes that do not contain any text should be removed, as well as invalid tags such as<script> <style> <form> <marquee> <meta> etc. Which are unrelated to the content, and then the structure tree is built.

Step 3: Judging the location of content

The aim of this process is to select the optimum node containing content. If a node is not satisfied with this condition, the text under this node is not identified.

Step 4: Extracting the content

The content is extracted by tools such as html parser. If the node is not satisfied with the conditions, return the step 3 in order to find the optimal nodes of the next level nodes (the child nodes of the node).

Step 5: Adjusting the extraction results from step 4

In step 3, only the node that most likely contains the content is selected. But if the structure of a web page is relatively decentralized, it is very prone to extract a section or a paragraph of the whole content. As the adjacent nodes on the same level are free of judge, in this step, we must adjust the above result. The text also should be extracted from the adjacent nodes that meet the conditions of the precise content extraction. So all the text will be extracted from the qualified nodes on the same level.

Step 6: Removing unnecessary contents

In this step, the leaf nodes are compared with the tokens and finally the necessary contents will be displayed to the user as results.

### *B. Process block Diagram*

As shown in the diagram below, web page from a website is downloaded and provided to HTML Parser as a input. HTML Parser then generates DOM tree for that web page by parsing it. From this generated DOM tree, tags which does not contain any data or information, e.g. script tag, gets filtered using filter tags database. The result of this filtration process is the final DOM tree for that web page. This DOM Tree structure gets evaluated and resulting structure is called as Site Style Tree (SST). Now input from remaining HTML pages of a website is provided to generate the SST for a website. Leaf nodes in the DOM tree contain actual content of the page like text, image. So based on the information in this actual contents importance of leaf node is decided. For this leaf nodes from the DOM tree are extracted and information analysis is done. So final SST gets analyzed by applying these informative measures. After analyzing these SST informative and non-informative blocks are identified.

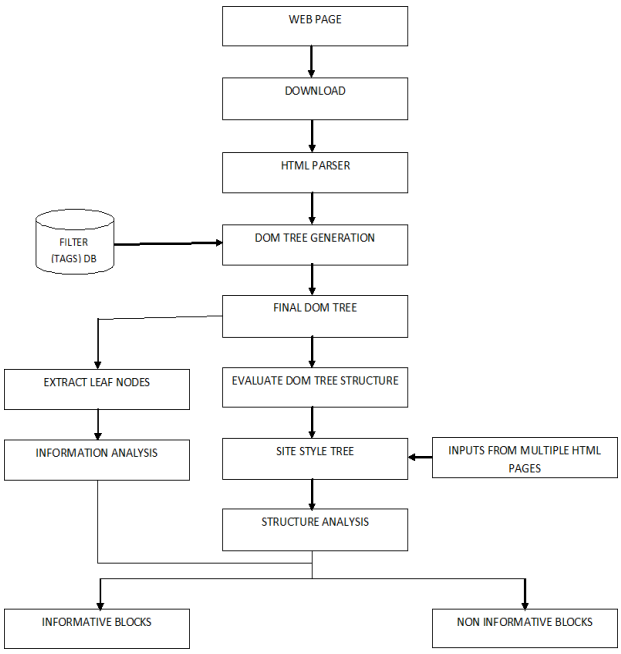


Fig. 4. Block Diagram

## IV. CONCLUSIONS

To improve web data mining, web pages should be clean. For cleaning the web pages non-informative blocks must be identified. As non-informative blocks share common presentation style and content, a site level structure is used Called Site Style Tree (SST). SST captures these common Presentation styles. An information-based measure is used to evaluate SST element nodes. When new pages are added to the website, these pages from the site are mapped to the SST. After removing non-informative blocks, the storage space and time for a webpage can be saved.

## REFERENCES

[1] R. Gunasundari, S. Karthikeyan, “Removing Non-informative Blocks from the Web Pages”, *Communication Control and Computing Technologies* (ICCCCT), 2010 IEEE International Conference.

[2] B. Liu, K. Zhao, and L. Yi, “Eliminating Noisy Information in Web Pages for Data Mining”, *Proc. Ninth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining*, pp. 296-305, 2003.

[3] Bar-Yossef, Z. and Rajagopalan, S., "Template Detection via Data Mining and its Applications", 2002.

[4] Shian-Hua Lin and Jan-Ming Ho., "Discovering Informative Content Blocks from Web Documents", KDD-02, 2002.

[5] S. Debnath, P. Mitra, and C.L. Giles, N.Pal “Automatic Identification of informative sections of Web Pages", *IEEE Transaction on Knowledge and Data Engineering* , 2005.

[6] Chia-Hsin Huang, Po-Yi Yen, Yi-Chan Hung, Tyng-Ruey Chuang, and Hahn-Ming Lee, "Enhancing Entropy-based Informative Block Identification Using Block Preclustering Technology", *IEEE International Conference on Systems, Man, and Cybernetics*, October 8-11, 2006, Taipei, Taiwan.

[7] Hung-Yu Kao, Jan-Ming Ho, Member, IEEE, and Ming-Syan Chen, Fellow, IEEE, "WISDOM: Web Intrapage Informative Structure Mining Based on Document Object Model", *IEEE transaction on Knowledge and Data Engineering*, VOL. 17, NO. 5, MAY 2005.

[8] Jushmerick, N., "Learning to remove Internet advertisements", *AGENT- 99*, 1999.

[9] Yao, Z. and Choi, B., 2007. “Clustering Web Pages into Hierarchical Categories,” *International Journal of Intelligent Information Technologies*, Special Issue on Web Mining, Vol. 3, No. 2, pp.17-35.