ISSN 2395-1621

# A Web Based Data Extraction Using Hierarchical (DOM Tree) Approach

[#1]Nagawade Megha Prabhakar, [#2]Narawade Shubhada Maruti, [#3]Shinde Manjusha Bhagwat
[#4]Prof. B. R. Burghate

[1]nagawademegha@gmail.com
[2]shubhadanarawade30@gmail.com
[3]shindemanjusha123@gmail.com

[#1234]Deparment of Computer Engineering
JSPM's
Bhivarabai Sawant Institute of Technology & Research,
Wagholi, Pune – 412207

**ABSTRACT**

**The exact information retrieval from the Web is now a great challenge for the researchers to device new methodologies for web mining. Different Web sites contain information on various topics in various formats. Large amounts of effort are often required for a user to manually locate and extract data of interest from the Web pages. Every time you need analyse data, you need to visit number of web sites. It is very time consuming process to construct wrapper to visit those sites and collect data. DEUDS, it is a page level data extraction system that automatically discovers extraction pattern from web pages for selected data section and extracts data. DEUDS uses visual cues to identify data records while ignoring noise items such as advertises and navigation bars.**

ARTICLE INFO

**Article History**

Received: 28th September 2015

Received in revised form :

1st October, 2015

Accepted : 5th October , 2015

**Published online :**

**6th October 2015**

*Keywords*- **DOM Tree, Web Page Renderer, Selector, Web Data Extraction**

## I. INTRODUCTION

Data mining is nothing but the process of extracting useful information from collected databases. Extraction of the information from the big databases is called the "Knowledge Discovery". It is an analytical tool for analyzing data. It allows user to analyze data from many different aspects or angles, categorize it, and conclude the relationships identified. Technically, it is the process of finding correlations or patterns among loads of field in large relational databases. Internet has made the WWW a popular place for collecting and sharing information. Mining on the Web has becomes an important task for extracting useful information from the Web. However, useful information on the Web is often accompanied by a large amount of noise such as banner advertisements, navigation bars, copyright notices, etc. The information contained in these noisy blocks can seriously harm Web data mining. Thus eliminating these noises is of great importance. An innovative idea is to use a DOM tree structure to abstract out the noises in web pages to improve the performance of mining. The aim of this work is to analyze and eliminate the noisy blocks in web documents using a DOM TREE structure. DOM Tree is constructed to determine the logical structure of a web document.

## II. RELATED WORK

A In that we will be use novel approach to extract data, which uses visual cues and CSS Selector for attributes of DOM tree to construct pattern for that three- step strategy to solve the problem of data extraction.

1) In this step, page is divided into sections. Section is nothing but the part of page containing useful data. Some technique uses MDR for dividing page into section. We are using visual cues to find data records. Visual information helps in two ways:

a) It enables to identify gaps that separate data records, which helps to segment data records correctly because the gap within a data record is typically smaller than that in between data section.

b) System identifies data records by analysing HTML tag trees or DOM trees. A tag tree is built by following the nested tag structure in the HTML code. However, we have to take care of missing or ill formatted tags. The visual or display information can be obtained after the web page is rendered by a Web browser, it also contains information about the hierarchical DOM tag tree structure.

2) In second step, grammar or more precisely pattern is generated using DOM tree and selectors from DOM Tree.

 

3) In third step, data is extracted from web pages using the grammar generated in second step.

Our three step approach called DEUDS (Data Extraction Using DOM Tree and Selectors). We propose a visual approach which identifies sections of data, and also a single data item, which is a basic content block in a data record. And also, our visual approach directly retrieve positional information and visual features of each item on the page, avoiding the need to interpret increasingly complex HTML source code and tag trees. Our proposed technique DEUDS presents number of advantages over existing systems.

a) Here, User can generate extraction rules with few mouse clicks.

b) Identifying data region using visual cues is very simple, because cost of comparing DOM tree is reduced.

c) It provides separation of extraction pattern generation and wrapper generation. This separation allows wrapper to use new extraction rules.

## III.LITERATURE REVIEW

Gibson D, Punera K, he has proposed the Volume and Evolution of Web Page Templates. According to Gibson et al. About 40%- 50% of the data on the Web are noises. In addition to noises, the heterogeneity of pages and demands for automation and efficiency make it difficult to extract contents from pages. If Web pages were written according to a common template, it could easily extract content simply by writing a regular expression. However, this quickly becomes impractical when dealing with hundreds of Web pages that are generated from different templates.

Laender, show the user inputs to the system a small set of example data objects to describe, from his viewpoint, what to extract . The system consists of two main modules called GUI and Extractor. The user uses the GUI module to assemble (in a hierarchical structure) a small set of example objects to be extracted, which are then used to generate object extraction patterns (OEPs). These OEPs exploit a combination of structural and textual information to extract new objects from new Web pages. The extractor module applies a bottom-up extraction algorithm that, given a set of Web pages as input, recognizes objects matching the generated OEPs and extracts these objects as an output.

## IV.PROPOSED WORK

The system DEUDS includes three components, a web page renderer which accepts an input Web page. After a web page is displayed using browser, DOM tree creator create DOM tree. Section selector divides web page into the Data sections, from which you can select particular record or whole data section. Pattern generator generates patterns based on data section selected. Here patterns generated are relative not absolute, so no need to worry about the change of structure of web page. The pattern generator includes a pattern generation from attributes, and a pattern validator. The pattern generator retrieves patterns discovered in a Web page[7] . The graphical user interface is used by users to view the data extracted by extraction rule. As user selects extraction rule conforming to his information desire, the extractor phase can use it to extract information from similar web pages.

Web Page Renderer is the first component, it performs three tasks.

(i) It accepts URL by user issues an http request and fetches corresponding document. This web page is used to derive grammar.

(ii) It cleans bad and ill formatted html tags.

(iii) It generates DOM tree from retrieved web page.

Section Selector divides input web page in to data section. Here we dividing page into different sections like list section, single valued section, multi valued section etc. it performs three tasks.

(i) Identifying data section in retrieved web page.

(ii) Identifying important semantic tokens and attributes and there logical path in DOM tree.

(iii) Identifying useful hierarchical structure.

Pattern Generator is responsible for generating extraction pattern to extract data of interest. It performs three tasks.

(i) It generates pattern from token and attributes retrieved in section selector.

(ii) Pattern is validated based on uniqueness of pattern in document.

(iii) Data is extracted using pattern.

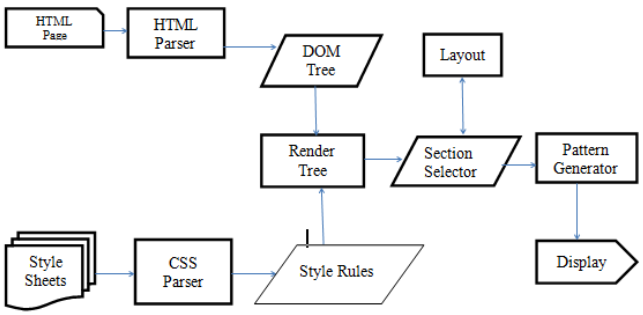


**Fig.1** System Architecture

Fig 1 shows the overview of the system. Web page from which we want to extract data is given as input to DEUDS. Web page renderer visualize input page from which we can see the various features of page or more precisely different region containing data to be retrieved. Section selector selects data region called section using visual cues like boundary, width etc. when we select section, pattern generator create pattern by using selectors. Data extractor uses pattern derived by pattern generator to extract data and store this data in the database.

 

**Web Page Renderer**

The system DEUDS includes three components, a web page renderer which accepts an input Web page. After a web page is displayed using browser, DOM tree creator create DOM tree. Section selector divides web page into the Data sections, from which you can select particular record or whole data section. Pattern generator generates patterns based on data section selected.

**URL Acceptor**

Your Here we are giving URL of page. When browser receives URL, browser sends http request to access requested document from web. We know that web page is made of the tag tree and the Cascading Style Sheet (CSS) of the page. A layout engine generates page from nodes of tag tree, according to the styles contained in the CSS. This process, called rendering, draws a rectangular box around the minimum boundary of each visible node on the page. We refer to each box as a visual block. The position of each visual block is represented by its four borders in the four directions on the two-dimensional plane. The outer block contains many inner blocks. The inner block which does not contains any further inner blocks is called basic block, which may contains data value.

**Repairing Ill Tags**

As soon as document is fetched, process of repairing bad or ill formatted tag begins. This process inserts missing tags, removes useless tags e.g. tag starting with !pr is end tag having no start tag. It also checks proper nesting of all tags. This process of cleaning document is applicable to all html pages.

**Building DOM Tree**

After bad and ill formatted tags are removed from web page source code, we can use this code to build DOM tag tree. Each html element consists start tag, optional attributes, optional embedded content, and end tag. DOM API is used to construct the tree for web page. Each page contains zero or one doc type nodes, one root element node, and zero or more comments or processing instructions; the root element serves as the root of the tree for the page. Parser converts source document into syntactic token, from this token tree is generated.

**Section Selector**

This section focuses on the segmenting the Web page to identify individual data section. In this step we do not extract any data records. When we select section, simultaneously attributes of selected sections are also retrieved from DOM tree. Here we constructed DOM tree of rendered page using SWT. Data section may contain many basic blocks, which actually contains data values. We designed java script to highlight the selected data section or selected basic block.

**Pattern Constructor and Data Extractor**

This section focuses on constructing pattern. DOM tree nodes are classified in about 12 types as attribute, element, text, data section, entity reference, entity, comment, document, document type, processing instruction, document fragment, and notation. Out of which we are considering attribute node only to extract data. Node type of attribute node is attribute, node name returns attribute name, and node value returns attribute value. We are using attribute nodes as a CSS selector, to construct pattern. There are various types of CSS selectors like Universal selector, attribute selectors, descendant selectors, type selectors, child selectors, adjacent sibling selectors, id selectors, and class selector.

**Web Page Extraction**

The main aim of the system is to have a data which is simpler to read to user. For that system is going to first extract the necessary data into database and then using different queries it produces the analyzed data. The system extracts data from web pages for that it uses different algorithms like line picker, boundary extractor, pattern generator, etc.

## V. ALGORITHM

Input: Web pages.
Output: Extracted structured data in database.

1. Identify web page of which analysis is to be made.
2. Get HTML response of that web page.
3. Divide data using HTML tags.
4. For each line remove the HTML tags.
5. Boundary extractor: Remove header and footer contents which are not necessary.
6. Pattern generator: Match the structure with rules.
7. Extract text mode design schema in database.
8. Using different queries data to be analyzed is presented in tabular and graphical format.

## VI. CONCLUSION

World Wide Web is a source of information where large amount of data is stored. These web pages mainly consist of noisy data. Extracting useful information from these web pages is very complex task. For this we are proposing DEUDS system solve problem of page-level data extraction. There is three stages web page renderer, Section selector, and Pattern generator. Content extraction approach that uses DOM tree structure to represent the data in better format. The system will extract the content dynamically from the different structured web pages such as blogs, forums, articles etc. In future work, we plan to extend our approach to extract hidden data, and to automatic label extracted data.

## REFERENCE

1. L. Liu , C. Pu, and W. Han, “XWRAP: An XML-Enabled Wrapper Construction System for Web Information

 

Sources," Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE), San Diego, California, 2000, pp. 611-621.

2. Gibson D, Punera K, Tomkins A. The volume and evolution of web page templates. In: Proceedings o f WWW'05. New York, NY, USA, 2005: 830-839.

3. NoDoSE (Northwestern Document Structure Extractor): An interactive tool for extracting data from semi-structured documents (plain text or HTML pages) [Adelberg, 1998].

4. Shine N. Das, Pramod K. Vijayaraghavan, Midhun Mathew, " Eliminating Noisy Information in Web Pages using featured DOM tree", International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868.

5. S. Mythili, T. Vetriselvi," Analytics of Noisy Data in Web Documents Using a Dom Tree", International Journal of Advanced Research in Computer Science and Software Engineering.

6. A. Sahuguet and F. Azavant , "Building intelligent Web applications using lightweight wrappers," Data and Knowledge Engineering 36(3): 283-316, 2001.

7. Vinayak B. Kadam , Ganesh K. Pakle ,"DEUDS: Data Extraction Using DOM Tree and Selectors", International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 1403-1410.

8. Vivek D. Mohod, Mrs. J. V. Megha,"A Survey on Data Extraction of Web Pages Using Tag Tree Structure", International Journal of Computer Science and Information Technology.

9. Laender, 2002b; Ribeiro-Neto,"DEByE (Data Extraction By Example): An interactive data extraction system [ 1999]."

10. Appukuti Chandrashekhar,Dr. P. Venkata Subba Readdy, " Html Tag Based Web Data Extraction and Tree Merging From Template Page", International Journal of Advance Research in Computer Science and Management Studies.

11. Chia-Hui Chang1, Mohammed Kayed2, Moheb Ramzy Girgis3 And Khaled Shaalan, " Criteria For Evaluating Information Extraction Systems"

 