ONLINE VOTING SYSTEMS VULNERABILITIES AND DEFENSES

***Lavanya.V*[1*]*, N.Surya Prakash Raju*[2*]**
1. M.Tech (CSE) Student, Dept of CSE, Aditya Engg College, Surampalem, Dist: E.Godavari, AP, India
2. Sr.Asst.Prof, Dept of CSE, Aditya Engg College, Surampalem, Dist: E.Godavari, AP, India

Keywords:
Voting System, Security, Internet voting, Transparency

Abstract: *A long-standing goal of Web research has been to construct a unified Web knowledge base. As of our technique utilizes both the visual content features on the result page as displayed on a browser and the HTML tag structures of the HTML source file of the result page. This differentiates our technique from other competing techniques for similar applications. Our experimental results indicate that our technique can achieve high extraction accuracy. In the future, we plan to utilize additional visual features (such as font type and color) to further reduce the reliance on HTML tag structure. Hence in this paper, we likely to give the template based structured data may lead to high performance.*

1. INTRODUCTION

Structured data collected in the form of Hyper Text Markup Language is not designed for structured data extraction at the first place, but mainly for data presentation. HTML pages are usually ‘dirty’, may have improper closed tags, improper nested tags, and incorrect parameter value of a tag are examples of such problems. In order to reduce parsing error over ill formed web pages caused by loose HTML standard, World Wide Web Consortium recommend stricter standard Markup Languages, such as XHTML and XML. However, parser of search engine still needs to cope with existing web pages that do not conform to the new standards. The embedded data is crawled together with the HTML pages by Google, Microsoft and Yahoo!, which use the data to enrich their search results. These companies have so far been the only ones capable of providing insights into the amount as well as the types of data that are currently published on the Web. While a previously published study by Yahoo! Research provided many insights, the analyzed web corpus not publicly available. This prohibits further analysis and the figures provided in the study have to be taken at face value. In the process of extracting data from web pages that share a common schema for the information they exhibit, and share a common template to encode this information, is significantly different from the extraction tasks that apply to unstructured (textual) Web pages. While the former harvest the exhibited data mainly by relying on properties w.r.t. the common features, the latter usually work by means of textual patterns and require some initial bootstrapping phase. Since a HTML document is based on nested tags it can be interpreted as a tag tree or a DOM (Document Object Model) tree. Many early systems work solely with string representations, but their methodology could nevertheless be abstracted into tree structure manipulation. Even though tree representation requires the code to be strictly structured, the web pages don’t always require the source to be in valid XML to be displayed correctly. There are however tools that correct the code from common errors. Hence, the tool sometime may useful in extracting template of similar matching having non homogenous document or pages.

*** Lavanya.V**
M.Tech (CSE) Student, Dept of CSE, Aditya Engg College, Surampalem, Dist: E.Godavari, AP, India

2. RELATED WORK

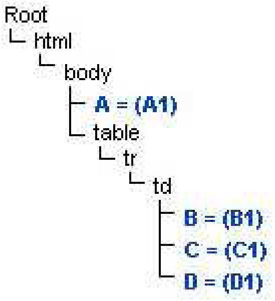
There are huge data available for extraction of data form web page, A good aspect of extracting information from web pages is that even if the page is badly formatted and contains sloppy code, it almost always has some sort of underlying structure. The HTML code itself promotes it with the use of <table> or <div>-tags. The rows and columns might be packed with non-consistent information and aesthetic layout tags, but they're at least confined in a table.

In the context of writing this paper we found various research groups have focused on the problem of extracting structured data from HTML documents. Much of the research is in the context of a database system, and the focus is on wrappers that translate a database query to a Web request and parse the resulting HTML page. Our focus is on template based data extraction: where target Web sites, extracting structured data, performing domain-specific feature extraction and resolution of missing and conflicting data, and making the data available to local database applications.

Example: (Template of html Code)

```
<Html>
    <Head>
        <Title>Page My name</title>
   </Head>
   <body bgcolor="green">
       <a href="http://www.some url./">LINK</a>
   </body>
</Html>
```

Furthermore discussing about the spanned tree structure of Html based Extraction.



The root and followed by html head and body in a spanned shoes the extraction is easy if structure it a form of tree. As of the html page goes on integration becomes complex. Hence final task in Web data extraction is to integrate data from multiple, related Web pages. There are two reasons why this is necessary. First, some Web sites use HTML frames for layout, which breaks up a logical data unit into separate HTML documents. Second, some Web sites break up the data across multiple "sibling pages" so as not to overload a single page with too much information. For instance, Yahoo! Finance contains comprehensive financial information on each company in its database, but each of their Web pages contains only a fraction of the data. The concept proposed to give rise to template having high complexity. The tag based template may not match to another which may similar but tags are same. Hence as of we go further to analyze in which and how much extend the concept is useful it's become as tedious task for the client based user.

3. METHODS

As of the techniques on web information extraction are based on the analysis of HTML tag structures. We believe that regularities in visual content (strings, images, etc. as shown on web pages) should also be utilized to achieve higher performance. Many visual content features that are designed to help people locate and understand information on a web page can help information extraction i.e. the structured data based ion template extraction. For example, the profile (or contour) of the left side of each SRR on the same result page tends to be very similar to each other, there are visual separators (e.g., blank lines) between consecutive SRRs, all SRRs tend to be arranged together in a special section on the result page, and this section occupies a large portion on the result page, and it also tends to be centrally located on the page. We describe some basic visual content features that are used in this study in the following sub-sections.

The first task is to locate the data-rich regions of the web page. Product web pages often contain lots of information that isn't interesting for the extraction, like advertisement, navigation links, company information etc. Even though this information is useful for the browsing user, it has a tendency to complicate the extraction process and there are various methods on how to solve this.

As of some templates are hidden; this hidden template must be found in order to create the wrapper or to automatically extract the information. Since many systems just do a syntactical extraction of the data, they tend to miss the information that is structurally implied by the page layout. Some systems solve this by extracting nested structures and derive this by aligning data tables when the extracting process is completed. In some systems exclude this part; this process is an important task when we are extracting information into an already known database schema.

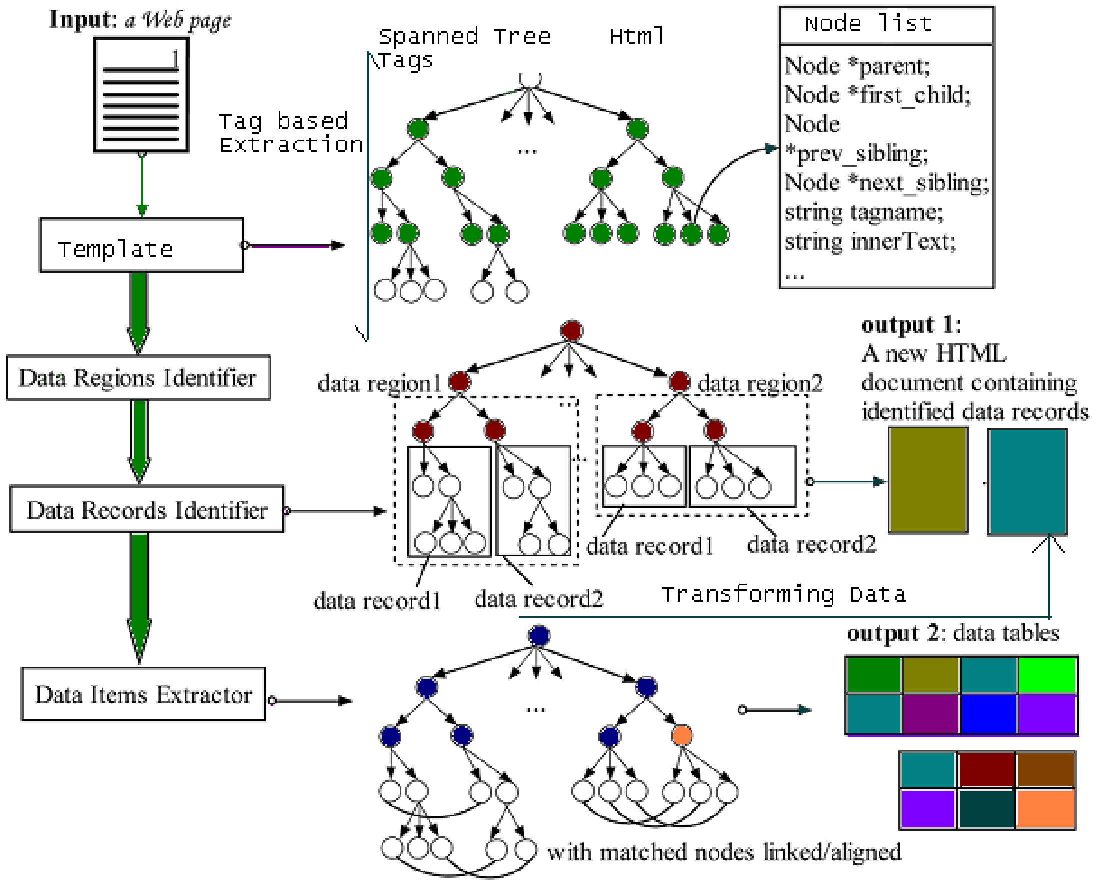


Fig. 3.1 Showing Architecture of Extraction of structured data from Html template Based Page.

In the phases of Extracting structured data from Web sites requires following distinct phases: finding target HTML pages on a site by following hyperlinks(tag based html page), extracting the template, extracting relevant pieces of data from these pages, distilling the data and improving its structure, ensuring data homogeneity (data mapping), and merging data from separate HTML pages (data integration). We discuss each problem in the following sections.

GENERATING TEMPLATE EXTRACTION:

Template depends upon the page where extraction based on structural tags. The semantic clues given by the user is used to generate the extraction template. Firstly we can make some interesting conclusions based on earlier assumptions. As disjunctions are omitted, each type of variable, constant or wildcard will share the same unique path from the root to the leaf. As each type of variable, constant or wildcard have a unique path, every D-node in the input document that doesn't share the same path can't belong to any of the selected types. Thus, it can be removed in order to improve and simplify the process, which is an effective way of locating data-rich regions. It doesn't mean that all remaining nodes should be interpreted as any of the labeled D-nodes. Some D-nodes share the same path as labeled D-nodes but they reside outside any repetitive patterns inferred by the node inference process.

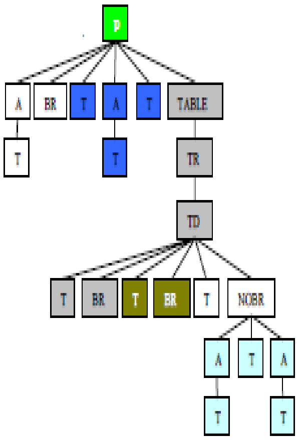


Fig. 3.2 showing the node based Spanning tree template Extraction on structure basic

In the above figure 3.2 we will make the following definitions. A data tree (D-tree) is the original tree structure parsed from the document. An intersection template tree (I-tree) is a tree generated by merging all selected D-nodes into a minimal tree only consisting of merged path-ways from the root to each selected D-node. This I-tree describes each labeled D-node and can be used to make an intersection between it and the original D-tree, to sort out and discard non-interesting information. Figure 5.4 exemplifies a labeled D-tree with the corresponding I-tree in figure 5.5. A match tree (M-tree) is the final pattern used to extract the data, which consist of a regular expression-type tree that describes repeated patterns. Since the I-tree shows the merged path from the root to each labeled D-node, the goal now is to convert this I-tree into a complete M-tree. An I-tree can be generated in linear time, but as illustrated by the example the variables B, C and D share the same path, and thus the I-tree alone isn't enough to distinguish the variables from each other.

4. CONCLUSION

Web based integration and extracting information from the template is the reverse engineering process. In regards of the system in context it is now clear that the extraction part is just a small part of the solution. It has been revealed during the testing that the navigation is a far more complex issue than just following links specified in the tags. Many documents are packed with JavaScript code that overrides the reference and uses functions to perform HTTP-post requests with individually defined parameters. Likely to be these function must sometimes be interpreted semantically since the script code formats the parameters before they are sent to the server. The process of navigating and retrieving the documents has been proved to be as time consuming as writing wrapper programs manually. In this we have considered the template based extraction which makes easy to structure the data as of the requirement of the page. The evolvement of new technologies on the web makes the navigation far more complex, especially when considering Web 2.0 and Ajax-technologies. The representation of the information is also beginning to grow in complexity as many companies fill their data tables with scripts to add effects to their representation. In the further step of this paper it to be good enough to take the non homogenous spanning matched template to extract the personalized deep content.

5. REFERENCE

[1] Xwrap: An xml-enabled wrapper construction system for web information sources. In ICDE '00: Proceedings of the 16th International Conference on Data Engineer-ing, page 611, Washington, DC, USA, 2000. IEEE Computer Society.

[2] A fully automated object extraction system for the world wide web. In ICDCS '01: Proceedings of the The 21st International Conference on Distributed Computing Systems, page 361, Washington, DC, USA, 2001. IEEE Computer Society.

[3] Arvind Arasu, Hector Garcia-Molina, and Stanford University. Extracting structured data from web pages. In SIGMOD '03: Proceedings of the 2003 ACM SIG-MOD international conference on Management of data, pages 337–348, New York, NY, USA, 2003. ACM Press

[4] Robert Baumgartner, Michal Ceresna, and Gerald Ledermuller. Deepweb navigation in web data extraction. In CIMCA '05: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Inter- net Commerce Vol-2 (CIMCA-IAWTIC'06), pages 698–703,Washington, DC, USA, 2005. IEEE Computer Society.

[5] Www. wikipedia.com

[6] C.-H. Chang, C.-N. Hsu, and S.-C. Lui, "Automatic Information Extraction from Semi-Structured Web Pages by Pattern Discovery," Decision Support Systems, vol. 35, no. 1, pp. 129-147, 2003.

[7] V. Crescenzi and G. Mecca, "Grammars Have Exceptions," Information Systems, vol. 23, no. 8, pp. 539-565, 1998.

[8] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 109-118, 2001.

[9] D.W. Embley, Y.S. Jiang, and Y.-K. Ng, "Record-Boundary Discovery in Web Documents," Proc. ACM SIGMOD, pp. 467- 478, 1999.

[10] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krpl, and B. Pollak, "Towards Domain Independent Information Extraction from Web Tables," Proc. Int'l World Wide Web Conf. (WWW), pp. 71-80, 2007.

[11] J. Hammer, J. McHugh, and H. Garcia-Molina, “Semistructured Data: The TSIMMIS Experience,” Proc. East-European Workshop Advances in Databases and Information Systems (ADBIS), pp. 1-8, 1997.

[12] http://research.microsoft.com/pubs/77622/extract.pdf

[13] http://www.w3.org/People/Raggett/tidy/.

[14] Mary Tork Roth and Peter Schwartz. Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources. Proc. International Conference on Very Large Data Bases (VLDB), Athens, Greece, August 1997.

[15] Compaq’s Web Language, Compaq Computer, http://www.research.digital.com/SRC/WebL/index.htm.

[16] Web Interface Definition Language, W3C Note. September 1997. http://www.w3.org/TR/NOTE-widl.

[17] XHTML: The Extensible HyperText Markup Language, W3C Recommendation, January 2000. http://www.w3.org/ TR/xhtml1.

AUTHORS :

Mr.N. Surya Prakash Raju, well known & excellent teacher Received M.Tech (CSE) from JNTU, Kakinada is working as Sr.Asst.Professor, Department of CSE at Aditya Engineering College. He has 10 years of teaching experience in various Engineering Colleges. To his credit couple of publications both national and international conferences /journals. His area of Interest includes Computer Networks, Information Security, Soft Computing Techniques and other advances in computer Applications. And guided many Projects.

Lavanya.V M.Sc (cs),
M.Tech(CSE)pursuing in Aditya Engineering College,
Surampalem, East Godavari Dt.