International Journal of Computer Science and Engineering Open Access
Survey Paper Volume-2, Issue-5 E-ISSN: 2347-2693

Web Data Scraper Tools: Survey

Sneh Nain[1*], Bhumika Lall[2]

[1*]*Computer Science Department, MDU University, India*
[2]*Faculty of Computer Science Department, MDU University, India*

www.ijcseonline.org

Received: 11/04/2014 Revised: 28/04/ 2014 Accepted: 14/05/May 2014 Published: 31/05/2014

***Abstract*—** World Wide Web contains a huge amount of information that is increasing rapidly. Usually data stored on the web are in unstructured and semi-structured form. In order to obtain the essential data from the web, certain data scraper tools had been invented. In this paper we intend to briefly survey Web Data Scraper Process, the taxonomy for characterizing Web Data Scraper Tools and provide qualitative analysis of them. Hopefully, this work will simulate other studies aimed at a more comprehensive analysis of data scraper approaches and tools for Web data.

***Keywords/Index Term*—**Wrapper; Scraper;Document Object Model(DOM)

I. INTRODUCTION

With the enhancement of the World Wide Web, large amount of data on many different subjects has been available online that provide certain benefits to users.

Webpage is a main source of data consists of certain parts which are not equally important. It usually contain the combination of the essential and un-essential information such as with the main content it also contain advertisement, boxes containing images or animated ads, navigational bars on top and/or on the side etc. Therefore, the need of automated and flexible web data scraper tools has to be introduced that separate the essential data from the non-essential data of the Webpage and transfer it into a meaning and useful structure. So eliminating them will be a saving in storage, timing and indexing.

Usually, users retrieve Web data by browsing or searching keywords, these are the general forms of accessing data on the Web. But, these search strategies have several drawbacks. Like browsing is not suitable for finding particular items of data because following certain hyperlinks are very tedious. Keyword searching provides large amount of data, far beyond the capacity of user to handle.

There are certain traditional approaches for extracting data from web sources. Computer Science scientific literature counts many valid surveys on the web data scraper problems. After obtaining the essential information from the web page, then generate the wrapper. In Information Technology, a wrapper is data that precedes or frames the main data or a program that sets up another program so that it can run successfully [1].Wrapper can be developed manually, automatically and by both. But developing wrapper manually has many well known shortcomings, mainly due to the difficulty in writing and maintaining them. Recently, many tools have been proposed to better address the issue of generating wrappers for web data scraper [2, 3, 4, and 5]. Such tools are based on several distinct techniques such as declarative languages, HTML structure analysis, natural language processing, machine learning, data modeling and ontologies.

As more and more tools for web data scraper continue to appear, the need for analysis of their capabilities and features arises. In this paper, we introduce a comprehensive review of different web data scraper tools that provide user only the essential information/data which is needed. Based on this survey, we can choose the suitable web data scraper tool that will be integrated in our future work.

Rest of the paper is organized as follows: in Section 2 we will take a look at related work, in Section 3 we track a complete profile of web data scraper systems, in Section 4 we show the taxonomy for characterizing Web Data Scraper Tools, Section 5 contain the complete overview of Web Data Scraper Tools description, Section 6 presents the qualitative analysis of web data scraper tools. Finally, Section 7 concludes including some recommendations, contributions and suggestion for future work.

II. RELATED WORKS

This work is a brief survey on the tools and problems faced by the user in extracting the data from the web. Computer Science scientific literature counts many researchers have valid surveys on the web data scraper problems:
Laender [Laender 2002], in 2002, he presented a notable survey about the taxonomy to classify web data extraction systems. They introduced certain sets of criteria and qualitative analysis of tools.
Kushmerick [Kushmerick 2002] provide the complete profile of finite-state transducers approach of the problem. After the development of tools, survey is also required so Kuhlins in 2003 surveyed tools for generating wrappers. Till now information is not completely updated but analysis was required for further research. After this Flesca in 2004 and Kaiser and Miksch in 2005 did survey on wrapper induction

Corresponding Author: *Sneh Nain*

problem. Latter in particular modeled a representation of Information extraction system architecture.
Chang [2006] defined a tri-dimensional categorization of Web Data Extraction systems, based on task difficulties, techniques used and degree of automation. New Tools were developed by Fiumara in 2007. Certain other research and analysis were performed in 2008 and 2009. Baumgartner [2009] provide a short- survey on state of the art of the discipline. In 2010 Tanaka et al provide ontology extraction from table on the web. Further researches are also going on in this field.

III. WEB DATA SCRAPER SYSTEM

Before knowing what is web data scraper system? We have to know what is Scraper? Scraper is a tool or device used for scraping, especially for removing dirt or unwanted material from a surface. In general term, scraper means extraction. Now we can define a Web Data scraper system, it provides the sequence of procedures that extract information from Web sources/page.
We obtain the information from the two fundamental aspects:-

- Interaction with the Web Pages
- Generation of a Wrapper
- Storing in Repository

Baumgartner [2009] define a web data extraction system as "a software extracting, automatically and repeatedly data from Web pages with changing contents and the delivers extracted data to a database or some other application"[6].
There are five points required to solve the problem of Web Data Scraper:

3.1Interaction with Web Pages:

To extract the data from the system, first we required the Web Interaction. User can interact with the web through URL in order to obtain the information.
Some commercial systems, Lixto include a Graphical User Interface (GUI) for fully visually and interactive navigation of HTML Pages, integrated with data extraction tools. Extraction of data can also be done by deep web navigation, simulating the activity of users clicking on DOM elements of pages through macros or more simply, filling HTML forms. Information can also be extracted from Static Web Page and Dynamic Web pages.

3.2 Generation of a Wrapper:

As we defined earlier, wrapper provide a method of gathering the information from web pages and also convert the unstructured information into structured form of data. A Web Data Scraper system must implement the support for wrapper generation and wrapper execution.

3.3 Data Transformation:

Data can be obtained from various resources, which means from different wrappers that usually store data in different structured formats. Before delivering the extracted information, data cleaning step should be performed.
Most powerful Web Data extraction systems provide tools to perform automatic schema matching from multiple wrappers, then packaging data into a desired format to make it possible to query data, normalize structure and de-duplicate tuples.

3.4 Storing in a Repository:

Data obtained after data transformation step should be stored in repository. So that user can use it whenever required more quickly.

3.5 Use of Extracted Data:

When the extraction task is completed, and acquired data are packaged in the needed format, this information is ready to be used; the last step is to deliver the package, now represented by structured data, to a managing system.

IV. TAXONOMY FOR CHARACTERIZING WEB DATA SCRAPER TOOLS

This section shows the main technique used by each tool to generate wrapper. Tools used in data scraper process are classified as:-

- Wrapper Development Languages
- HTML Aware Tools
- NLP Based Tools
- Wrapper Induction Tools
- Modeling Based Tools
- Ontology Based Tools

4.1 Languages for Wrapper Development:

First step required to solve the problem of wrapper generation was the development of languages that specially designed to assist users in constructing wrappers. Extraction system relied on standard scripting languages or general purpose languages to create the environment for wrapper generation. Some of the tools are Minerva, TSIMMIS, and Web-OQL.

4.2 HTML Aware Tools:

These tools are depends on the *intrinsic formal structure* of the HTML document to extract data. Firstly these tools build a DOM tree from HTML tags. DOM tree show the hierarchy structure of HTML Tag. Extraction rules are generated semi-automatically or automatically. Some of the tools occurred in this category are W4F, XWRAP, Road Runner.

4.3 NLP Based Tools:

Natural Language Processing tools are developed for information extraction. These techniques have been used by several tools to learn extraction rules for extracting relevant data existing in natural language documents. Extraction rules are based on syntactic and semantic constraints that help to identify the relevant information within a document. Tools lie in this category are RAPIER, SRV, WHISK. These tools are generally used to solve specific problem of data extraction like extraction of facts from speech transcriptions in forums, email messages, newspaper articles etc.

4.4 Wrapper Induction Tools:

Delimiter based extraction rules are generated by Wrapper Induction Tools. Rules are derived from a given set of training examples or from formatting features. Basic difference between NLP tools and Induction tool is that it is does not based on linguistic constraints, but rather in formatting features. Representative tools are WIEN, SoftMealy, and STALKER.

4.5 *Modeling Based Tools*:

These tools can find one or more objects in the page matching the primitive items. It means identify object from the target structure, try to locate in Web pages portions of data that implicitly conform to that structure. Tools which adopt this approach are NoDOSE, DEByE.

4.6 Ontology Based Tools:

Till now all the techniques are depend on page structure or presentation feature of the data but this technique directly depend on the data. Ontology can be applied successfully on specific well-known domain applications such as social networks and communities. Representative tool of this approach is being developed by Brigham Young University Data Extraction Group which is BYU tool.

V. WEB DATA SCRAPER TOOLS: OVERVIEW

In this section, we provide the overview of the Web Data Scraper Tools that we have studied. We have tried to cover all tools that have appeared in the recent literature but list of the tools covered here must not be regarded as complete.

5.1 Languages for Wrapper Development:

5.1.1Minerva

Minerva tool allows researchers to work on large corpus of text by bringing together data visualization and text annotations. This tool is generally designed, for the development of wrappers. Minerva combines a declarative grammar-based approach with features of procedural programming languages. For each document, a set of productions is defined for the development of wrapper and with each production, exceptional clause is also added. Each production defines the structure of a non-terminal symbol of the grammar, in terms of the terminal symbol and other non-terminal symbols.

5.1.2 TSIMMIS:

TSIMMIS generate wrapper that can be prepared by the specification files written by the user. Specification files are composed by a sequence of commands that define extraction steps. Each command is of the form [variables, sources, and pattern] where source define the input document to be considered, pattern shows the matching of the data of interest within the sources and variable contain extracted result after performing matching with the source.

5.1.3 Web-OQL:

Web-OQL is a declarative query language that is capable of locating selected pieces of data in HTML pages. This technique generates the hyper tree by parsing the page, which is act as an input. Hyper tree is generated by writing the queries that locate the data of interest and then output these data in a suitable format.

5.2 HTML Aware Tools:

5.2.1W4F:

W4F stands for World Wide Web Wrapper Factory, is used for developing the wrappers. It generates the wrapper by retrieving the document from the web according to retrieval rule and then generates the DOM Model by feeding the document to HTML parser. Extraction rule should be defined in order to obtain the data from parsing tree and then extracted data should be stored using W4F internal format called NSL (Nested String List). W4F provide the wizard which help the user in generating the extraction rule.

5.2.2 XWRAP:

XWRAP stands for XML Enabled Wrapper. Through XML Enabled we mean that the metadata about the information content that are implicit in the original web page will be extracted and encoded explicitly as XML tag in the wrapped document. In this method query based content filtering process is performed against the XML documents. It provides semi-automatic generation of wrapper program. The tool features a component library that provides basic building blocks for wrapper, and a user friendly interface to ease the task of wrapper development.

5.2.3 Road Runner:

This HTML Aware tool extracts the data from the HTML sites through the use of automatically generated wrapper [7]. It produced the schema by comparing the HTML structure of two or more given sample pages belonging to the same page class. In order to capture accurate structural variation, it is highly recommended to provide at least more than two sample pages or as many sample as possible. This technique is based on algorithms that compare tag structure of the sample pages and generate regular expressions that handle structural mismatch found between the two structures.

5.3NLP Based Tools:

5.3.1RAPIER:

RAPIER stands for Robust Automated Production of Information Extraction Rules. This tool extracts the data from free text because NLP tools are generally used to solve specific problems. Data should be extracted with the help of templates, it is used to learn data extraction patterns to extract data for populating its slots. RAPIER tool only extract single record from each document so it is also known by single slots. Tools learn the pattern of single slot extraction that makes use of syntactic and semantic information including part of speech tagger. Extraction Rule is based on three patterns:

- Pre filler:- Matches text immediately foregoing the filler

- Filler:- Pattern matches the real slot filler
- Post filler:- Matches the text immediately

5.3.2 SRV:
SRV is different from the previous techniques by learning over an explicit set of simple and relational features. In which simple feature map a token with a discrete value and relational feature maps a token with another token. The main disadvantage of SRV is that it is only used to solve slot filling problems. Learning of rules consists in identifying and generalizing the features found in the training examples. It is also a single slot tool.

5.3.3WHISK:
In this technique, extraction rules are generated from a given set of training example documents. These rules are based on a form of regular expression patterns that identify the context of relevant phrases and the exact delimiters of those phrases. Algorithm used in this system follows the top-down approach. There are certain disadvantage occurred in this approach like it required large volume of training data to produce extraction rules, must see all the permutation of the items. But WHISK is multi slot tool, it means it is capable of extracting multiple records from a document.

5.4 Wrapper Induction Tools:
5.4.1WIEN:
WIEN tools were developed in 1997. In order to extract data it is assumed that items are always in fixed and known order. It takes a set of pages as input where data of interest is served as an example and returns a wrapper that is consistent with each labeled page. It is easy to extract the data as fast as possible but it also deal with certain disadvantage that it cannot handle with permutation and missing items, does not use semantic classes.

5.4.2 SoftMealy:
Like WIEN, Soft Mealy is also a Wrapper Induction Tool that generates extraction rules by using a special kind of finite state automata known as a finite state transducer. To extract the data from web page, firstly wrapper segments an HTML string into tokens then an algorithm tries to induce extraction rules based on the context formed by the separators of adjacent attributes present on given training examples. Secondly FST takes sequence of token as an input and matches the context separators with contextual rules to determine state transitions.

5.4.3 STALKER:
STALKER tool deals with the hierarchical data extraction. It represent the page as a tree like structure in which leaves contain the data that should be extracted and internal nodes represent list of tuples, each item in a tuple can either be a leaf or another list. A wrapper can extract any leaf by determining the path from root to the corresponding leaf. Tree like structure also known as an Embedded Catalog Tree.

5.5 Modeling Based Tools:
5.5.1NoDOSE:
NoDOSE stands for Northwestern Document Structure Extractor, it is an interactive tool for users to hierarchically decompose semi-structured document. These tools are semi automatically determined the structure of document and then extract data. Using GUI, user hierarchically decompose the document into certain levels, with each level of decomposition user builds an object with a complex structure and then decompose into simple structure.

5.5.2DEByE:
DEByE stands for Data Extraction by Example. It generates extraction pattern by taking set of objects from a web page as an input. These patterns allow extracting new objects from other similar pages Using GUI, users are able to assembled nested tables using piece of data received from the sample page. DEByE generates object extraction patterns that indicate the structure and the textual surroundings of the objects to be extracted. It adopts a bottom up extraction strategy.

5.6 Ontology Based Tools:
5.6.1BYU:
BYU tool is developed in Brigham Young University. This tool is different from the other tool as we discussed earlier as it directly rely on data. In BYU, Extraction rule does not depend on the presentation structure. Ontology tools are previously constructed to describe the data of interest, including relationships, lexical appearance, and context keywords. By parsing this ontology, the tool can automatically produce a database by recognizing and extracting data present in documents or pages given as input. It mainly created with graphical editing tool. If the ontology is representative enough the extraction process is fully automatic.

VI. QUALITATIVE ANALYSIS CRITERIA

This section shows the criteria for analyzing the tools of Web Data Scraper. It defines how the studied tools support some features which are more important for data extraction.

6.1 Degree of Automation:
It determine amount of human effort required to perform web data scraper tools or you can say how much work user has to perform for generating a wrapper and for obtaining essential web data.

6.2 Supports for Complex Object:
Web pages usually contain large amount of data, so object formation in the web sources could be complex. Only some systems are able to handle this kind of data. It is the responsibility of tools to represent the web page data in much simpler form as possible.

6.3 Page Contents:

Wrapper generation tools can be applied in two types of pages. It means page content can be distinguished into two categories: Semi-structured Text and Semi-structured Data. First category bring free text from which data items can only be inferred and second category represents data items implicitly formatted to be recognized.

6.4 Ease of Use:

Last generation tools can develop the wrapper with the help of Graphical User Interface (GUI), because it offers the wizards to develop the wrapper. Platforms often features what you see is what you get (WYSIWYG) editor interfaces, integration with web browser etc.

6.5 XML Output:

XML criteria play a significant rule in data representation and exchanging of data on the web. According to W3C, XML is a standard language for representing the data in semantic web form. The tools must have the capability to show the extracted data in XML format.

6.6 Supports for Non-HTML Sources:

Large amount of data stored on the net are stored in semi-structured form that is not represented by the HTML pages but it is present in text files form such as email messages, program code etc. These text files are also be considered by the extraction tools. NLP based tools fits better in this domain.

6.7 Resilience and Adaptiveness:

Web sources are usually updated without any forewarning. It means frequency of changing the web page is not fixed, so that's why the system, which generates the wrapper, must have high degree of resilience in order to show better performance.

Also the adaptiveness of the wrapper is also equally important, moving from a specific web sources to another within the same domain is a great advantage.

VII. CONCLUSION

In this paper we presented a short survey on Web Data Scraper tools in order to extract the data from the web. Data should be extracted by generating the wrapper with the help of extraction rules. Extraction is necessary because web contains huge amount of unstructured data so there is need to develop the structured data. The complete objective of this paper is to show, how tools help the user in extracting the data without using much effort. Analysis criteria also help in identifying the certain features supported by tool. Table 1 presents the complete Qualitative Evaluation.

VIII. SCOPE FOR FURTHER RESEARCH

There are some future directions and challenges that can be foreseen. Some of them comprise how to address enormous scaling issues of the extraction problem, robustness of the process and design and implementation of auto-adaptive wrapper.

REFERENCES

[1] Searchsoa website : www.searchsoa.techtarget.com

[2] Adelberg, B.Nodose: A Tool for semi-Automatically extracting structured and semi-structured data from text documents.In proceeding of ACM SIGMOD International conference on management of data (Seattle, WA, 1998) pp. 283-294.

[3] Arocena, G.O., Mendelzon, A.O.WebOQL: Restructuring Documents, Databases and Web. In proceedings of the 14th IEEE international conference on data engineering (Orlando, Florida, 1998) pp. 24-33.

[4] Califf, M.E., Mooney, and R.J.: Relational learning of pattern-match rules for information extraction. In proceeding of 16th national conference on artificial intelligence and 11th conference on innovative applications of artificial intelligence (Orlando, Florida, 1999) pp. 328-334.

[5] Crescenzi, V., Mecca, G.: Grammer have exceptions. Information Systems 23, 8 (1998), 539-565.

[6] Baumgartner, R., Gatterbauer, W., Gottlob, G. 2009: Web data extraction system. Encyclopedia of database systems, 3465-3471.

[7] Valter, G. Mecca, Paolo 2001: Road Runner Toward Automatic Generation from Large Web Sites

[8] Noha Negm, Passent, Abdel. B. Salem 2012:A survey of Web Information Extraction Tools

[9] Alberto, Berthier, Altigran, Julianan S.Teixeira : A brief survey of Web Data Extraction Tools

[10] Emilio Ferrara, Giacomo F., Robert Baumgartner: Web Data Extraction, Applications and Techniques: A survey. In ACM Transcations on Computational Logic June 2010.

[11] Baumgartner, R., Flesca, S., and Gottlob, G. Visual Web information extraction with Lixto. In Proceedings of the 26th International Conference on Very Large Database Systems (Rom, Italy, 2001), pp.119-128.

[12] Buneman, P. Semistructured data. In Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (Tucson, Arizona, 1997), pp.117-121.

[13] Califf, M. E., and Mooney, R. J. Relational Learning of Pattern-Match Rules for Information Extraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence (Orlando, Florida, 1999), pp.328-334.

[14] Crescenzi, V., and Mecca, G. Grammars Have Exceptions. Information Systems 23,8 (1998), 539-565.

[15] Crescenzi, V., Mecca, G., and Merialdo, P. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In Proceedings of the 26th International Conference on very large Database Systems (Rome, Italy, 2001).

[16] Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Kai Ng, Y., Quass, D., and Smith, R. D. Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages. Data and Knowledge Engineering 31, 3 (1999), 227-251.

Tools		Extraction Rule Obtained By	Method used to Extract Data	Degree of Automation	Support for Complex Objects	Easy to Use	XML O/P	Support for Non-HTML Sources	Type of Page Contents	Resilience and Adaptiveness
Languages For Wrapper Development	Minerva	Production	Procedural Language	Manual	Coding	No	Yes	Partial	*SD	No
	TSIMMIS	Commands	File Specification	Manual	Coding	No	No	Partial	*SD	No
	Web-OQL	Query	SQL Language	Manual	Coding	No	No	None	*SD	No
HTML-Aware	W4F	Wizard	HTML Tag	Semi-Automatic	Coding	Yes	Yes	None	*SD	No
	XWRAP	Component Library	HTML Tag	Automatic	Yes	Yes	Yes	None	*SD	No
	Road Runner	Schema	HTML Structure	Automatic	Yes	Yes	No	None	*SD	No
NLP-Based	RAPIER	Template	Learning Algorithm	Semi-Automatic	No	Yes	No	Full	*ST	No
	SRV	Token Oriented	Learning Algorithm	Semi-Automatic	No	Yes	No	Full	*ST	No
	WHISK	Training Examples	Regular Grammar	Semi-Automatic	No	Yes	No	Full	*ST	No
Wrapper Induction	WIEN	Induction Heuristic	HTML Tag	Semi-Automatic	No	Yes	No	Partial	*SD	No
	SoftMealy	Finite State Automata	HTML Tag	Semi-Automatic	Partial	Yes	No	Partial	*SD	No
	STALKER	Token	HTML Tag	Semi-Automatic	Yes	Yes	No	Partial	*SD	No
Modeling Based	NoDoSE	Object	GUI	Semi-Automatic	Yes	Yes	Yes	Partial	*SD	No
	DEByE	Object Pattern	GUI	Semi-Automatic	Yes	Yes	Yes	Partial	*SD	Partial
Ontology-Based	BYU	Database	HTML Tag + GUI	Manual	Coding	Yes	No	Full	*ST/SD	Yes

Table 1: Qualitative Evaluation

*SD stands for Semi-structured Data
*ST stands for Semi-structured Text

Sneh Nain, Did B.Tech in 2011 and now pursuing M.Tech in Computer Science Engineering. Her research interest includes Deep Web Information Retrival, Digital Watermarking, Cryptography and Networking.

