ISSN:0975-9646

A Survey on HTML Structure Aware and Tree Based Web Data Scraping Technique

Vinayak B. Kadam , Ganesh K. Pakle

Department of Information Technology,
SGGS IE & T, Nanded,
Maharashtra, India-431606

***Abstract*— Vast amount of information is available on web. Data analysis applications such as extracting mutual funds information from a website, daily extracting opening and closing price of stock from a web page involves web data extraction. Huge efforts are made by lots of researchers to automate the process of web data scraping. Lots of techniques depends on the structure of web page i.e. html structure or DOM tree structure to scrap data from web page. In this paper we are presenting survey of HTML aware web scrapping techniques.**

***Keywords*— DOM Tree, HTML structure, semi structured web pages, web scrapping and Web data extraction.**

I. INTRODUCTION

With the explosive growth of World Wide Web, vast amount of data is offered on line. There is number of data analysis applications such as extract competitor's price list from web page regularly to stay ahead of competition, extract data from a web page and transfer it to another application, scrape tabular data from the Web and transfer it to Excel, extract people's data from web page and put it in a database, extract opening and closing price of stock from a web page, extract mutual funds information from a website daily, extract tabular data from the web and transfer it your own application, which requires web data scraping. There is lots of work in the field of web data extraction. There is number of techniques proposed in literature for web data scraping by number of researchers.

In this paper, we focus on number of techniques available for web data scraping, which uses knowledge of html structure or DOM tree structure.

The rest of the paper is organized as follows: Related work is presented in Section 2. Section 3 presents overview of each html aware techniques and finally, Section 4 concludes the paper.

II. RELATED WORK

In the last few years, number of approaches for web data scraping are proposed these includes machine learning and pattern mining techniques, with different degrees of automation. In this section we presented, survey of previously proposed classification for web data extraction tools made by the number of researchers.

Laender classified data extraction tools based on the main technique used by each tool to generate a wrapper [1]. They grouped tools as: languages for wrapper development, Ontology-based tools, NLP-based tools, Modeling-based tools, Wrapper induction tools and HTML-aware tools. In languages for wrapper development group they listed tools which are developed by specific language designed for wrapper generation. Example of such tools is Minerva [2], TSIMMIS [3] and WebOQL[4]. In HTML- aware group they listed tools which use structure of html document to extract data. These tools uses parsing tree to accomplish extraction, example of such tool is W4F [5], XWRAP [6] and RoadRunner [7]. NLP-based tools uses NLP-based techniques such as filtering, lexical semantic analysis to derive relationships and extraction rules. Examples in this category are WHISK[8], RAPIER[9] and SRV[10]. In Wrapper Induction tool category, tool which generate extraction rules from set of training examples are grouped, some tools in this category are WIEN[11], SoftMealy[12] and STALKER[13]. Other tools are grouped as Modeling-based tools e.g. NoDoSE [14] and DEByE [15],[16], and Ontology-based tools e.g., BYU [17].Laender used following 7 characteristics to compare tools: XML output, degree of automation, page contents, support for non-HTML sources, availability of a GUI, resilience, support for complex objects and adaptiveness.

Another survey on web data extraction is by Chang, Kayed, Girgis, and Shaalan. They used three dimensions as follows: Task difficulties, The Techniques Used, and Automation Degree for comparing information extraction system. They grouped information extraction systems into four categories as follows: manually-constructed information extraction Systems, supervised information extraction Systems, semi-supervised information extraction Systems and unsupervised information extraction Systems. In manually-constructed information extraction Systems they grouped tools which are developed by general purpose programming languages. Examples in this category are TSIMMIS[3], Minerva[2], Web-OQL[4], W4F[8] and XWRAP[6]. Supervised WI systems take a set of web pages labeled with examples of the data to be extracted and output a wrapper. Some examples in this category are SRV[10], RAPIER[9], WHISK[8],WIEN[11], STALKER[13], SoftMealy [12], NoDoSE [14] and DEByE [15] [16]. In Semi-Supervised information extraction systems takes rough example from user to generate rules. Some systems in this category include IEPAD[19], OLERA[20] and Thresher[21]. In Un-Supervised information extraction systems do not use any labeled training examples and have no user interactions to generate a wrapper. Some examples in this category are RoadRunner[7], EXALG[18], DeLa[23] and DEPTA[22]. RoadRunner and EXALG, works on page-level extraction task, while DeLa and DEPTA works on record-level extraction task.

In next section we will see the details of tools which are depend on the HTML structure of web page and DOM tree.

III. HTML STRUCTURE-AWARE AND TREE BASED TECHNIQUES

In this survey we grouping tools which are depend on HTML structure and DOM tree structure to extract data. HTML-aware: This type of tools relies on structural features of HTML documents for data extraction. A transformation of the source document to a parsing tree is realized and it reflects its HTML tag hierarchy. Therefore, extraction rules are generated either semi automatically or automatically and then applied to the tree.

A. RoadRunner

RoadRunner [7] uses HTML structure of web page to generate wrapper. Model of page creation is used in RoadRunner. RoadRunner assumes site generation is process of encoding the database content into HTML code. As a consequence, data extraction is considered as a decoding process. As a result, generating a wrapper for a set of HTML pages corresponds to inferring a grammar for the HTML code. RoadRunner compares given input pages belonging to same page class, to infer template of page. RoadRunner begins with the input page as its initial template. Then, It checks for each subsequent pages that page can be generated by the current template. If it is unable to match template, it modifies its current template. Extraction process in RoadRunner is based on comparison of tag structure of sample pages.

B. W4F

W4F (Wysiwyg Web Wrapper Factory) is a Java toolkit to generate Web wrappers [5]. Wrapper development process in W4F consists of three independent steps: retrieval, extraction and mapping step. In the retrieval phase, document to-be processed is retrieved and given as input to parser that constructs a parse tree. In the extraction phase, extraction rules are applied on the parse tree to extract information. Mapping phase is used to map extracted data to NSL structures.

C. XWRAP

XWRAP [6] data extraction consist of four components namely structure normalization, data extraction, code generation, program testing and packaging. Structure normalization performs three tasks as it reads input page, Cleans bad and ill formatted tags, and converts page into syntactic token tree. In information extraction module extraction rules are derived. Code is generated using extracted rules. Last module validates the wrapper. In summary, wrapper generation process includes two phases: structure analysis, and source-specific XML generation. First, XWRAP reads, and generates a tree-like structure of the web page. Then system identifies data regions. In the second phase, the system generates a XML template file, and then constructs a source-specific XML generator, XWRAP.

D. IEPAD

IEPAD [19] generates extraction patterns from unlabeled Web pages. This method assumes that if a Web page contains multiple data records to be extracted, they are displayed regularly using the same template. If the page is well encoded repetitive patterns can be found. In IEPAD PAT trees data structure is used, which is a binary suffix tree to discover repetitive patterns in a Web page. IEPAD consists of three components, an extraction rule generator takes input Web page, a GUI, called pattern viewer, which shows repetitive patterns discovered, and last data extractor, extracts information needed from similar Web pages according to the rule. HTML tag structure is very important in IEPAD to generate rules. Techniques of pattern mining are implemented in the rule generator. Rule generator consists of a translator, a pattern tree constructor, a pattern finder, validator, and an extraction rule composer. The output of rule generator is extraction rules discovered from Web page. The GUI enables users to view the information extracted by each extraction rule. Then user selects extraction rule conforming to his information needed, extractor module uses this rules to extract information from other pages having similar structure with the input page.

E. FiVaTech

FiVaTech [27] assumes data extraction problem as the decoding process of page generation based on structured data and tree templates. FiVaTech consist of two modules as tree merging module and schema detection module. The first module converts input web pages into DOM tree and then merges all input DOM trees into a structure called fixed/variant pattern tree. In the second module fixed/variant pattern tree is used to detect the template of the Website. FiVaTech assumes that data instances of the same type have the same path from the root in the DOM trees of the input pages. FiVaTech then performs four steps to extract data as follows:

I. peer node recognition
II. Matrix alignment
III. Pattern mining, and
IV. Optional node detection.

F. DELA

DELA (Data Extraction and Label Assignment for Web Databases) [23] uses two consecutive steps to generate wrappers. First, Data-rich Sections are identified from Web pages by comparing the DOM trees for two Web pages. Second, repeated patterns are found using suffix trees. The input to the wrapper generator is collection of pages. It produces regular expression based on HTML tag structures of the page. Wrapper generator considers each page as a sequence of tokens composed of HTML tags. Text string enclosed within HTML tag pairs presented using special token “text”. Repeated HTML tags are then extracted and regular expression wrapper is derived from the repeated substrings Wrapper generation consist of Data-rich section extraction, C-repeated pattern, Optional attributes and disjunction, Data Alignment. Data aligner works in two consecutive steps data extraction and attributes separation. In first step data is extracted from web pages by using the

wrapper produced by wrapper generator. The basic assumption of the attribute separation phase is that if several attributes are encoded into one string, then there must be special symbol in the string as the separator to visually support users to separate the attributes.

G. DEPTA

DEPTA (Data Extraction based on Partial Tree Alignment) [22]: DEPTA founds repeated substring by comparing only adjacent substrings with starting tags having the same parent in the HTML tag tree. In DEPTA single page containing lists of data records is used to extract data. DEPTA consist of following four components Building HTML tag tree, Mining data region, Identifying data records, and Data item extractor. In first step DOM tree is constructed by finding four boundaries of rectangle of each HTML tag. Data region mining step finds the data region by comparing tag strings. Similar nodes are labeled as data region. Generalized node is used to denote each similar node. Neighbour generalized nodes form a data region. Gaps between data records are used to eliminate false node combinations. From data region data records are identified from generalized nodes. Data items are extracted based on partial tree alignment technique. Two steps are performed in data extraction, first is production of one rooted tag tree for each data record; Sub trees of all data record are arranged into a single tree, and second is Partial tree alignment: Tag trees of data records in each data region are aligned using partial alignment. Number of tag trees of multiple data records is needed to be aligned in order to extract data.

H. ViPER

ViPER (Visual perception based Extraction of Records) [25] ViPER is a fully automated information extraction tool which works on the web page containing at least two consecutive data records which exhibits some kind of structural and visible similarity. ViPER extracts relevant data with respect to user's visual perception of the web page. Then Multiple Sequence Alignment (MSA) method is used to align these relevant data regions. ViPER uses both visual data value similarity features and the HTML tag structure to first identify repetitive patterns. ViPER is a two-step process i.e. Data Extraction and Data Alignment. In Data Extraction HTML document can be viewed as labelled unordered trees. Data Extraction consists of following sub steps

i. Preprocessing
ii. pattern search
iii. primitive tandem repeats
iv. identifying data regions and record,
v. visual data segmentation
vi. visual data region weighting.

I. ViNTs

ViNTs (Visual information and Tag structure based wrapper generator) [26] it uses both visual and tag structure features to generate a wrapper from a set of training pages from a website. It first uses the visual data value similarity to identify data value similarity regularities, referred as data value similarity lines. Wrapper is generated by combining data value similarity lines and HTML tag structure regularities. Both visual and non-visual features are used to weight the relevance of different extraction rules.

J. CTVS

CTVS (Data Extraction and Alignment using Combining Tag and Data Value Similarity) [24] is a novel approach that uses Tag and Data Value similarity, to automatically extract data from query result pages. It extracts data by identifying and segmenting the query result records (QRRs) in the result pages. Then align the segmented QRRs in a table. In table the data values of the same attribute are put into same column. This method also handles QRRs which are not contiguous, as query result page may contains comment, advertisement, navigational panels. CTVS consists of following two-steps, to extract the QRRs from a query result page.

1. Record extraction: It identifies the QRRs (Query Result Records) in result page and involves following steps:

   i. Tag tree construction
   ii. Data region identification
   iii. Record segmentation
   iv. Data region merge
   v. Query result section identification

2. Record alignment: It aligns the data values of the QRRs from query page into a table so that data values for the same attribute are aligned into the same column. QRR alignment is performed by three-step data alignment method that combines tag and value similarity.

   i. Pairwise QRR alignment
   It aligns the data values in a pair of QRRs.
   ii. Holistic alignment
   It aligns the data values in all the QRRs.
   iii. Nested structure processing
   It identifies the nested structures that exist in the QRRs.

K. Mining Data Records in Web Pages

This technique [28] is proposed by Liu, Grossman, and Zhai to extract data from web page. They proposed three step solution to web data extraction.

1. Building the HTML Tag Tree
2. Mining Data Regions
   This step mines every data region that contains similar data records from web page. Instead of first extracting data records, they extracted generalized nodes in a page. A sequence of adjacent generalized nodes forms a data region. Then from each data region, actual data records are identified
   i. Comparing Generalized Nodes
   ii. String Comparison Using Edit Distance
   iii. Determining Data Regions.
3. Identify Data Records
   In this step data records from data region are mined.

L. MSE

MSE (Multiple Section Extraction) [29] is algorithm used to generate wrapper. Sample result pages are given as input to MSE. The output of MSE is a wrapper or set of rules for extracting all dynamic sections as well as all search result record within page. They used structure of search result page.

IV. CONCLUSIONS

In this paper, we presented the survey on web data extraction techniques which uses HTML structure or DOM tree information to extract information from web page. Most of the techniques first clean the fetched web page by removing bad and ill formatted tags. Then these tools construct parsing tree of page, and this tree is used in different way to extract data from that page.

Some techniques uses DOM tree to extract repeated pattern, then this repeated pattern is used to extract data. While another technique uses tree directly e.g. comparing DOM tree to extract data. Some techniques use visual features along with HTML structure information to extract data.

To solve the problem of web data extraction use of HTML structure and DOM tree structure is very important.

REFERENCES

[1] A. Laender, F. H, B. Ribeiro-Neto, DA Silva and Teixeira, "A brief survey of Web data extraction tools," SIGMOD Record 31(2): 84-93,2002 .

[2] V. Crescenzi, and G. Mecca, "Grammars have exceptions," Information Systems, 23(8): 539-565, 1998.

[3] J. Hammer, J. McHugh, and Garcia-Molina, "Semistructured data: the TSIMMIS experience," In Proceedings of the 1st East-European Symposium on Advances in Databases and Information Systems (ADBIS), St. Petersburg, Rusia, pp. 1-8, 1997.

[4] G. O. Arocena and A. O. Mendelzon," WebOQL: Restructuring documents, databases, and Webs," Proceedings of the 14th IEEE International Conference.

[5] V. A. Saiiuguet, and F. Azavant, "Building intelligent Web applications using lightweight wrappers," Data and Knowledge Engineering 36(3): 283-316, 2001.

[6] 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, pp. 611-621, 2000.

[7] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: towards automatic data extraction from large Web Sites," Proceedings of the 26th International Conference on Very Large Database Systems (VLDB), Rome, Italy, pp. 109-118, 2001

[8] S. Soderland, "Learning information extraction rules for semi-structured Data Extraction," Master's thesis, Department of Computer Science, Federal University of Minas Gerais, Brazil, 2001.

[9] M. E. Calife, and R. J. Mooney, "Relational learning of Pattern-Match Rules for Information Extraction," In Proceedings of sixteenth national conference on artificial intelligence and eleventh conference on innovative application of artificial intelligence(Orlando, FL, 1999), pp. 328-334.

[10] D. Freitag, "Machine Learning for Information Extraction in Informal Domains," Machine Learning 39,2/3(2000) ,169-202.

[11] N. Kushmerick, "Wrapper induction: Efficiency and expressiveness," Artificial Intelligence Journal 118,1-2(2000),15-68.

[12] C.-N. HSU and M.-T. Dung, "Generating finite state transducer for semi-structured data extraction from the web," Information Systems 23,8 (1998) 521-538.

[13] I. Muslea, S. Minton, and C. Knoblock, "Hierarchical wrapper induction for semistructured information sources," Autonomous Agents and Multi-Agents Systems 4, 1/2 (2001), 93-114.

[14] B. Adelberg, "NoDoSE: A tool for semi-automatically extracting structured and semi-structured data from text documents," SIGMOD Record 27(2): 283-294, 1998.

[15] A. H. F. Laender, B. Ribeiro-Neto, and A. S. DA Silva, "DEByE Data Extraction by Example," Data and Knowledge Engineering, 40(2): 121-154, 2002.

[16] A. H. F. Laender, B. Ribeiro-Neto, and A. S. DA Silva, "Extracting semi-structured data through examples," Proceedings of the Eighth ACM International Conference on Information and Knowledge Management (CIKM), Kansas City, Missouri, pp. 94-101, 1999.

[17] D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, Y. Kai Ng, D. Quass, and R. D. Smith, " Conceptual-model-based data extraction from multiple-record Web pages," Data and Knowledge Engineering, 31(3): 227-251, 1999.

[18] A. Arasu, and H. Garcia-Molina, "Extracting structured data from Web pages," Proceedings of the ACM SIGMOD International Conference on Management of Data, San Diego, California, pp. 337-348, 2003.

[19] C.-H. Chang, and S.-C. Lui, "IEPAD: Information extraction based on pattern discovery," Proceedings of the Tenth International Conference on World Wide Web (WWW), Hong-Kong, pp. 223-231, 2001.

[20] C.-H. Chang, and S.-C. Kuo, "OLERA: A semi-supervised approach for Web data extraction with visual support," IEEE Intelligent Systems, 19(6):56-64, 2004.

[21] A. Hogue, and D. Karger, "Thresher: Automating the Unwrapping of Semantic Content from the World Wide web," Proceedings of the 14th International Conference on World Wide Web (WWW), Japan, pp. 86-95, 2005.

[22] Y. Zhai and B. Liu, "Web Data Extraction Based on Partial Tree Alignment," Proc. Int'l Conf. World Wide Web (WWW-14), 2005, pp. 76-85.

[23] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. Int'l Conf. World Wide Web (WWW-12), 2003, pp. 187-196

[24] W. Su, J. Wang, F. H. Lochovsky, and Yi Liu, " Combining Tag and Value Similarity for Data Extraction and Alignment", IEEE Trans. Knowledge and Data Eng., vol. 24, no. 7, pp.1186-1200, July. 2012.

[25] K. Simon and G. Lausen , "Viper: augmenting automatic information extraction with visual perceptions," In CIKM Conference, pages 381–388, New York, NY, USA, 2005.

[26] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, "Fully Automatic Wrapper Generation for Search Engines," Proc. 14th World Wide Web Conf, pp. 66-75, 2005

[27] Mohammed Kayed and Chia-Hui Chang, "FiVaTech: Page-Level Web Data Extraction from Template Pages," IEEE transactions on knowledge and data engineering, vol. 22, no. 2 , pp. 249-263, February 2010.

[28] B. Liu, R. Grossman, and Y. Zhai. "Mining data records in web pages," In SIGKDD conference, New York, NY, USA, 2003, pp. 601–606.

[29] H. Zhao, W. Meng, and C. Yu. "Automatic extraction of dynamic record sections from search engine result pages," In VLDB Conference, 2006, pp. 989–1000.